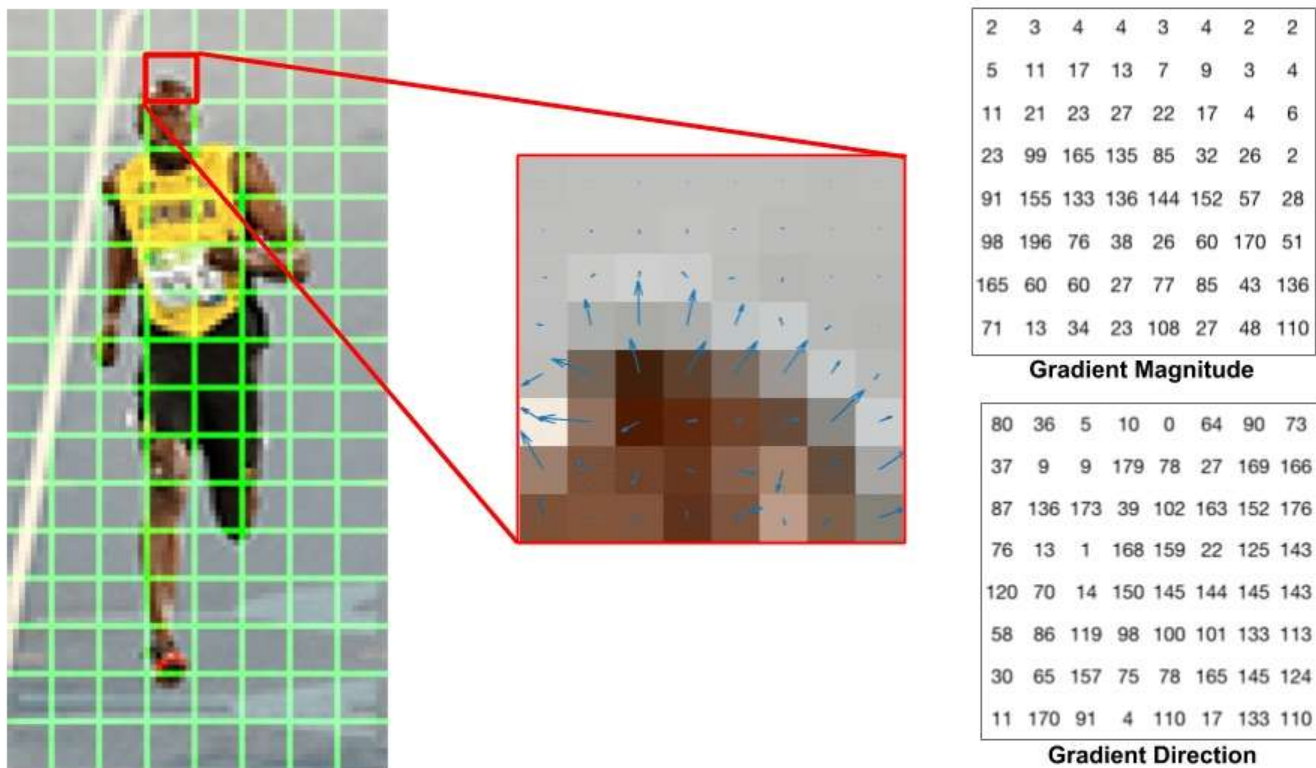


الگوریتم **HOG** یا **Histogram of Oriented Gradients** یکی از روش‌های کلاسیک و بسیار پرکاربرد در بینایی ماشین برای تشخیص اشیاء (به‌ویژه انسان) است. ایده‌ی اصلی این الگوریتم بر پایه‌ی استخراج ویژگی‌های محلی از تصویر است؛ یعنی به جای نگاه کردن به شدت روشنایی پیکسل‌ها، تمرکز روی **جهت و شدت لبه‌ها** قرار می‌گیرد. دلیل این کار آن است که ساختار و شکل اجسام (مثل بدن انسان) بیشتر با لبه‌ها و تغییرات شدت نور مشخص می‌شود تا با خود رنگ یا روشنایی.

فرآیند کار HOG به این صورت است: ابتدا تصویر به بخش‌های کوچک‌تر (cell) تقسیم می‌شود. سپس در هر cell، گرادیان شدت روشنایی برای هر پیکسل محاسبه می‌گردد. این گرادیان‌ها نشان می‌دهند که در آن نقطه تغییرات تصویر به چه سمتی است (مثلاً لبه‌ی عمودی یا افقی). بعد از محاسبه گرادیان‌ها، یک هیستوگرام از جهت‌ها ساخته می‌شود که نشان می‌دهد در آن cell کدام جهت لبه‌ها غالب است. برای مقاوم‌تر کردن ویژگی‌ها در برابر تغییر نور، چند cell در قالب یک block نرمال‌سازی می‌شوند تا اثر شدت نور کمتر شود.



در نهایت، همه‌ی این هیستوگرام‌های محلی کنار هم قرار گرفته و یک توصیف‌گر (descriptor) بلند برای کل تصویر یا ناحیه‌ی موردنظر ساخته می‌شود. این توصیف‌گر می‌تواند به یک طبقه‌بند مثلاً SVM داده شود تا تشخیص دهد آیا در تصویر یک شیء خاص (مثلاً انسان) وجود دارد یا خیر. یکی از مزایای HOG این است که نسبت به تغییرات جزئی نور و موقعیت مقاوم است و ویژگی‌های قوی و کم‌خطایی ارائه می‌دهد.

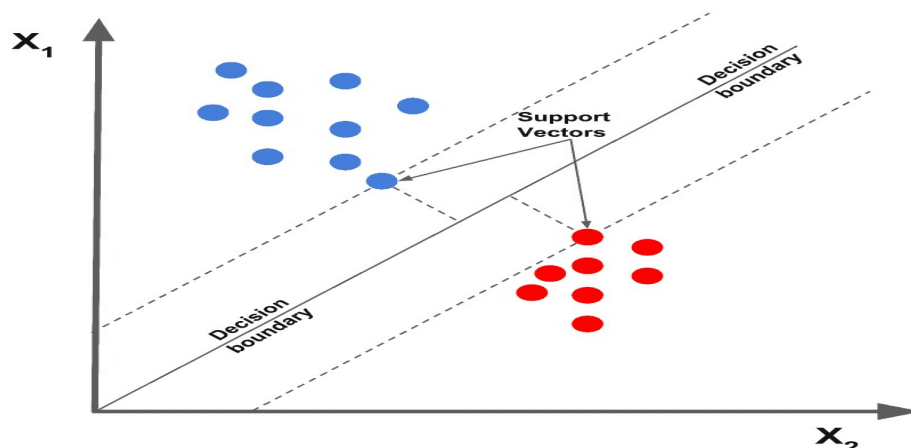
الگوریتم HOG در بسیاری از کاربردها استفاده شده است، اما معروفترین کاربرد آن تشخیص عابر پیاده در تصاویر ویدیویی یا سیستمهای رانندگی خودکار است. با این حال، به دلیل پیشرفت یادگیری عمیق، امروزه اغلب جای خود را به شبکههای عصبی داده است؛ با این وجود هنوز هم به عنوان یک الگوریتم ساده، سریع و قابل فهم در بینایی ماشین ارزش بالایی دارد.

SVM :

الگوریتم SVM یا **Support Vector Machine** یکی از روشهای قدرتمند یادگیری ماشین برای طبقه بندی (Classification) و گاهی رگرسیون (Regression) است. ایده اصلی آن بر این اساس است که داده ها را در یک فضای n بعدی تصور کرده و سپس یک مرز تصمیم گیری (Decision Boundary) پیدا کند که بتواند دسته های مختلف داده ها را از هم جدا کند. این مرز در حالت دوبعدی یک خط، در حالت سه بعدی یک صفحه و در حالت های بالاتر یک ابرصفحه (Hyperplane) است.

ویژگی کلیدی SVM در این است که بهترین مرز را طوری انتخاب می کند که بیشترین فاصله ممکن از نزدیک ترین نقاط هر دسته داشته باشد. این نقاط خاص که مرز دقیقاً با آنها تعیین می شود، **Support Vectors** نام دارند. به عبارت دیگر، الگوریتم سعی می کند خط یا صفحه ای را پیدا کند که نه تنها داده ها را جدا کند، بلکه با حاشیه ی (Margin) بیشترین فاصله، جداسازی را انجام دهد. این خاصیت باعث می شود SVM در برابر داده های جدید مقاومت بیشتری داشته باشد و تعمیم بهتری بدهد.

الگوریتم SVM در بسیاری از زمینه ها استفاده شده است؛ مانند تشخیص چهره، دسته بندی متن، تشخیص سرطان بر اساس داده های پزشکی و حتی در بینایی ماشین همراه با ویژگی هایی مثل HOG برای شناسایی عابر پیاده. هرچند امروزه شبکه های عصبی عمیق در بسیاری از مسائل جایگزین آن شده اند، اما به دلیل سادگی، دقت بالا در داده های کوچک و توانایی خوب در مسائل با ابعاد زیاد، SVM همچنان یکی از روش های پایه ای و مهم در یادگیری ماشین محسوب می شود.



کد تشخیص عابر پیاده:

```
import cv2
import matplotlib.pyplot as plt

# Create a HOGDescriptor object
hog = cv2.HOGDescriptor()

# Initialize the People Detector
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

# Load an image
image = cv2.imread('images/pedestrian.jpg')
(bounding_boxes, weights) = hog.detectMultiScale(image,
                                                  winStride=(4, 4),
                                                  scale=1.05)

# Draw bounding boxes on the image
for (x, y, w, h) in bounding_boxes:
    cv2.rectangle(image,
                  (x, y),
                  (x + w, y + h),
                  (0, 255, 255),
                  4)

plt.imshow(image[...,::-1])
```

در این کد:

```
hog = cv2.HOGDescriptor()
```

Histogram of Oriented Gradients یک شیء HOGDescriptor ایجاد می‌شود که قابلیت استخراج ویژگی‌های را از تصویر دارد. HOG ویژگی‌های محلی لبه‌ها و جهت‌های گرادیان تصویر را محاسبه می‌کند که برای شناسایی اشیاء مانند انسان مناسب است.

```
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
```

در این مرحله یک طبقه‌بند SVM پیش‌آموزش‌دیده برای افراد به HOG متصل می‌شود. این طبقه‌بند قادر است با استفاده از ویژگی‌های HOG نقاطی را که احتمالاً انسان هستند، شناسایی کند.

[illegible]

• تصویر ورودی خوانده می شود.

• تابع `detectMultiScale` برای جستجوی چندمقیاسی افراد روی تصویر استفاده می شود.

- `winStride=(4,4)` مشخص می کند که پنجره لغزان (Sliding Window) با چه فاصله ای حرکت کند.
- `scale=1.05` مشخص می کند اندازه پنجره در هر مرحله چند درصد کوچک یا بزرگ شود تا افراد در اندازه های مختلف شناسایی شوند.

• خروجی:

- `bounding_boxes` مختصات مستطیل های اطراف افراد شناسایی شده.
- `Weights` درجه اطمینان هر شناسایی.

پارامترهای `winStride` و `scale` در تابع `hog.detectMultiScale` نقش مهمی در دقت و سرعت تشخیص افراد دارند:

`winStride`

- تعریف: فاصله ای که پنجره ی لغزان (Sliding Window) در طول تصویر حرکت می کند.
- مثال `winStride=(4,4)`: یعنی پنجره در هر حرکت ۴ پیکسل افقی و ۴ پیکسل عمودی جابجا می شود.
- تأثیر بر دقت:
 - مقدار کوچک تر (مثلاً $(2,2)$) پنجره با گام های کوچک تر حرکت می کند و تصویر دقیق تر اسکن می شود → شانس شناسایی افراد کوچک یا نزدیک لبه ها بیشتر می شود.
 - مقدار بزرگ تر (مثلاً $(8,8)$ یا $(16,16)$) پنجره با گام بزرگ تر حرکت می کند → احتمال از دست دادن بعضی افراد افزایش می یابد.

• تأثیر بر سرعت:

- مقدار کوچک تر → تعداد پنجره های بررسی شده بیشتر → پردازش کندتر.
- مقدار بزرگ تر → تعداد پنجره ها کمتر → پردازش سریع تر.

`scale`

- تعریف: میزان تغییر اندازه پنجره در هر مرحله چندمقیاسی (Multi-scale) تشخیص.

- مثال $scale=1.05$: یعنی در هر مرحله پنجره حدود ۵٪ کوچک تر می شود تا افراد با اندازه های متفاوت شناسایی شوند.

- **تأثیر بر دقت:**

- مقدار کوچک تر (مثلاً ۱.۰۵ یا ۱.۰۲) → الگوریتم پنجره ها را با تغییر اندازه های ریز بررسی می کند → دقت در شناسایی افراد کوچک و بزرگ بیشتر می شود.
- مقدار بزرگ تر (مثلاً ۱.۲) → بررسی تغییر اندازه ها با گام های بزرگ تر → بعضی افراد با اندازه های متفاوت ممکن است شناسایی نشوند.

- **تأثیر بر سرعت:**

- مقدار کوچک تر → تعداد اندازه های بررسی شده بیشتر → پردازش کندتر.
- مقدار بزرگ تر → تعداد اندازه ها کمتر → پردازش سریع تر.