

اگر سیستم HOG به محاسبه گرادیان نیاز داریم پس اول از صدها بریم سیستم گرادیان چه؟

گرادیان یعنی چی؟ یعنی گرفتن مشتق جزئی نسبت به محورها (مثلا محورها x و y یا x و z و فضای ۳ بعدی) لے مثال:

$$f(x, y) = x^2y + 3y$$

$$\frac{\partial f}{\partial x} = 2xy \quad \frac{\partial f}{\partial y} = x^2 + 3$$

$$\nabla f(x, y) = (2xy, x^2 + 3) \rightarrow \text{ماسبه بردار گرادیان}$$

گرادیان دو مفهوم دارد:

- ① بردار گرادیان \leftarrow جهت سریع ترین افزایش تابع: یعنی هر جا باشی اگر در جهت گرادیان حرکت کنی، مقدار تابع با بیشترین نرخ ممکن افزایش پیدا می کند.
- ② اندازه گرادیان \leftarrow بیانگر مقدار این نرخ (شیب)

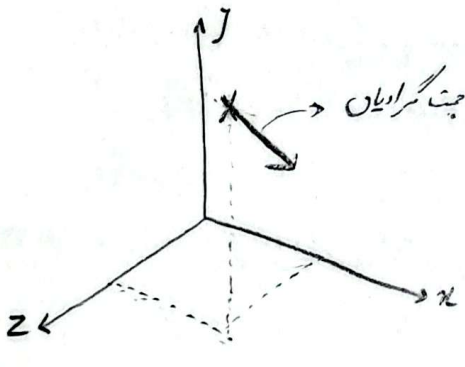
$$\hat{u} = \frac{\nabla f}{\|\nabla f\|} \rightarrow \text{گرادیان } f$$

$$\|\nabla f\| \rightarrow \text{اندازه گرادیان } f$$

$$\nabla f = (0, \Delta, \square)$$

مشتق جزئی نسبت به x مشتق جزئی نسبت به y مشتق جزئی نسبت به z

$$\|\nabla f\| = \sqrt{0^2 + \Delta^2 + \square^2}$$



این یعنی اگر به نحوی از این نقطه (مختصات) حرکت کنی و مقدار تابع با بیشترین سرعت ممکن زیاد بشه باید در امتداد راستای این فلش حرکت کنی!

HOG چیست؟ \leftarrow HOG یا (Histogram of oriented Gradients) یک توصیفگر ویژگی است. این الگوریتم بر این فرض استوار است که ظاهر و شکل یک شیء ادویه تصویر برای مقایسه خوبی با توزیع گرادیان های شدت نور یا به ما توصیف می کند. HOG به جای اینکه به مقایسه شکل ها (که به نور حساس هستند) نگاه کند، به جهت و شدت لبه ها در نواحی کوچک به تصویر نگاه می کند و از متون به بردار ویژگی (Feature Vector) پیاز که برای مقایسه شبیه اثر انگشت.

مراحل الگوریتم HOG

فرض می کنیم یک تصویر 64×128 پیکسلی داریم:

① پیش پردازش (Preprocessing):

تصحیح ناما \leftarrow اول نور تصویر و نرمال سازی می کنیم تا اثرات سایه های تند یا نورهای شدید کاهش یابد. این کار به افزایش دقت گرادیان کمک میکند.

تبدیل به تصویر grayscale ← HOG به رنگ نیاز نداره چون فقط با لبه ها و شکل ها سروکار داره پس تصویر رو در scale gray تبدیل می کنیم تا محاسبات برای سیم راحت تر بشه.

۷) سایه گرادیان

می خواهیم بدوینیم که برای "هر پیکسل" شدت تغییر روشنایی در جهت افقی (X) و در جهت عمودی (Y) چقدر است.

ساده ترین راه برای اینکار اینه که برای یک پیکسل هم پیکسل های بالا، پایین، چپ و راستش رو داشته باشیم:

فرض کنی می خواهیم گرادیان افقی و عمودی پیکسل وسطی رو بدست بیاریم:

	100	
70		120
	50	

$$G_x = 120 - 70 = 50$$

$$G_y = 50 - 100 = -50$$

اما برای اینکه نتیجه دقیق تر و مقادیر تری نسبت به نور داشته باشیم، از اپراتور سوبل (Sobel Operator) استفاده می کنیم. اپراتور سوبل فقط به ۴ همسایه اصلی نگاه میکنه بلکه کل همسایگی 3x3 یک پیکسل (یعنی تمام 8 پیکسل اطراف آن) رو در نظری بگیره. این اپراتور با وزن دهی به پیکسل های مختلف، گرادیان رو محاسبه میکنه. برای مثال برای محاسبه G_x به تفاوت بین ستون راست و چپ وزن بیشتری میده.

❖ سوال: پیکسل های گوشه مقادیر که به سری از پیکسل های کناری رو ندارند چی؟ مثلاً بالا ترین و سمت چپ ترین پیکسل به عکس، نقطه پیکسل های سمت راست و پیکسل های پایین رو در یک همسایگی 3x3 داره. برای محاسبه گرادیان اون پیکسل چکار می کنیم؟

❖ جواب: برای پیکسل های که همسایگی کامل ندارند، می تونیم کمترین 3x3 رو مستقیماً اعمال کنیم. برای حل این مشکل، چند راه حل استاندارد وجود داره که رایج ترینشون تکنیک لایه گذاری (padding) است. ایده اصلی این است که ما قبلی از سایه گرادیان، یک "قاب" یا "لایه" مصنوعی در اطراف تصویر اصلی اضافه می کنیم تا هر پیکسل در تصویر اصلی، یک همسایگی کامل 3x3 داشته باشه.

اولی روش های padding عبارتند از:

۱- لایه گذاری با صفر ← اضافه کردن یک قاب از پیکسل های سیاه (با مقدار صفر) در تصویر

۲- تکرار ← در این روش مقدار پیکسل های به دوگی می کنیم و در قاب مصنوعی قرار میدیم یعنی مقدار پیکسل به بیرون کشیده میشه.

۳- انعکاس ← این روش که معمولاً بهترین نتیجه رو میده، پیکسل های تصویر رو مثل یک آینه توی یک منعکس میکنه.

راه حل دیگه برای محاسبه گرادیان پیکسل های یک تصویر، نادیده گرفتن پیکسل های به است. یعنی اگر تصویر ورودی ۱۰۰x۱۰۰ پیکسل باشه، تصویر خروجی گرادیان ۹۸x۹۸. این روش زمانی مورد قبوله که تصویر اندازه بزرگتر باشه که از دست دادن یک ردیف پیکسل در به ما اهمیت زیادی نداشته باشه.

بعد از محاسبه G_x و G_y برای هر پیکسل، دو مقدار مهم رو بدست میاریم:

۱- اندازه گرادیان (Magnitude): نشان دهنده شدت گرادیان است. به عبارتی (مثلا سرزمین یا شیء سیاه و به پس زمینه سفید) اندازه گرادیان بزرگی داره.

$$\text{Magnitude} = \sqrt{G_x^2 + G_y^2}$$

۲- جهت گرادیان (oriented / Direction): نشان دهنده زاویه گرادیان است.

$$\text{oriented}(\theta) = \arctan\left(\frac{G_y}{G_x}\right)$$

جهت ما معمولا در بازه 0 تا 180 درجه (بدون علامت) یا 0 تا 360 درجه (با علامت) می باشد. برای تشخیص علامت ما، جهت های بدون علامت (0 تا 180 درجه) به عنوان مثبت و جهت های با علامت (180 تا 360 درجه) به عنوان منفی در نظر می گیریم.

۳- ساخت هیستوگرام در سلول ما:

ایجاد دو متغیر به نام های Cell (سلول) و block (بلاک) داریم.

سلول (Cell) ← غیر هم پوشان (Non-overlapped) با اندازه پیش فرض 8×8 پیکسل

بلاک (block) ← هم پوشان (overlapped) با اندازه پیش فرض 2×2 سلول و همپوشانی یک سلول

$$1 + \underbrace{(\text{اندازه بلاک})}_{\text{۸x۸}} - (\text{تعداد سلول ها}) = \text{تعداد بلاک ها}$$

که اندازه بلاک است.

هیستوگرام (Histogram) ← یک نمودار میله ای که تعدادی متغیر مختلف رو نشون میده.

در هر سلول، یک هیستوگرام از جهت های گرادیان ساخته میشه. این هیستوگرام معمولا ۹ بین (bin) داره که هر کدام از این بین ها یک بازه 20 درجه ای از زوایا رو پوشش میده (0 تا 180 درجه). برای هر پیکسل از 4×4 پیکسل یک سلول، یک رای به یکی از این های هیستوگرام داده میشه. وزن این رای برابر با اندازه گرادیان همان پیکسل هست.

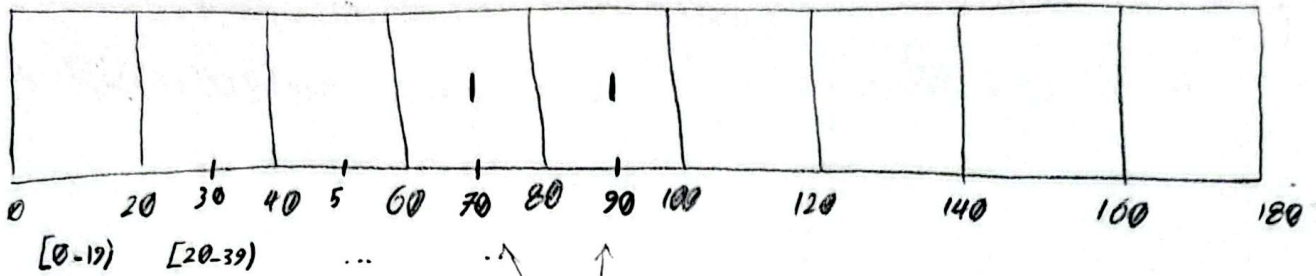
مثال: فرض کن برای یک سلول از یک تصویر، اندازه گرادیان و جهت گرادیان پیکسل هاش رو مطابق زیر بدست آوردیم. حالا می خواهیم به پیکسل بالا چه رای بدیم؟

20	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	12	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	28	108	27	48	110

Gradient magnitude



Gradient - direction (pixel) = 80

Gradient - magnitude (pixel) = 2

$$\left. \begin{array}{l} 90 - 80 = 10 \\ 80 - 70 = 10 \end{array} \right\} \rightarrow 10 = 10 \rightarrow$$

$$\frac{10}{20} \times \frac{\text{Gradient - magnitude}}{2} \text{ واحد به bin ۸۰ تا ۹۰ و}$$

$$\frac{10}{20} \times \text{Gradient - magnitude} \text{ واحد به bin ۸۰ تا ۱۰۰ می‌رسد}$$

مثال ۲: حال فرض کنیم می‌خوایم به پیکسل سمت راست پیکسل قبل بینیم
 $\text{pixel}(2): \begin{cases} 36 \text{ جهت گزینان} \\ 3 \text{ اندازه برداشتن} \end{cases}$

Gradient - direction (pixel 2) = 36

Gradient - magnitude (pixel 2) = 3

۳۶ بین ۲۰ تا ۴۰ می‌رسد:

این کار را برای تمام ۶۴ پیکسل در سلول انجام می‌دهیم. در پایان، هر سلول ۸x۸ بایکس می‌گردد و یعنی (یک بردار ۹ بعدی) تصفیه می‌شود که توزیع به ما در آن سلول روشن می‌دهد.

④ نرمال سازی در بلوک ما:

می‌توانیم برای هر سلولی به تغییرات کمی روشی (کنتر است) حساس هستیم. برای حل این مشکل سلول ما و حتی گروه‌های بزرگتری به اسم بلاک‌ها (صفحه قبل توضیح داده شد) دسته‌بندی می‌کنیم. یک بلاک معمولاً از ۲x۲ سلول تشکیل شده (یعنی ۱۶ x ۱۶ پیکسل) ۱- الحاق می‌توانیم ما به می‌توانیم برای هر سلول درون یک بلوک رو به هم می‌چسبیم. چون هر سلول به بردار ۹ بعدی داشت، پس در هر بلاک به بردار ۳۶ = ۴x۹ بعدی داریم!

۲- نرمال سازی به این بردار ۳۶ بعدی رو با استفاده از یک روش ریاضی مثل مثلاً L2-NORM نرمال سازی می‌کنیم. این کار من

باعث می‌شود که توصیف گریز نسبت به تغییرات در دگرگشت مقادیر باشد.

یه نکته! ← گفتیم که بلوک مامیو شانی دارن یعنی یه بلوک با بلوک بعدی در یک ردیف یا ستون از سلسله مشترکین. (این هم پرشانی باعث می‌شود که نیچر ما پایدار تر و عملکرد الگوریتم بهتر بشه.

⑤ ساخت بردار ویژگی‌های:

در غایت، بردارهای فرمال شده تمام بلوک های مقوسه رو به ترتیب به عددیگه اکتاف می‌گیریم تا به بردار ویژگی‌های دیوار بلند ساخته بشه. این بردار توصیف گریز HOG برای کل پنجره مقصوره:

مثلا ترهسین عکسی که ما اول کار داشتیم (128×64) مقادیر زیر رو داریم:

$$\left\{ \begin{array}{l} \text{تعداد بلوک های افقی: } 7 \\ \text{تعداد بلوک های عمودی: } 15 \\ \text{اندازه بردار هر بلوک: } 36 \end{array} \right. \quad \left\{ \begin{array}{l} \text{تعداد کل بلوک ها: } 105 \\ 15 \times 7 = 105 \end{array} \right. \quad \left\{ \begin{array}{l} \text{اندازه بردار ویژگی های: } \\ 105 \times 36 = 3780 \end{array} \right.$$

سوال: چجوری ما باید تر قراره با مثلا 3780 تا بردار (در واقع پند) می‌فهمه که عکسی که بخش می‌بینیم عکس آدم هست یا نه؟

☆ جواب: بردار 3780 به HOG به خودی خود بی معنیه. پس بی معنی ازش می‌کنه؟ ← یادگیری ماشین

این بردار به یه الگوریتم طبقه بندی مثل ماشین برار پشتیبان (SVM) داده میشه. SVM به الگوریتم یادگیری نظارت شده است که با دیدن هزاران مثال از بردارهای HOG مثلا متعلق به "انسان" (مثال مثبت) و بردارهای HOG متعلق به "غیر انسان" (مثال منفی)، یاد می‌گیره که به مرز تقسیم $(Decision Boundary)$ بین این دو کلاس پیدا کنه. SVM برخلاف الگوریتم های مثل درخت تقسیم $(Decision Tree)$ ، با به اعمال چندین کار می‌کنه تا به اگر درین عرضی ترین داده ممکن که دو کلاس داده رو از هم جدا می‌کنه. این مرز تقسیم ابر صفحه $(Hyperplane)$ نامیده می‌شه.