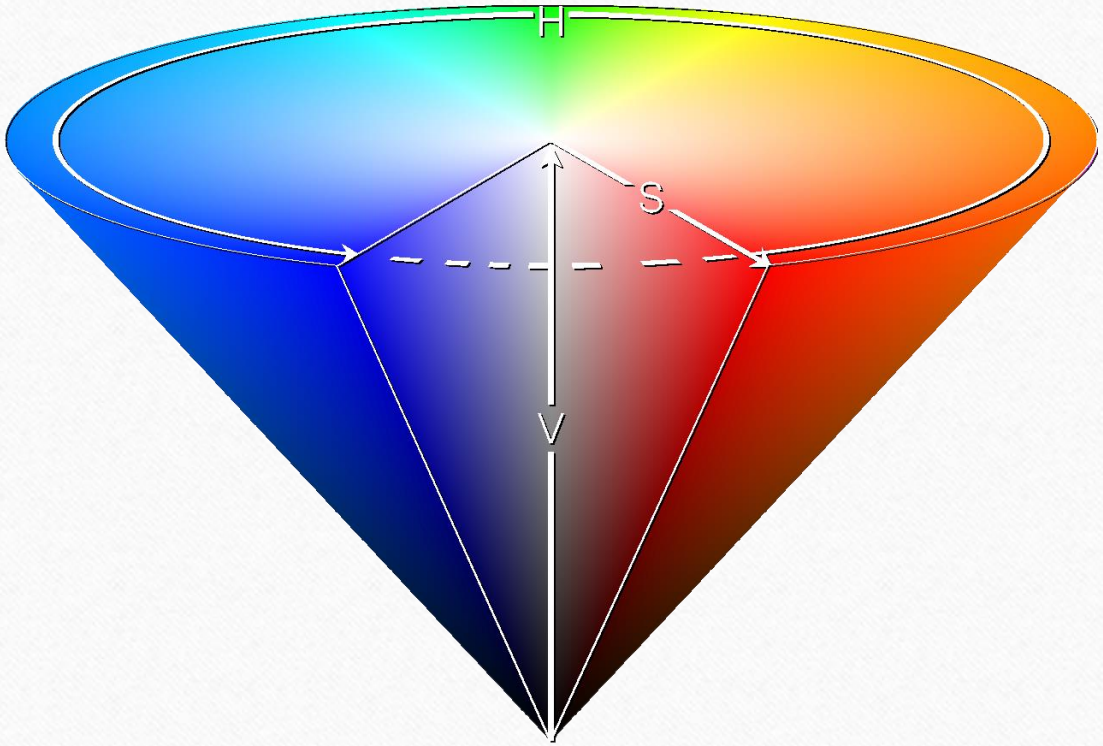
دوره آموزشی بینایی ماشین کاربردی

آکادمی ربوتک – آزمایشگاه تعامل انسان و ربات

جلسه 5 – آشنایی با HOG و شروع بحث Classification

آنچه گذشت ؟ !



الگوریتم SIFT ؟

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} \quad \cdots \quad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \cdot \\ \cdot \\ \cdot \\ u_{n-3} \\ u_{n-2} \\ u_{n-1} \\ u_n \end{bmatrix}$$

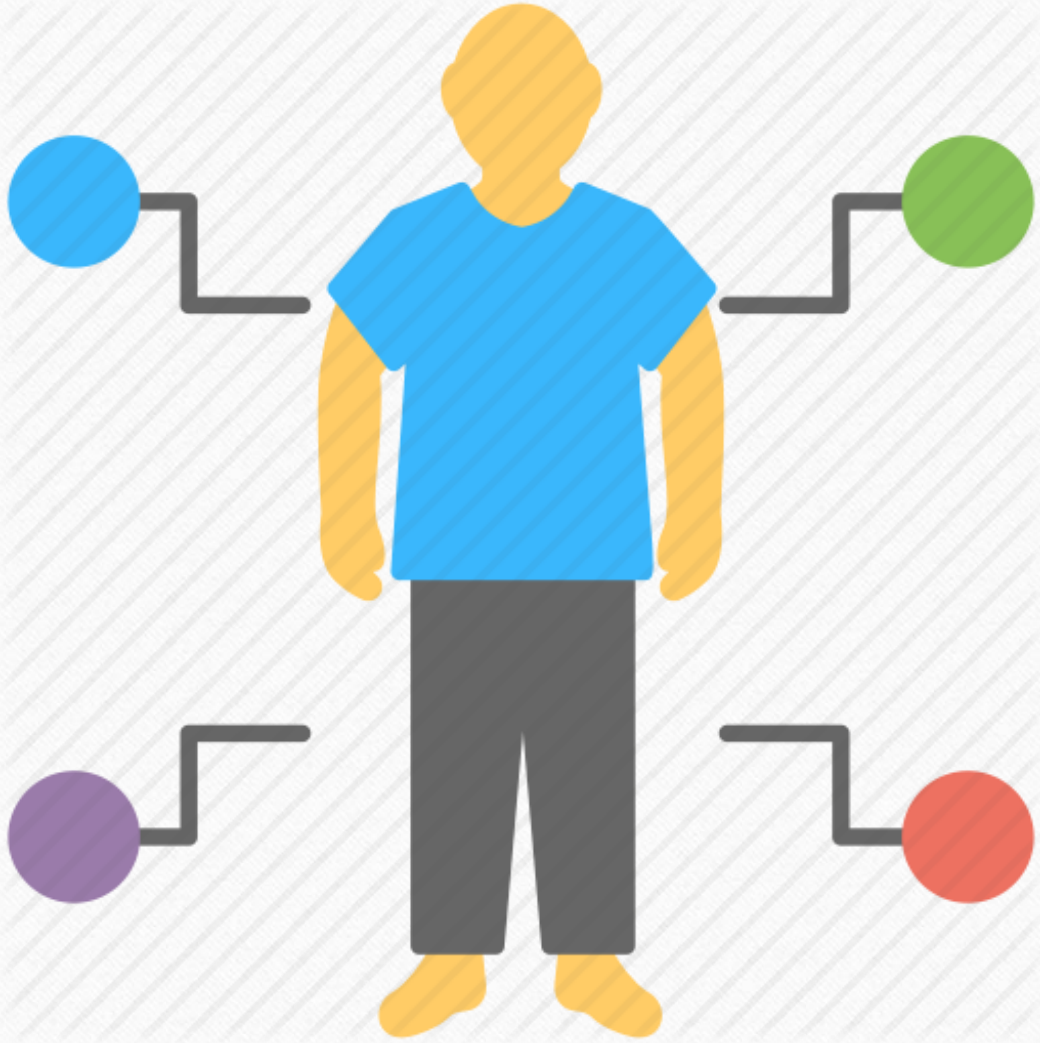
آنچه در این جلسه خواهیم گفت

1. کار با الگوریتم SIFT و حل دو مثال
2. الگوریتم HOG
2. طبقه بندی (Classification)

کمی کار با الگوریتم SIFT

بررسی رابطه ی بین نقاط کلیدی و Scaleو Rotation

حل یک مثال:



تشخیص حضور یک جسم به کمک الگوریتم SIFT

تمرین : تشخیص اعداد به کمک SIFT

الگوریتم تشخیص ویژگی HOG

توصیف کل تصویر با یک **بردار بزرگ**

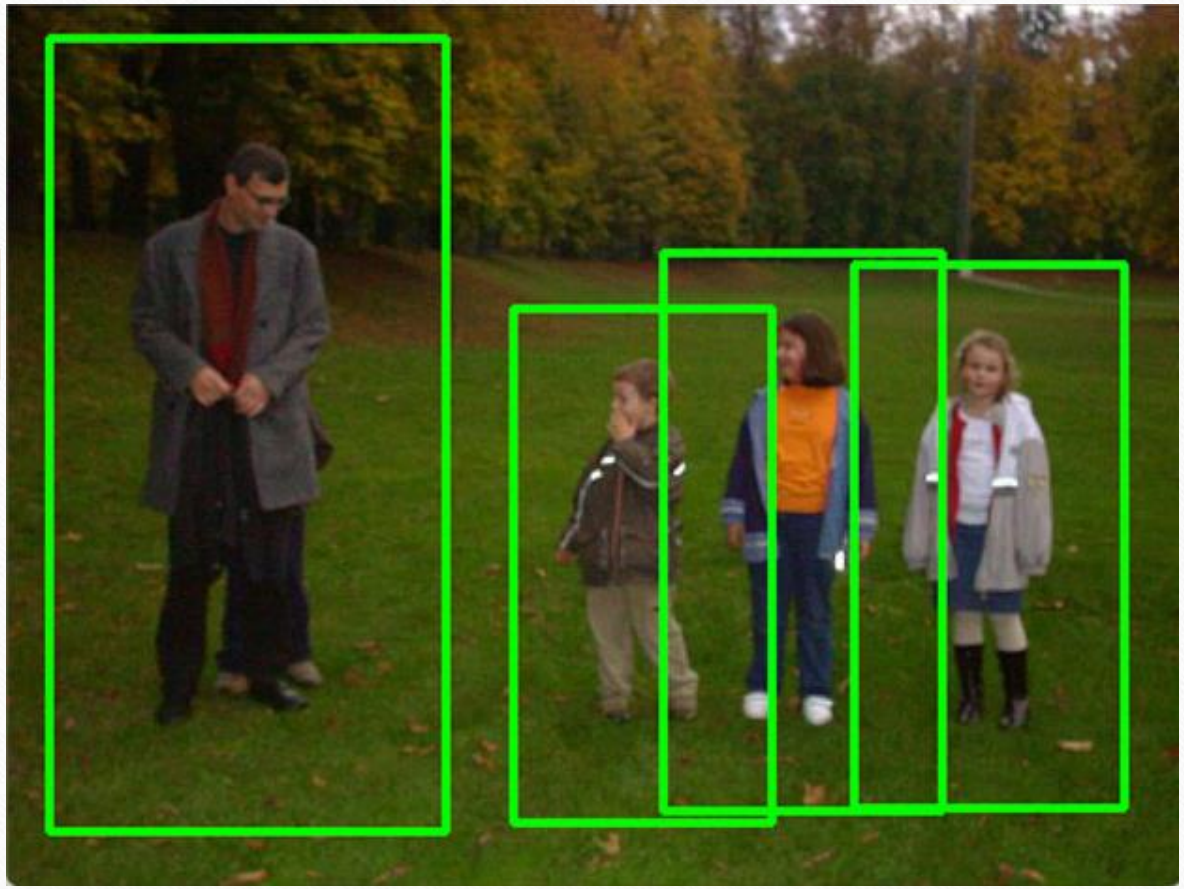
تفاوت با SIFT ؟

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix}$$

تاریخچه HOG

معرفی شده در مقاله :

- Histogram of Oriented Gradient for Human Detection - 2005



- الهام گرفته از الگوریتم SIFT
- با آن میتوان یک Custom Object detector ساخت.
- پیاده سازی شده در OpenCV

مراحل الگوریتم HOG

1. نرمالیزه کردن تصویر
2. محاسبه گرادیان تصویر
3. محاسبه هیستوگرام ویژگی
4. نرمالیزه کردن بلوکی
5. محاسبه بردار ویژگی تصویر

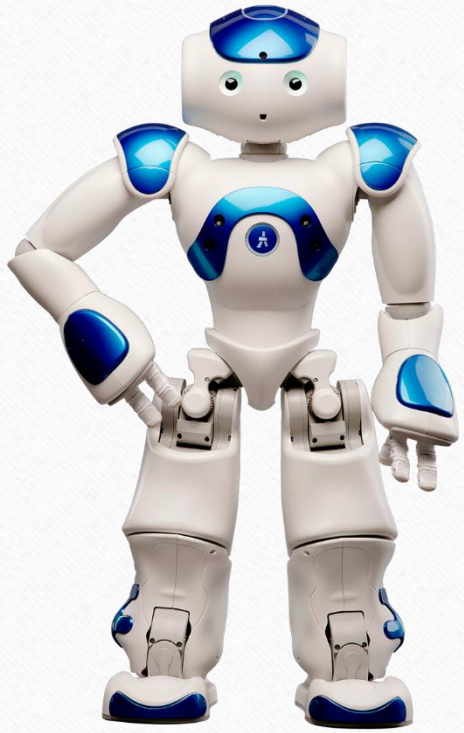
گام 1 : نرمالیزه کردن تصویر (اختیاری)

در این مرحله تلاش می شود تا با روش های مختلف مقدار عددی پیکسل ها کاهش یابد. چرا؟

Logaritmic Normalization

لگاریتم هر پیکسل محاسبه می شود.

$$p' = log\ p$$



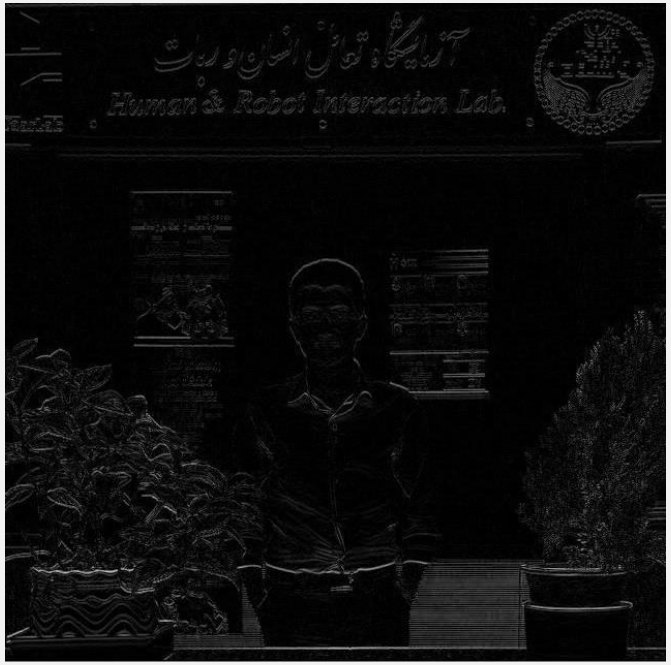
Square root Normalization

ریشه دوم هر پیکسل محاسبه می شود.

$$p' = \sqrt{p}$$

گام 2 : محاسبه گرادیان تصویر در جهات x و y

محاسبه Gx



کرنل: [-1, 0, 1]

محاسبه Gy

کرنل: [-1, 0, 1].T

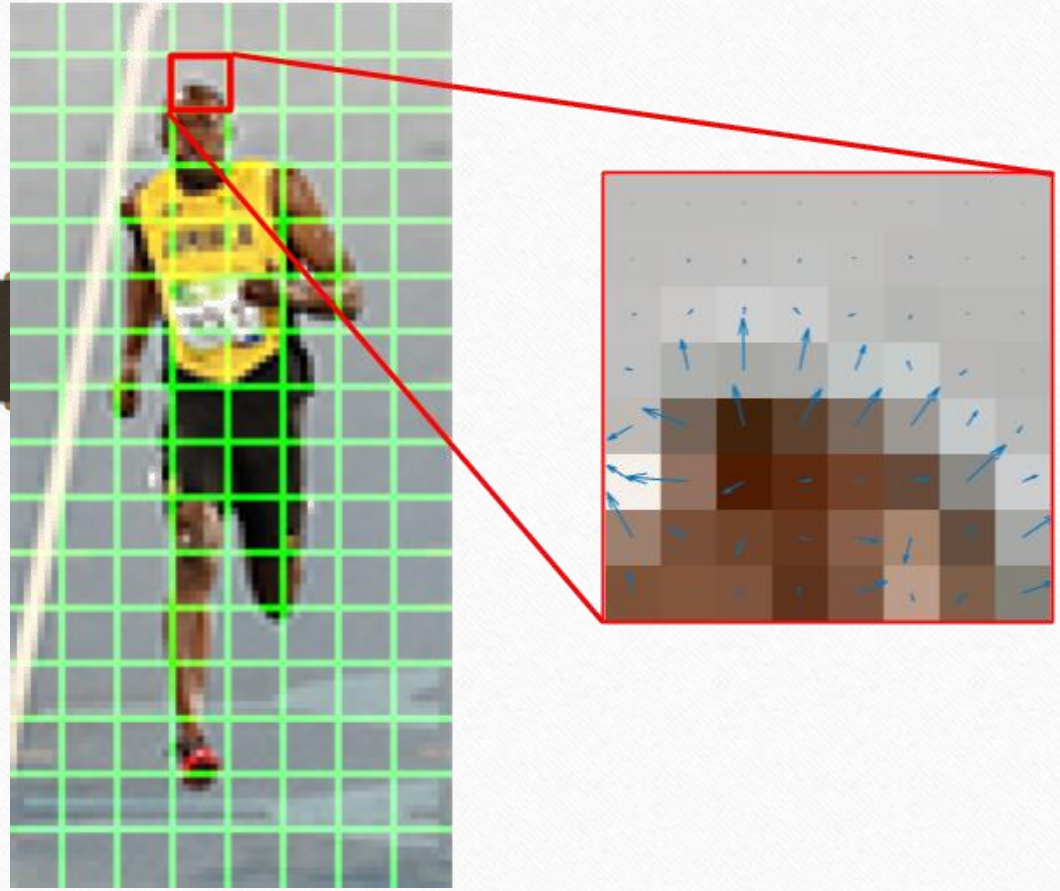
محاسبه اندازه و جهت گرادیان

$$\theta = \arctan2(G_y, G_x) \qquad G = \sqrt{G_x^2 + G_y^2}$$

مقایسه بین کرنل ها

Mask Type	1-D centred	1-D uncentred	1-D cubic-corrected	2×2 diagonal	3×3 Sobel
Operator	$[-1, 0, 1]$	$[-1, 1]$	$[1, -8, 0, 8, -1]$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Miss rate at 10^{-4} FPPW	11%	12.5%	12%	12.5%	14%

گام 3 : محاسبه هیستوگرام ویژگی



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

تصویر به سلول هایی 8 در 8 تبدیل می شود و برای هر کدام دو ماتریس اندازه و جهت معرفی می شوند.

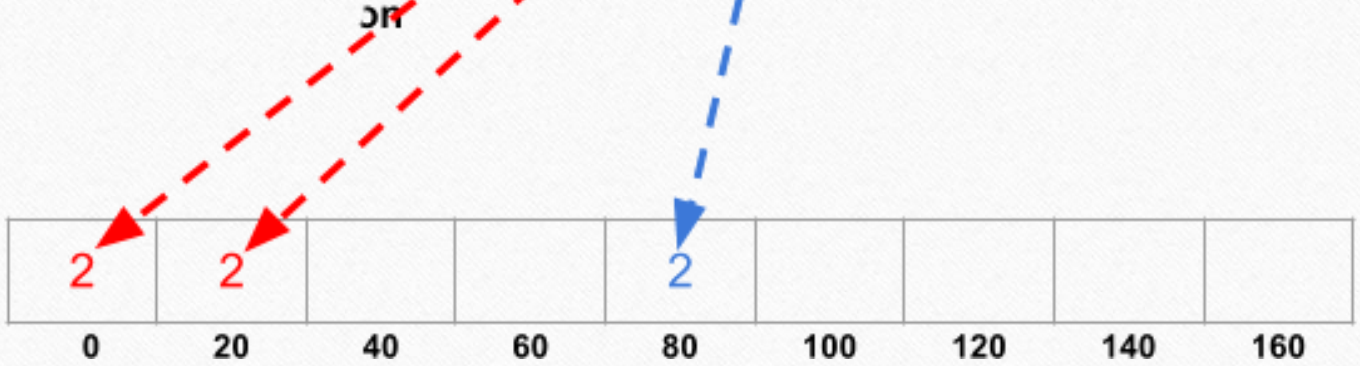
جهت ها از 0 تا 180 درجه تقسیم بندی می شود. (به جای 0 تا 360 درجه)

ماتریس جهت

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

ماتریس اندازه

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



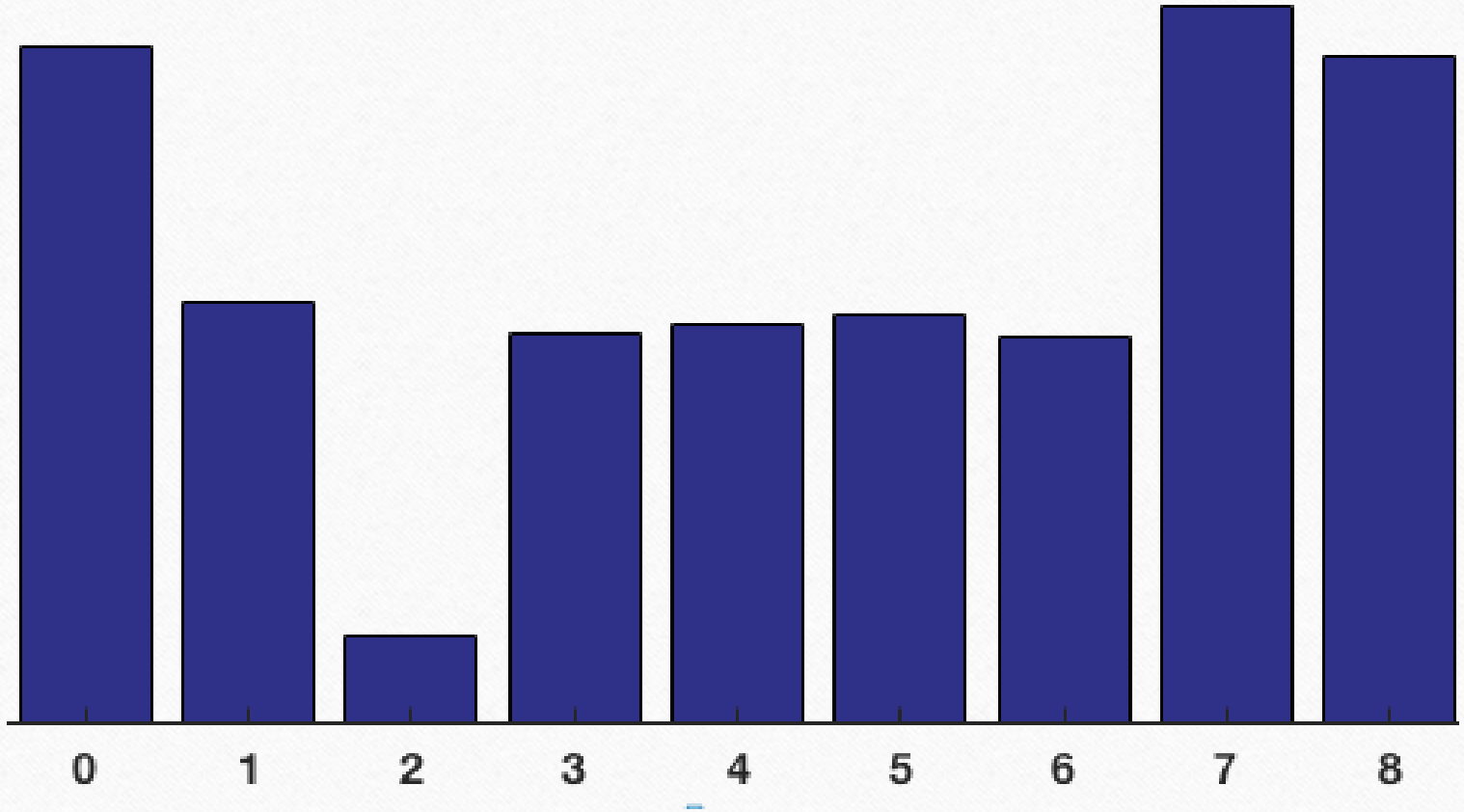
از 0 تا 180به 9 بخش تقسیم می شود. 0 و 20 و ... و 160

به کمک دو ماتریس مرحله قبل سعی میکنیم هیستوگرام را بسازیم.

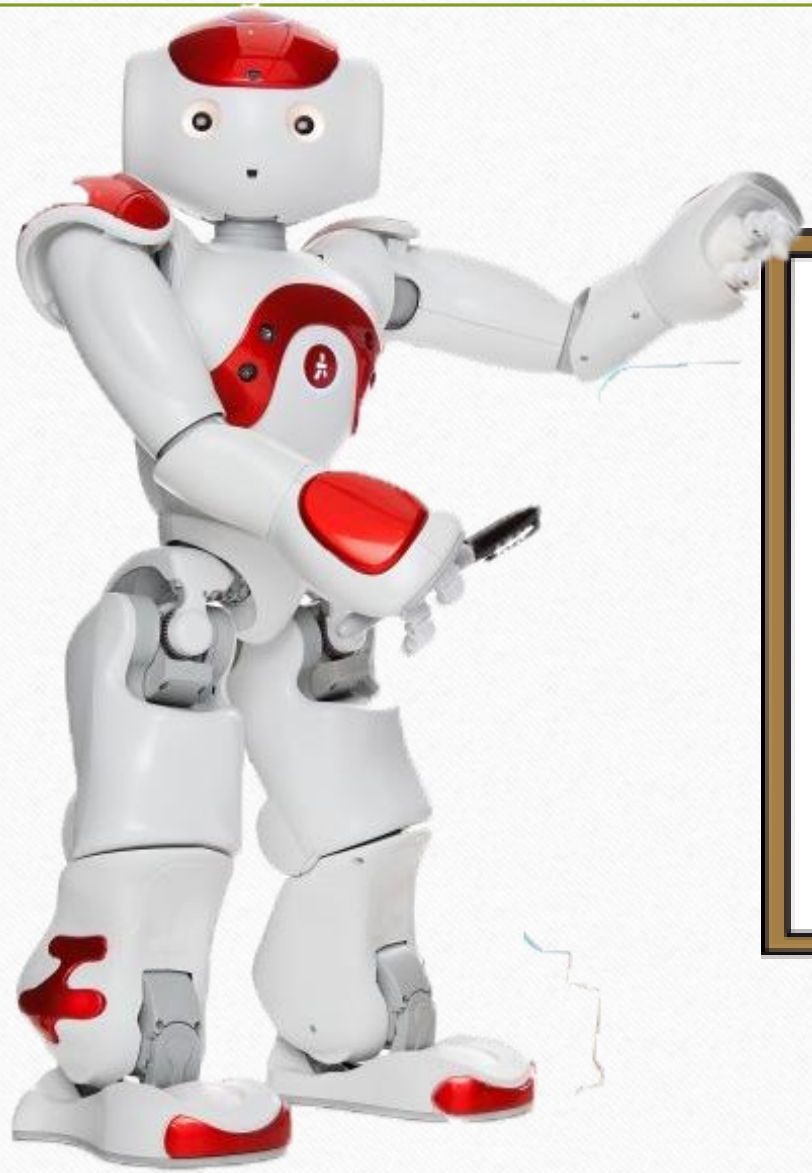
فرمول محاسبه

$$\frac{\max\_\lim - direction}{20} * magnitude$$

در نهایت هیستوگرامی شبیه به شکل زیر ساخته می شود:



یادآوری

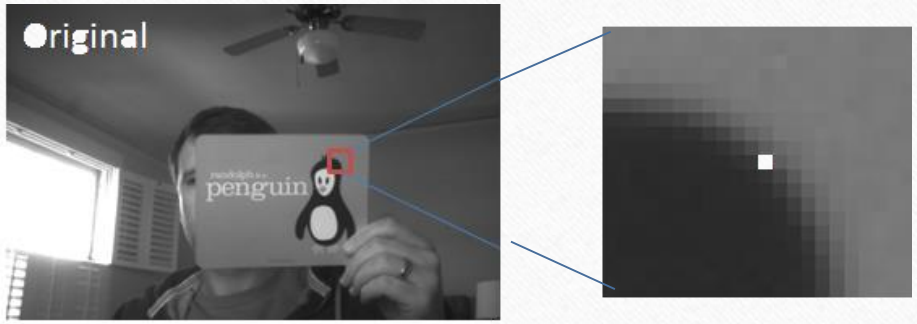


نرمالیزه کردن یک بردار ساده

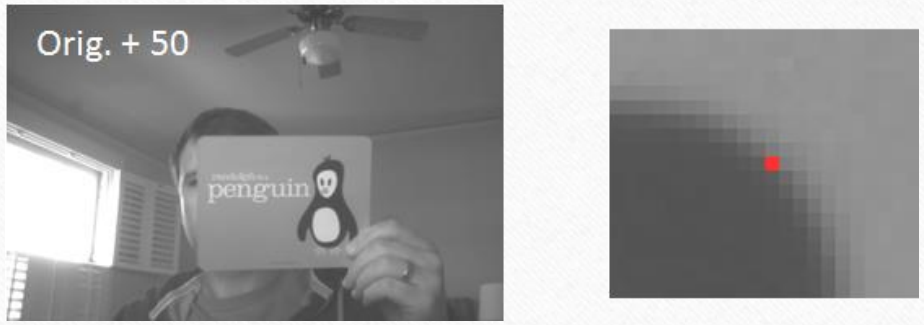
$$v = [128, 64, 32]$$

$$\|v\| = \sqrt{128^2 + 64^2 + 32^2}$$

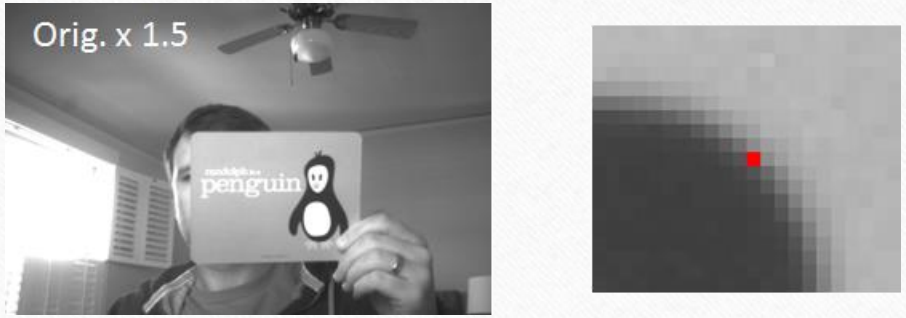
$$\textbf{بردار نرمال شده} = \frac{v}{\|v\|} = [0.87, 0.43, 0.22]$$



تصویر اصلی

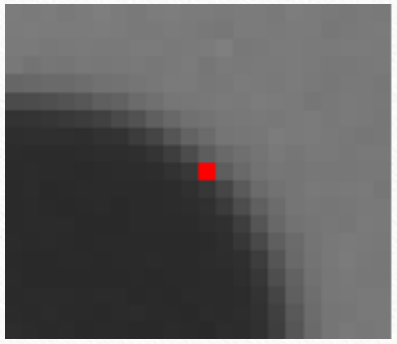


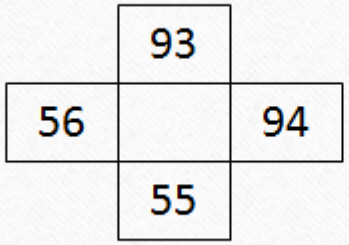
تصویر اصلی + 50



تصویر اصلی × 1.5

بردار نرمال شده

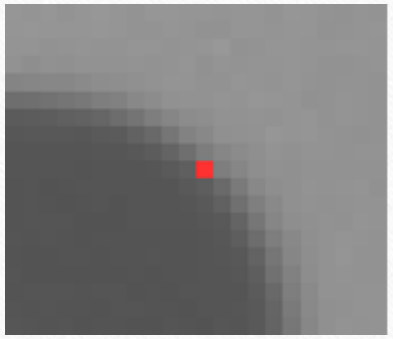


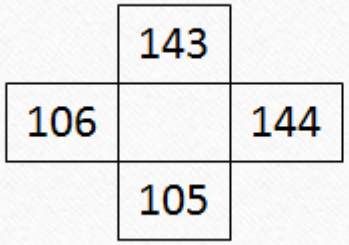


$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$

$$[0.71, 0.71]$$

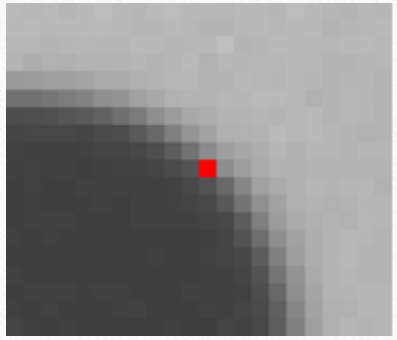


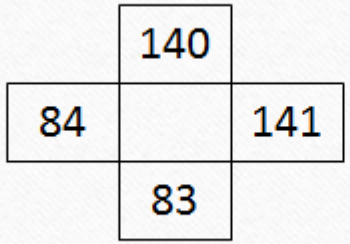


$$\nabla f = \begin{bmatrix} 38 \\ 38 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(38)^2 + (38)^2} = 53.74$$

$$[0.71, 0.71]$$



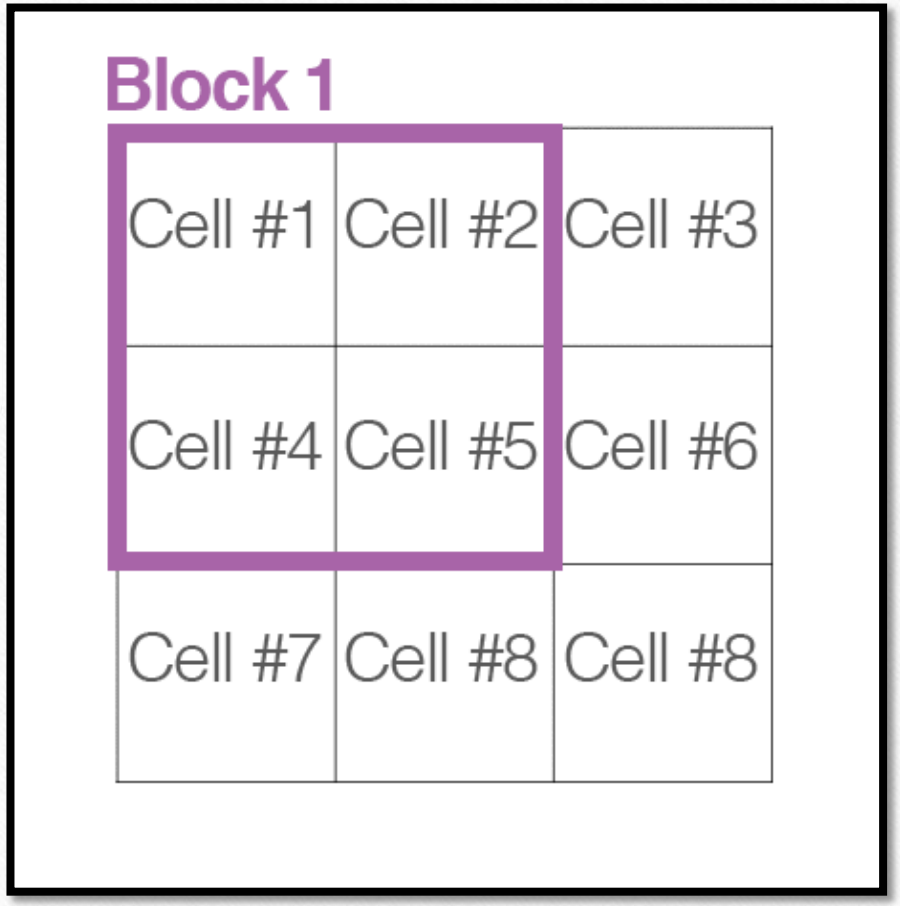


$$\nabla f = \begin{bmatrix} 57 \\ 57 \end{bmatrix}$$

$$|\nabla f| = \sqrt{(57)^2 + (57)^2} = 80.61$$

$$[0.71, 0.71]$$

گام 4 : نرمالیزه کردن بلوکی

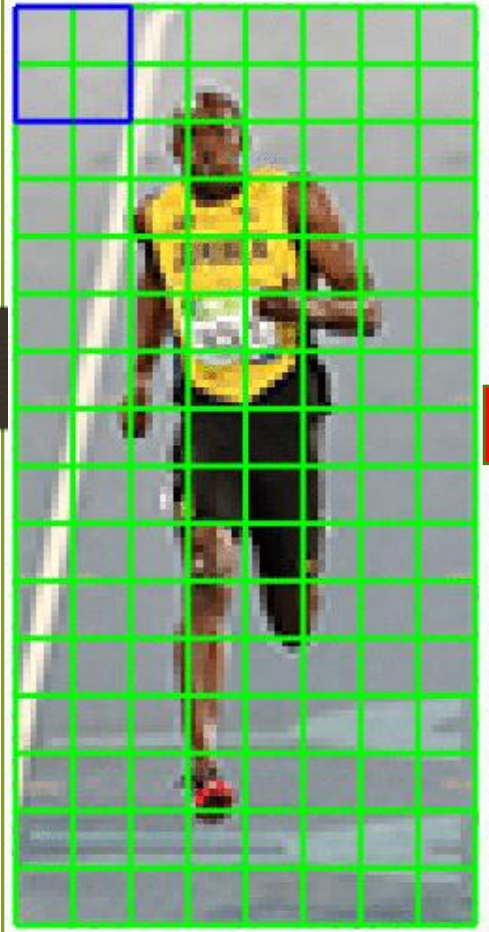


مشابه بردارها میتوانیم هیستوگرام سلول ها را نیز نرمالیزه کنیم.

کار بهتر آن است که بلوک ها را نرمالیزه کنیم. (هر بلوک 4 سلول است)

بلوک ها می توانند با هم Overlap داشته باشند. به این ترتیب هر سلول چند بار در خروجی ظاهر می شود.

گام 5 : استخراج بردار ویژگی



7 بلوک در راستای افقی

15 بلوک در راستای عمودی

هر بلوک یک بردار 36 تایی است

یک بردار 7 * 15 * 36

یعنی 3870 تایی

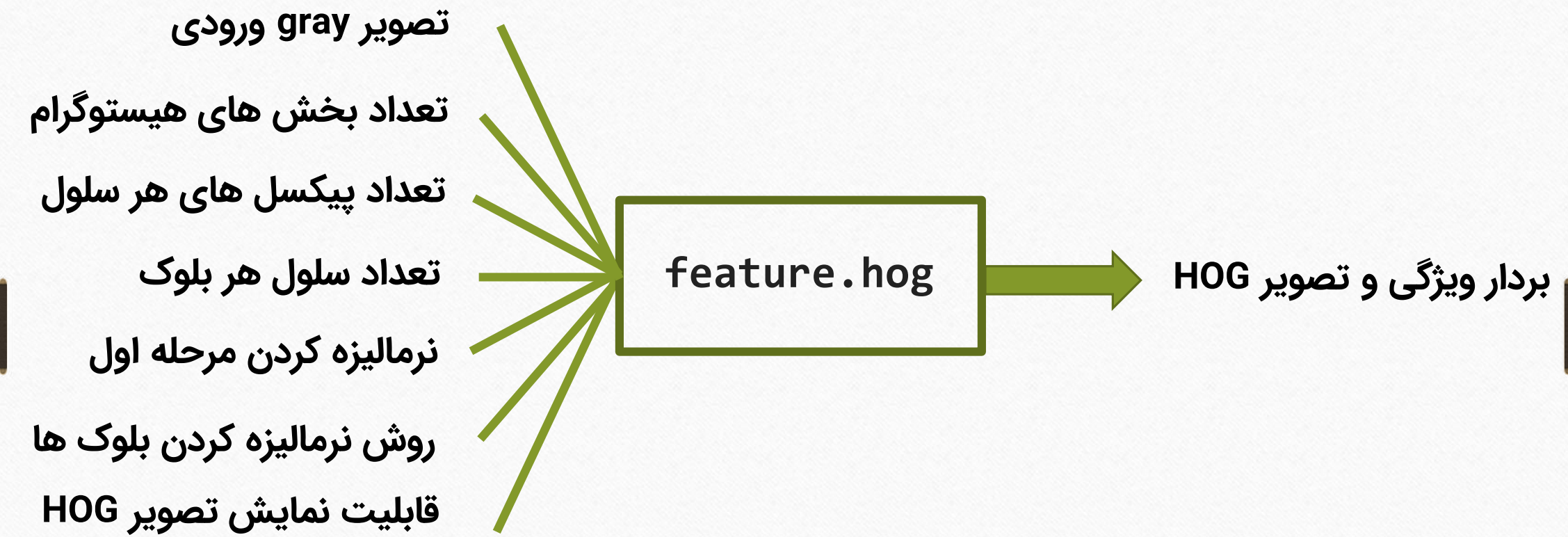
برای یک تصویر 8192 پیکسلی

کدنویسی

در OpenCV الگـوریتم HOG پیاده سازی شده است ولـی کتابخانـــــــه sikit-image انعطـــاف بـــالاتری در HOG دارد.

Sikit-image یک کتابخانـه در زمینه بینایی ماشین است که امکان کدنویسی به کمک پایتون را فراهم می کند.

دستور 32 : استفاده از HOG



مثال :

```
from skimage import feature
(H, hogImage) = feature.hog(logo, orientations=9, pixels_per_cell=(10, 10),
cells_per_block=(2, 2), transform_sqrt=True, block_norm="L1", visualise=True
```