

PRÁCTICA 7: Detector de obxectos (HoG + SVM)

Xosé R. Fdez-Vidal
xose.vidal@usc.es

Grao de Robótica. EPSE de Lugo — 7 de novembro do 2022

1. Introducción

Os algoritmos sinxelos de clasificación de imaxes só poden darnos unha etiquetaxe e categorización global dunha imaxe. Non poden proporcionar etiquetas locais da imaxe e nin indicarnos a rexión onde están. Para este obxectivo precisamos un detector de obxectos (ver Fig. 1).



Figura 1: Os algoritmos básicos de clasificación de imaxes só poden informar sobre se unha imaxe contén ou non un obxecto específico. Non poden detallar exactamente onde está o obxecto na imaxe.

o contexto de detección de obxectos, o termo “obxecto” soa un pouco abstracto, é a razón é porque é así. Un obxecto pode ser unha cadeira, unha persoa ou mesmo un vaso de cervexa. En xeral, calquera entidade física cunha estrutura semirrígida (o que significa que o obxecto non é demasiado deformable e pode alterar drasticamente a súa forma) pode considerarse un “obxecto”. Aquí van algúns exemplos do CALTECH-101 (Fig. 2), un popular conxunto de datos de referencia de detección de obxectos de 101 categorías:

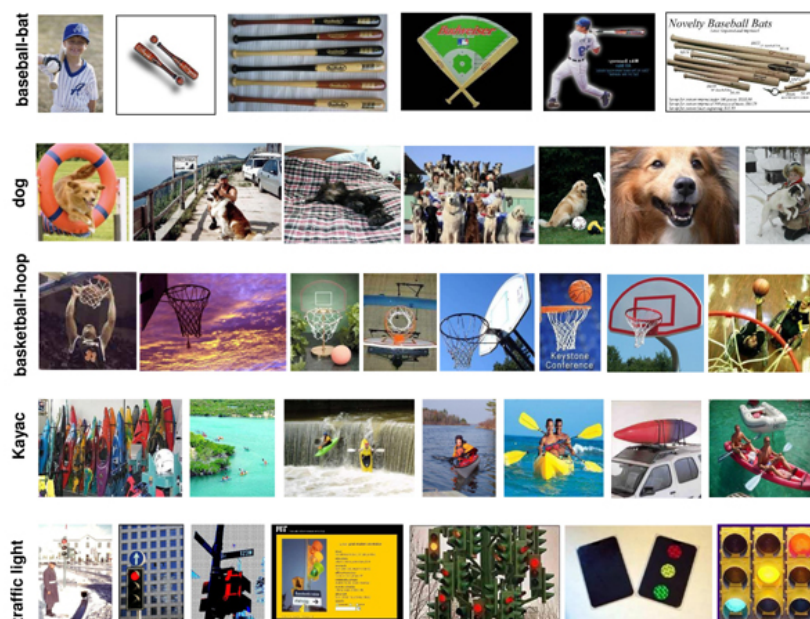


Figura 2: Exemplos de imaxes do conxunto de datos CALTECH-101.

O recoñecemento de obxectos é difícil pola mesma razón pola que a clasificación de imaxes é difícil. Os obxectos do mundo real poden presentar variacións substanciais no punto de vista, escala, deformación, oclusión, iluminación, desorde de fondo e variación intra-clase (tal e como demostra a Fig 2 anterior). Un bo detector de obxectos debe ser resistente aos cambios nestas propiedades e aínda así poder detectar a presenza do obxecto, mesmo en circunstancias que non sexan idóneas.

A detección de obxectos úsase na túa vida cotiá nun cantidade de tarefas amplo, que apuntamos a continuación:

- detección da presenza de caras en imaxes (procesos de etiquetado nas redes sociais, cámaras de autoenfoque, etc.),
- en sistemas de seguridade onde rastreamos persoas en diversas fontes de vídeo para supervisar a súa actividade (eventos culturais e deportivos, etc),
- en garaxes automatizados de aparcamento de vehículos para detectar se unha praza de aparcamento está libre ou non,
- no campo da busca visual para identificar, por exemplo, prendas de roupa que visten (camisa, jeans, etc.) os viandantes que se paran diante dun escaparate.
- en conducción asistida para avisar de sinais de tráfico importantes, peóns cruzando a rúa, etc.

1.1. localización, clasificación e avaliación

O título desta sección enumera tres pasos básicos dun *sistema de detección de obxectos*. Os obxectos candidatos están localizados dentro dunha caixa delimitadora rectangular, un cadro delimitador (Fig. 3) que é un exemplo especial dunha rexión de interese (RoI).

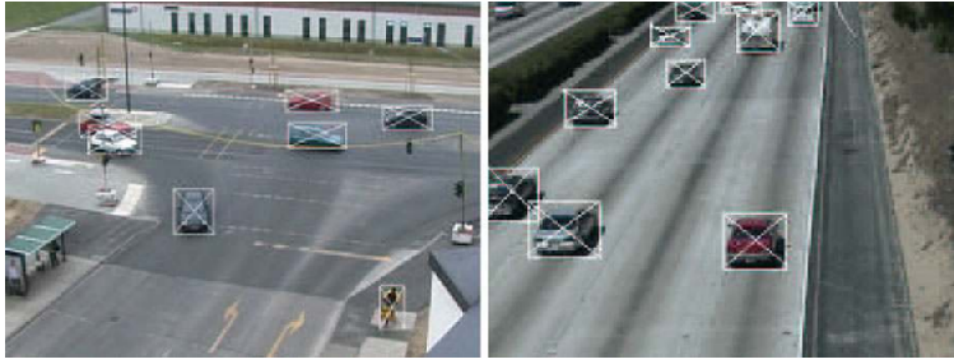


Figura 3: Exemplos de caixas delimitadoras na detección de obxectos.

Estas rexións localizadas son candidatas a obxectos e mediante un clasificador son asociados a obxecto detectado ou a candidato rexeitado. Por último, os resultados da clasificación deben ser avaliados dentro do sistema. Un verdadeiro positivo, tamén chamado hit ou detección, é un obxecto detectado correctamente. Un falso positivo, tamén chamado erro ou falsa detección, prodúcese se detectamos un obxecto onde non hai. Un falso negativo denota un caso no que perdemos un obxecto, e un verdadeiro negativo describe os casos nos que as rexións non obxecto están correctamente identificadas como rexións non-obxecto. A Fig. 4 contén un falso positivo (o cadrado máis grande) e dous falsos negativos (un home no medio e unha rapaza á dereita da imaxe). Unha cara de perfil (un caso na figura) non define unha cara no noso sistema (buscamos caras frontais).



Figura 4: Detección de caras cun falso positivos e dous falsos negativos.

1.2. Descritores, clasificadores e aprendizaxe

A clasificación defínese polo proceso de asociar a unha mostra a unha clase entre un conxunto disxunto de \mathbb{R}^n para un valor dado $n > 0$. Noutras palabras, as clases definen unha partición do espazo \mathbb{R}^n .

Un descritores, $x = (x_1, \dots, x_n)$ é un punto nun espazo \mathbb{R}^n multidimensional, chamado espazo de características e representa unha propiedade medida ou achada nun orde determinado (por exemplo, un descritor SIFT ten unha lonxitude $n = 128$).

Un clasificador asigna a cada descritor o número da clase¹. Este proceso, ten dúas partes diferenciadas: a parte de adestramento na que o clasificador aprende os parámetros internos a partir dun conxunto de descritores, x_1, \dots, x_m , xa etiquetados, e unha segunda fase de test ou produción, a partir do aprendizaxe asigna clases aos descritores sobre unha nova imaxe ou vídeo.

A aprendizaxe é o proceso de cálculo dos parámetros (adestramento) internos dun clasificador a partir dun conxunto de mostras representativas das clases a recoñecer. A clasificación é o proceso da aplicación do clasificador adestrado. Durante este último proceso, poderemos atopar algún comportamentos non desexados no clasificador o cal no levará a unha nova fase de adestramento.

No aprendizaxe supervisado un humano asigna manualmente os números aos que pertence cada descritor da nosa base de datos de adestramento. Sen embargo, no aprendizaxe non supervisado non dispoñemos dun coñecemento *a priori* acerca da clase a que pertence cada descritor. Polo tanto, o obxectivo é producir agrupamentos no espazo de características mediante un algoritmo de *clustering* a un conxunto dado de descritores para identificar a separación de \mathbb{R}^n en clases.

Na Fig. 5) móstranse ventás e parches de imaxes con etiquetas positivas e negativas.

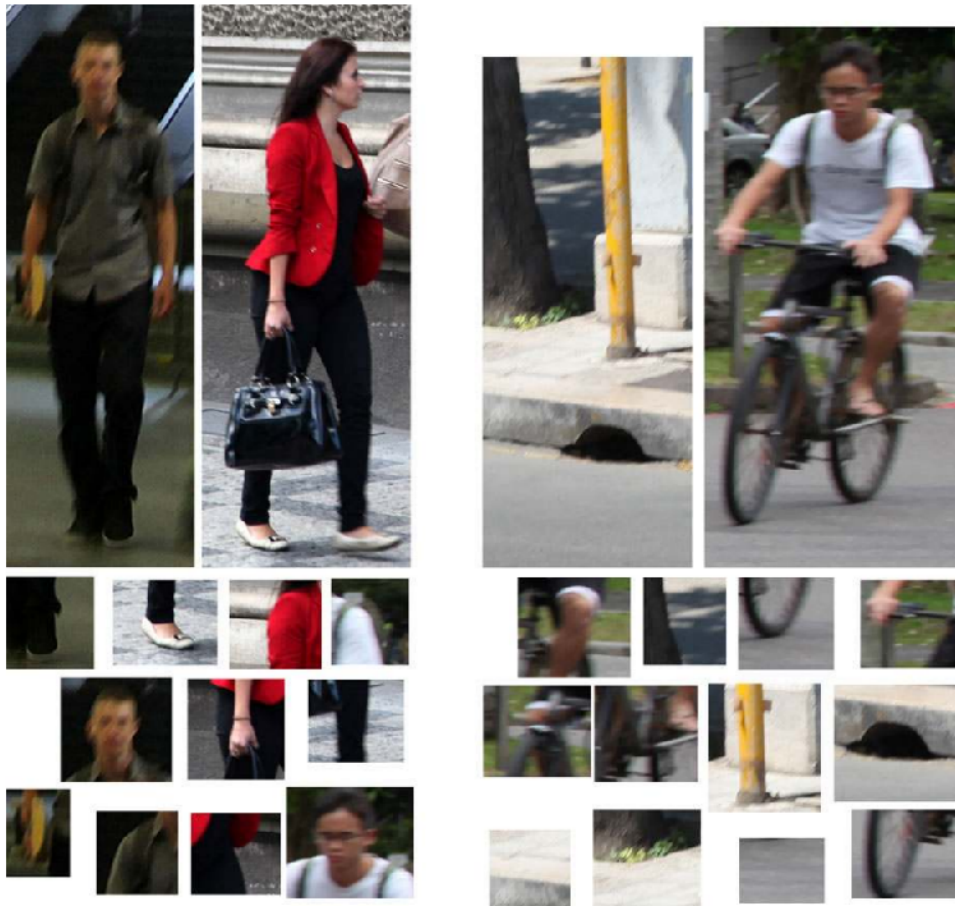


Figura 5: Para un sistema de detección de peóns. Arriba esquerda: dúas mostras de caixas delimitantes positivas. Arriba dereita: dúas mostras de caixas delimitantes negativas. Abaixo esquerda: parches positivos que probablemente pertencen a peóns. Abaixo dereita: parches negativos para peóns.

¹Un clasificador (xeral) asigna os números de clase 1, 2, ..., k para $k > 1$ clases e 0 para descoñecido

2. Formulación do problema

Neste traballo perseguimos un detector de obxectos que, mediante a aprendizaxe automática, se adestre un clasificador que se pode usar para categorizar un obxecto que desexe o estudante. Concretamente:

- Como empregar a base de datos CALTECH-101 para o recoñecemento de obxectos.
- Explorar os datos da base de datos, coñecer a súa organización e como estruturalos para o adestramento do recoñecedor de obxectos.
- Como combinar un descriptor e un clasificador e sintonizar os seus parámetros internos máis relevantes.
- Como avaliar o rendemento do detector de obxectos.

3. Base de datos CALTECH-101

Introducido por Fei-Fei et al. en 2004, a base de datos CALTECH-101 é un conxunto de imaxes de referencia moi popular para a detección de obxectos e foi utilizado por moitos investigadores, académicos e desenvolvedores de visión artificial para avaliar os seus algoritmos de detección de obxectos.

O conxunto de datos inclúe 101 categorías, que abarcan unha ampla gama de obxectos, incluíndo elefantes, bicicletas, balóns de fútbol e mesmo cerebros humanos, só por citar algúns.

Cando descargues o conxunto de datos CALTECH-101, notarás que inclúe dous cartafoles, un de imaxes e outro de anotacións. Para cada imaxe do conxunto de datos, ofrécese un cadro delimitador asociado (é dicir, coordenadas (x, y) do obxecto) segundo se amosa na Fig. 6. Lembremos que o noso obxectivo é tomar tanto as imaxes como os cadros delimitadores (é dicir, as anotacións) e adestrar un clasificador para detectar a presenza dun determinado obxecto nunha imaxe.



Figura 6: Dada unha imaxe que contén un sinal de stop (esquerda) e as anotacións (x, y) do sinal de stop na imaxe (centro), o noso obxectivo é adestrar a un clasificador para detectar o sinal de stop na imaxe (dereita).

4. Descriptor: Histograma de Gradientes Orientados (HoG)

O histograma de gradientes orientados (HoG) [1], é unha forma común de obter un descriptor para un cadro delimitador para un obxecto candidato. Por exemplo, unha xanela do tamaño do cadro delimitador esperado pode moverse a través dunha imaxe, e a exploración detense en posibles obxectos candidatos, posiblemente guiada polos resultados proporcionados pola visión estereoscópica (por exemplo, o tamaño esperado dun obxecto a unha distancia determinada) ou por un detector de características. Despois de que se identificou unha caixa delimitadora potencial, comeza un proceso para o cálculo do descriptor, e aquí explicamos o caso dos descritores HoG. Na Fig. 7 ilustra unha subdivisión dunha caixa delimitadora (dun peón) en bloques máis grandes e celas máis pequenas para calcular o HoG

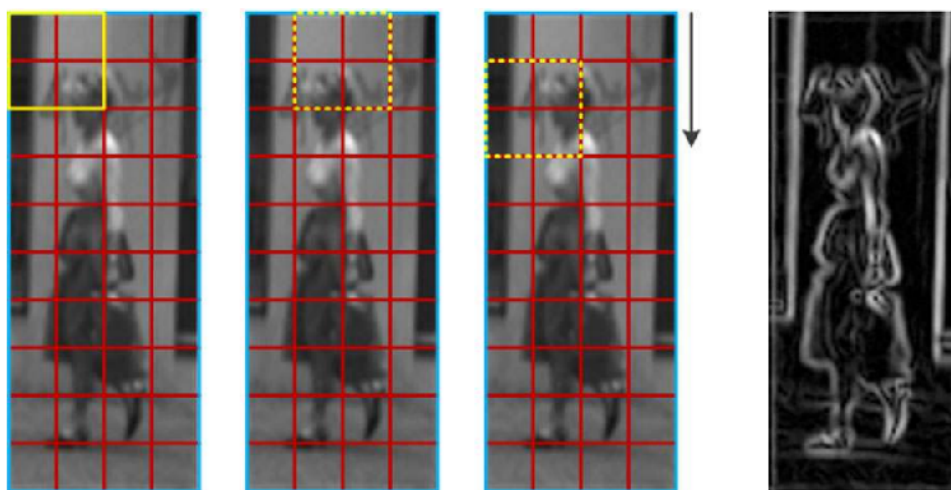


Figura 7: Bloques e celas ao calcular un descriptor HoG nun cadro delimitador (o rectángulo azul ángulo). Os rectángulos amarelos sólidos ou discontinuos denotan bloques, que se subdividen en rectángulos vermellos (é dicir, as celas). As tres fiestras da esquerda mostran como un bloque se move de esquerda a dereita, de arriba abaixo, a través dun cadro delimitador, ao xerar o descriptor. A xanela da dereita ilustra a magnitude do gradiente.

Esbozamos brevemente os pasos a dar para achar para calcular descritores de HoG a partir dunha caixa delimitadora seleccionada:

- Preprocesado. Aplicar unha normalización de intensidade e un filtro de suavizado na xanela da imaxe I .
- Calcula o gradiente estimando as derivadas direccionais x e y . Obtemos a magnitude (I_m) e a orientación (I_a) do gradiente.
- Mostraxe (binning) espacial:
 - (a) Agrupar píxeles en celas non superpostas (por exemplo, 8×8); ver Fig. 7, esquerda.
 - (b) Usar os mapas I_m e I_a para acumular valores de magnitude en bins de dirección (por exemplo, nove contedores a intervalos de 20 degree cada un, para cubrir un rango completo de 180 degree) para obter un vector de votación (por exemplo, de lonxitude 9) para cada cela; ver Fig. 8. Imaxes integrais poden utilizarse para un cálculo eficiente do tempo destas sumas.
- Normalizar os valores de votación para xerar un descriptor:
 - (a) Agrupamos celas (por exemplo, 2×2) nun bloque.
 - (b) Normalizamos os vectores de voto sobre cada bloque e combinámoslos nun vector de bloquer (por exemplo, catro vectores de celas nun vector bloque de lonxitude 36).
- Apilamos todos os vectores de bloques consecutivamente; isto produce o descriptor HoG final. Por exemplo, se un bloque consta de catro celas e o tamaño da caixa delimitadora é de 64×128 , o descriptor HoG ten 3,780 elementos; pode ser conveniente reorganizar este vector descriptor nunha matriz B (por exemplo, de tamaño 420×9). Caixas delimitadoras utilizadas para adestrar ou aplicar un clasificador para a detección de peóns adoitan a normalizarse a un tamaño idéntico (por exemplo, 64×192 , ambos múltiplos de 32). Tamén se poden empregar distintos tamaños de cela (por exemplo, 8×8 , 16×16 e 32×32) para xerar os vectores descritores HoG.



Figura 8: A lonxitude dos vectores en nove direccións diferentes en cada cela representa a magnitude acumulada dos vectores de gradiente nunha desas nove direccións.

4.1. Configuración do descritor HoG

Nesta sección imos a dar uns consellos de como configurar algunha das opcións do noso detector de características. Entre elas, as dimensións do descritor de HOG axeitadas xunto co tamaño da xanela deslizante.

- Elixir as dimensións da ventá deslizante e os parámetros HOG axeitados. Supoñamos que estou interesado en detectar vistas laterais de coches nunha imaxe e ao analizar os nosos datos de adestramento descubrimos que as rexións anotadas polo humanos (é dicir, as caixas delimitadoras do coche) teñen un ancho medio de 184 píxeles, unha altura media de 62 píxeles (unha relación de aspecto de 2,97), o que implica que o ancho da vista lateral dun coche é aproximadamente 3 veces maior que a altura. Polo tanto, necesitamos ter en conta estas dimensións e usalas para definir un tamaño de ventá deslizante axeitado, xunto cos parámetros do descritor HOG asociados.

Definir as dimensións da ventá deslizante e os parámetros HOG é un pouco unha “arte escura” que require práctica, persistencia e observación do que funciona (e máis importante, o que non). Dito isto, un bó punto de partida é tomar a anchura e a altura medias (das anotacións na túa base de adestramento) e dividilas entre dous, respectivamente. Isto déixanos un ancho de 92 px e unha altura de 31 px, pero cal é razón disto?. Principalmente dúas:

- **Razón #1:** os vectores de características HOG poden chegar a ser extremadamente grandes a medida que aumenta o tamaño da xanela, polo que é importante atopar un equilibrio entre o tamaño da xanela, a lonxitude do vector de características e a discriminabilidade do vector de características.
 - **Razón #2:** para axudar na detección multiescala. Se o tamaño da miña fiestra é demasiado grande, pode perder facilmente obxectos máis pequenos. Pola contra, se o tamaño da miña xanela é demasiado pequeno, desperdicio miles de millóns de ciclos de CPU en ROI² demasiado pequenas para conter un obxecto de interese; isto tamén implica que un obxecto só se pode atopar nas escalas máis pequenas dunha pirámide de imaxes (que, de novo, leva a ciclos de CPU desperdiciados). Ao dividir entre dous, supón un bo equilibrio entre un tamaño razoable da xanela e a lonxitude do vector de características, mentres (1) aseguro que non estou avaliando descoidadamente multitude de ROI e (2) confío na pirámide de imaxes para capturar obxectos que sexan máis pequenos ou máis grandes que as dimensións da miña fiestra.
- Agora que temos axustado o ancho e un alto (92px e 31px) da nosa xanela (**winSize**), pode pasar a definir os seguintes parámetros HOG. Os dous parámetros máis importantes a considerar no

²acrónimo de rexións de interese

noso descritor HOG son o número de píxeles por cela (**cellSize**) e o numero de celas por bloque (**blockSize**)³:

- As dimensións da nosas fiestras deslizante deben ser divisibles por estes valores. En caso contrario, o noso descritor HOG non “encaixará” no tamaño da xanela. Unha vez máis, axustar os parámetros HOG é un pouco de alquimia que adquirirás coa práctica, pero un bo punto de partida é elixir a tupla **cellSize** con valores múltiplos de catro. En canto as celas por bloque elíxense como múltiplos (habitualmente) do conxunto $\{1, 2, 3\}$ pero tes liberdade de elección só debes ter en conta a relación entre o tamaño da xanela (**winSize**) e **cellSize** para que todo encaixe. Volvendo ao noso exemplo de detección de coches, unha boa elección inicial sería **cellSize** = (4, 4) e **blockSize** = (2, 2).
- As elección feitas no punto anterior adoitan ser opcións comúns de parámetros. Isto implica agora que tanto o ancho da miña xanela deslizante como a altura da xanela deslizante deben ser divisibles por **catro**. Para logralo, redondeamos 92px e 31 px ao múltiplo de catro máis próximo, o que me dá 96px e 32px, respectivamente. Examinando a relación de aspecto, vemos que $96/32 = 3$, que está convenientemente preto da relación de aspecto das dimensións medias dos nosos datos de adestramento de coches (2, 97).

5. Clasificador Support Vector Machine (SVM)

A Support Vector Machine (SVM)[2] é un clasificador discriminativo formalmente definido por un hiperplano separador. Noutras palabras, dados os datos de adestramento etiquetados (aprendizaxe supervisada), o algoritmo produce un hiperplano óptimo que categoriza novos exemplos.

En que sentido se obtén o hiperplano óptimo? Consideremos o seguinte problema sinxelo: para un conxunto linealmente separable de puntos 2D que pertencen a unha das dúas clases, atopa unha recta de separación.

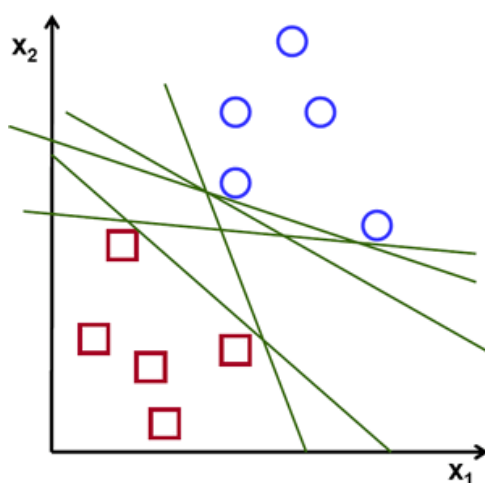


Figura 9: Problema simple de clasificación linearmente separable.

Na Fig. 9 podes ver que existen varias liñas que ofrecen unha solución ao problema. Algunha delas é mellor que as outras? Podemos definir intuitivamente un criterio para estimar o valor das liñas: unha liña é mala se pasa demasiado preto dos puntos porque será sensible ao ruído e non se xeneralizará correctamente. Polo tanto, o noso obxectivo debería ser atopar a liña que pase o máis lonxe posible de todos os puntos. Para unha maior profundidade, revisa o notebook 1-Tecnicas_reconecemento.ipynb) do tema 7 onde se explica o proceso de optimización.

³notación dos parámetros de HoG de OpenCV

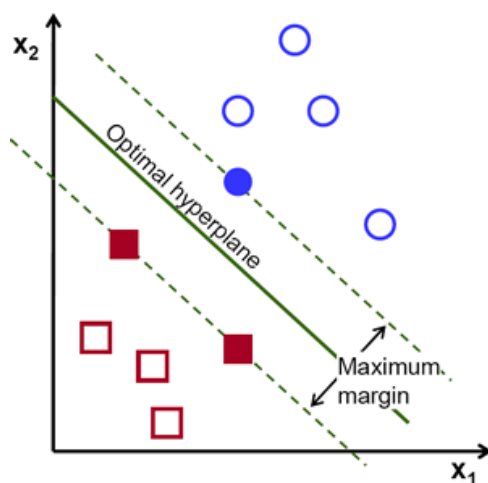


Figura 10: Hiperplano óptimo atopado no adestramento.

Fonte orixinal: [nesta URL](#)

6. Pasos para un adestramento do clasificador

Adestrar un detector de obxectos é moi similar a adestrar a un clasificador de imaxes para categorizar os contidos globais dunha imaxe ou un clasificador de contornos. Non obstante, nesta ocasión preséntannos non só as etiquetas das imaxes, senón tamén as anotacións correspondentes ao cadro delimitador que rodea cada obxecto. Tomaremos estes cadros delimitadores, extraeremos características deles que despois utilizaremos para adestrar o noso detector de obxectos.

Nesta sección, imos a abordar aspectos prácticos para levar a nosa misión de adestramento (seis pasos) dun detector de obxectos HoG+SVM:

- **Paso 1:** Selecciona P mostras positivas da base de datos dos obxectos que queres detectar (colocas nun cartafol chamado, p. ex., posPatches) e extrae descritores HOG destas mostras: (1) extraemos o cadro delimitador do obxecto, seguido de (2) calcularemos as características HOG sobre esta ROI. Estas funcións HOG servirían como exemplos positivos (é dicir, os exemplos que conteñen o obxecto que queremos detectar) que posteriormente introduciremos no noso clasificador.

- **Paso 2:** Selecciona N mostras negativas da base de datos de adestramento (colocas nun cartafol chamado, p. ex., negPatches). Estas imaxes non conteñen aos obxectos que queres detectar e extrae tamén os descritores HOG destas mostras. Na práctica $N \gg P$.

Este proceso pode ser automatizado. Por exemplo, se queremos detectar unha sinal de stop, podemos coller ventás aleatoriamente de imaxes dende unha base de datos, por exemplo de paisaxes naturais dado que é moi improbable que conteñan unha sinal de stop. Estas funcións HOG de cada parche seleccionado, serán os nosos exemplos negativos (é dicir, exemplos que non conteñen o obxecto que queremos detectar) que usaremos para adestrar o noso SVM lineal.

- **Paso 3:** Adestra o teu clasificador sobre as mostras positivas e negativas que conformaches.
- **Paso 4:** Minería de negativos difíciles (Mining Hard Negative). Unha vez adestrado o modelo sobre as mostras positivas e negativas é probable que aínda produce bastantes **falsos positivos**. Unha das causas deste problema é que non fixemos minería negativa que consiste en readestrar de novo o modelo sobre as rexións onde estamos a detectar os falsos positivos (son difícil predicir estes parches correctamente) que son difíciles para o modelo.

Para construír estes parches, cando executamos o detector en imaxes negativas completas, anotamos todos aqueles parches que se prevén como positivos (é dicir, etiqueta prevista = 1) nun directorio (chamémoslle falsePosPatches). Asegúrate de cambiar o tamaño deste parche ás mensuras da ventá deslizante do detector (**winSize**) antes de gravalos no directorio falsePosPatches.

Agora adestramos de novo o modelo coas imaxes positivas situadas en posPatches e coas imaxes negativas situadas nos cartafolios negPatches + falsePosPatches.

- **Paso 4:** Unha vez adestrado o teu clasificador, pódese aplicar ao conxunto de datos de test (non usados no adestramento). De novo, para cada imaxe test sabemos a súa anotación (clase e posición) o que nos permitirá achar os parámetros de rendemento do noso detector (ver sección 7).

7. Medidas de avaliación dos clasificadores

Como dixemos, a clasificación é un tipo de problema de aprendizaxe automática supervisado (no noso caso) onde o obxectivo é predicir, para unha ou varias observacións, a categoría ou clase á que pertencen.

Un elemento importante de calquera fluxo de traballo de aprendizaxe automática é a *avaliación do rendemento* do modelo. Este é o proceso no que usamos o modelo adestrado para facer predicións sobre datos etiquetados non vistos anteriormente. No caso da clasificación, entón avaliamos cantas destas predicións acerta o modelo.

En problemas de clasificación reais, normalmente é imposible que un modelo sexa correcto ao 100 %. Á hora de avaliar un modelo é, polo tanto, útil saber, non só o errado que está o modelo, senón de que xeito erra⁴.

Por exemplo, se tratamos de predicir se un tumor é benigno ou maligno, pode ser preferible que o modelo erre en predicir incorrectamente que un tumor é maligno (nun pequeno número de casos) a que prediga que é benigno cando cando non o é con graves consecuencias para o paciente.

Neste caso, optimizaríamos o modelo para que funcione mellor para determinados resultados e, polo tanto, podemos utilizar diferentes métricas para seleccionar o modelo final a empregar. Como consecuencia destas compensacións ao seleccionar un clasificador, hai unha variedade de métricas que debe utilizar para optimizar un modelo para o seu caso de uso específico.

A continuación, imos expoñer as oito métricas de rendemento diferentes que se poden usar para avaliar un clasificador.

- **Matriz de confusión** é unha ferramenta extremadamente útil para observar de que xeito o modelo é incorrecto (ou correcto!). É unha matriz que compara o número de predicións para cada clase que son correctas e as incorrectas.

Nunha matriz de confusión, hai 4 números aos que prestar atención.

Verdadeiros positivos (TP): o número de observacións positivas que o modelo predixo correctamente como positivas.

Falsos positivos (FP): o número de observacións negativas que o modelo predixo incorrectamente como positivas.

Verdadeiros negativos (TN): o número de observacións negativas que o modelo predixo correctamente como negativas.

Falsos negativos (FN): o número de observacións positivas que o modelo predixo incorrectamente como negativas.

A Fig. 11 mostra unha matriz de confusión para un clasificador. Usando isto podemos entender o seguinte:

⁴“Todos os modelos son incorrectos, pero algúns son útiles”. George Box

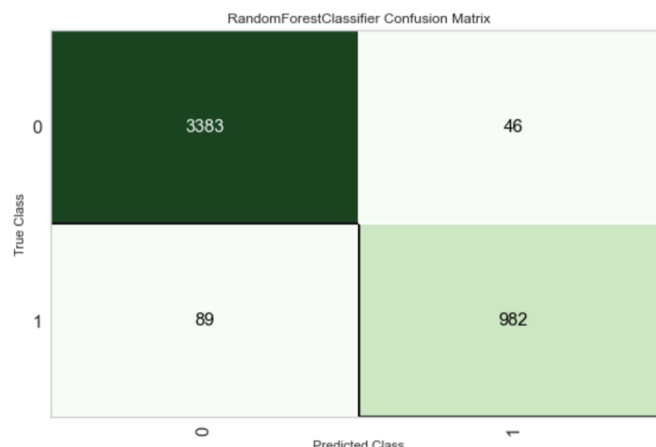


Figura 11: Mariz de confusión.

O modelo predixo correctamente 3383 mostras negativas pero predixo incorrectamente 46 como positivas. O modelo predixo correctamente 982 observacións positivas pero predixo incorrectamente 89 como negativas. Desta matriz de confusión podemos ver que a mostra de datos está desequilibrada, tendo a clase negativa un maior volume de observacións.

- **Exactitude (Accuracy):** a exactitude global dun modelo é simplemente o número de predicións correctas dividido entre p número total de predicións. Unha puntuación deste medida dará un valor entre 0 e 1, onde o valor de 1 indicaría un modelo perfecto.

$$ACC = \frac{\#TP + \#TN}{\#TP + \#FP + \#FN + \#TN}. \quad (1)$$

Se volvemos ao exemplo do cancro. Imaxina que temos un conxunto de datos onde só o 1 % das mostras son cancerixenas. Un clasificador que simplemente prevé todos os resultados como benignos conseguiría unha puntuación de precisión do 99 %. Porén, este modelo sería, de feito, inútil e perigoso xa que nunca detectaría unha observación cancerosa.

- **Precisión (precision):** mide o bo que é o modelo para identificar correctamente a clase positiva. Noutras palabras, de todas as predicións para a clase positiva, cantas eran realmente correctas? Usando só esta métrica para optimizar un modelo estaríanos *minimizando os falsos positivos*. Isto podería ser pouco útil para diagnosticar o cancro xa que teríamos pouca comprensión das observacións positivas que non se fan.

$$Pr = \frac{\#TP}{\#TP + \#FP}. \quad (2)$$

- **Sensibilidade (recall) ou True positive rate:** dinos o bo que é o modelo para predicir correctamente *todas as observacións positivas* do conxunto de datos. Non obstante, non inclúe información sobre os falsos positivos polo que sería útil no exemplo do cancro.

Normalmente, a precisión e sensibilidade obsérvanse xuntos construíndo unha curva de precisión-rememoración. Isto pode axudar a visualizar as compensacións entre as dúas métricas en diferentes limiares.

$$Re = \frac{\#TP}{\#TP + \#FN}. \quad (3)$$

- **F1-score:** esta é unha media harmónica da precisión e recall. O valor de F1 será un número entre 0 e 1. Se o valor é 1, indica unha precisión e recall perfectas. Se a puntuación de F1 é 0, isto significa que a precisión ou o recall son 0.

$$F_1 = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \quad (4)$$



Que medidas empregar no noso caso? Neste traballo construiremos a matriz de confusión para o número de clases que se usen no noso caso. A partir dela, determinaremos a accuracy xunto coa precision e recall, e a media harmónica destas últimas magnitudes, a F_1 - *score*.

8. Tarefas

Os estudantes deberán construír un detector de obxectos sobre a base de datos CALTECH-101 ([URL de descarga](#)).

Tarefa 1 : Detector de obxectos baseado en HoG + SVM

As tarefas que os estudantes teñen que levar a cabo esta parte son:

- (a) Dentro da base CALTECH-101 elixe un obxecto que desexes recoñecer seleccionando os cartafolios coas imaxes e as anotacións correspondentes. Cada estudante debe elixir un obxecto distinto da base de datos (non valen peóns/persoas), en caso de non acordo na distribución do obxectos a recoñecer se adxudicará por orde alfabético do estudante e da carpeta. Non se valorarán dúas propostas sobre o mesmo obxecto.
- (b) Prepara os datos nas carpetas de positivos e negativos para o adestramento. Tamén programa o código para ler as anotacións e a división das imaxes en mostras de adestramento e test (cun porcentaxe dun 80 % para adestramento e un 20 % para test).
- (c) Como punto de apoio, no notebook 3_Object_Detection_con_HOG_SVM.ipynb do tema, tes un exemplo de como combinar estas técnicas para detectar peóns con HoG e SVM.
- (d) Realiza un bo adestramento, seguindo as indicacións aportadas neste documento e estima as medidas de rendemento.
- (e) Intenta modificar os parámetros do HoG e do SVM para lograr maiores rendementos. Comenta os resultados comparativamente para cada un dos intentos que realices (non máis de tres).
- (f) Proba a cambiar de clasificador por unha árbore de decisión ou un random forest. No cartafol de Código/Python/Clasificadores/dt_and_rf tes un programa que ilustra como aplicar estes clasificadores (implementados no paquete sklearn). Compara os resultados co mellor intento que logaches con SVM.
- (g) Opcional: cambia o detector HoG por outro descriptor invariante que vimos na clase e están implementados nos notebooks do tema 6 (ollo: pensa sobre que puntos os vas a achar!).

Material aportado: código python do detector de peóns baseado en HoG + SVM e do uso dos clasificadores de árbore de decisión e random forest. Asi mesmo, no tema 5 temos notebooks que ilustran o cálculo de novos descritores invariantes locais (SIFT e SURF). Tamén se aporta a ligazón de descarga da base de datos CALTECH-101.

9. Entrega

Puntos que debe cumprir a entrega do documento que subas ao Campus Virtual:

1. A práctica debe ser autocontida nun caderno de Jupyter coas visualizacións correspondentes.
2. Tabulación dos datos en formato entendible por un humano.
3. Explicación das túas achegas e análises correspondentes.

10. Rúbrica da práctica

- Preparación da base de adestramento e test e toda a loxística necesaria (parches positivos, negativos, minería de negativos difíciles, lectura anotacións, etc.) → **15 pts**
- Proceso de adestramento HoG + SVN → **25 pts**
- Comparativa de clasificadores → **20 pts**
- Substitución do descritor HoG por outra elección do estudante → **20 pts**
- Análise e presentación dos resultados → **20 pts**

Referencias

- [1] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [2] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, pp. 273–297, 1995.