

Documentació de la pràctica de reconeixement automàtic de senyals de trànsit

Laboratori de Visió per Computador

1r Quadrimestre - curs 2020/2021

Marc Domènech i Vila
Gerard Palomares Castells

Índice

Introducció	3
Descriptors	4
HOG	4
Color	4
Ratio	5
Cercles (Transformada de Hough)	5
Vertexos (Harris)	6
Classificadors	7
Experimentació	7
Característiques	7
Segmentació	8
Resultats (Matriu confusió)	8
Funcions i Llibreries usades	10
References	11

Introducció

L'objectiu de la pràctica consisteix en identificar correctament una senyal de trànsit d'entre una de les 43 senyals proposades:



Fig 1. Les 43 senyals que s'han de classificar

Per a aquest objectiu s'ha facilitat un conjunt d'imatges de senyals de trànsit correctament classificades. Aquestes imatges, però, no són perfectes, ja que estan extretes de casos reals (tenen il·luminació diferent, fons diferent, escala diferent, etc.).

L'objectiu per tant, es pot assolir complint aquestes etapes:

- Crear un vector de característiques per a cada imatge que modeli les característiques adients de la senyal.
- Entrenar un classificador perquè, donat un vector de característiques extret d'una imatge, classifiqui correctament la senyal de la imatge

Descriptors

En aquest apartat es detallen els diferents descriptors utilitzats per a crear el vector de característiques de cada imatge.

HOG

La característica que més bons resultats ens ha donat és l'histograma d'orientació de gradients (HOG). Aquest serveix per a descriure la forma de la imatge, interpretant-la com la distribució d'intensitat i la direcció dels gradients. Per aquesta raó hem cregut que era bona idea fer-los servir, ja que la forma és una característica bastant important. Per a dur-la a terme, es redimensiona cada imatge a la mida de la imatge més petita del dataset (així el HOG de totes les imatges té la mateixa mida), es computa el HOG i s'insereix al vector de característiques.



Fig 2. Exemple visual de HOG

Color

El color de les senyals és, a priori, una característica a tenir molt en compte a l'hora de classificar-les. Al llarg de la pràctica hem extret aquest descriptor de moltes maneres diferents. Primer vam intentar usar una funció que podem trobar al Mathworks Files Exchange anomenada `image2palette`. Aquesta funció utilitza k-means color clustering i L^*a^*b color space per retornar els k colors més usats (codificat en L^*a^*b), el seu nom, i el percentatge de representació del color. En el nostre cas vam fer proves per a $k = 4$. Vam intentar afegir al vector de característiques les components a i b dels 4 colors més importants i els seus

percentatges. En aquest cas, vam considerar que la il·luminació era un valor que no ajudava a diferenciar senyals (no era un element distintiu).

Veient que els resultats d'aquesta prova no van ser els esperats, vam optar per agafar els histogrames RGB de la imatge i usar-los com a descriptors. Aquests últims si van ajudar a classificar una mica millor els senyals.

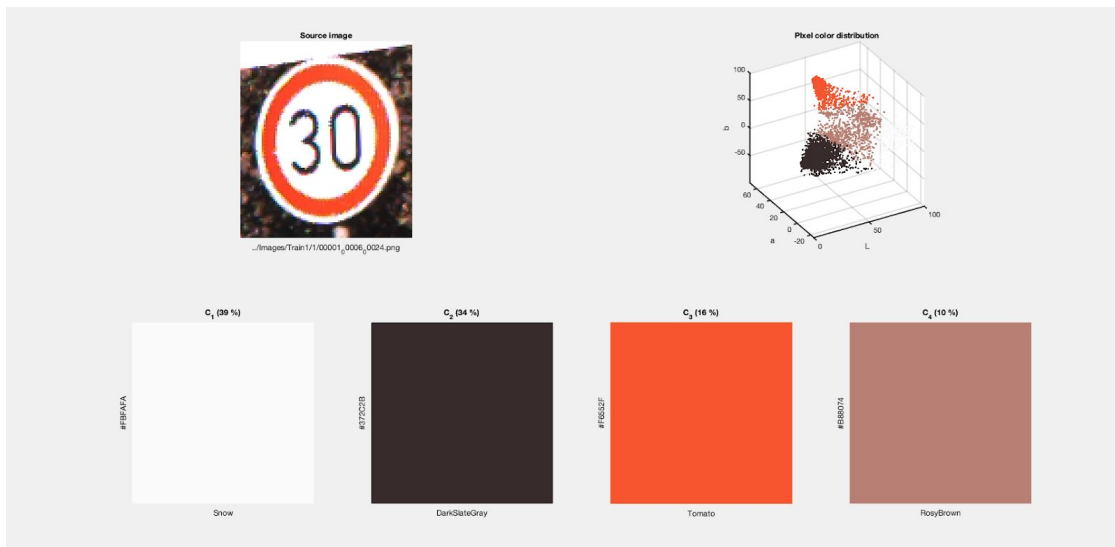


Fig 3. Exemple de la funció `image2palette`

Ratio

En aquest cas ens fixem en l'àrea del senyal de trànsit. Per calcular aquesta característica utilitzem la funció de matlab `activecountourn`. Aquesta funció separa el senyal del fons basant-se en un algorisme de region growing que utilitza la tècnica de contorns actius.

Amb la màscara resultant s'ha computat la ràtio entre píxels blancs (àrea) i negres (resta de la imatge), fent que el resultat es posi al vector de característiques.

No obstant això, com que no hem pogut aconseguir una segmentació realment eficient per a totes les imatges (aquesta es comporta molt diferent en funció del tipus d'imatge) i el fet que la funció `activecountourn` fa que l'execució sigui molt lenta, s'ha decidit no afegir aquesta característica.

Cercles (Transformada de Hough)

Si ens fixem en les senyals de trànsit, ràpidament arribarem a la conclusió de que la forma de la senyal és un tret força característic per distingir-la de la resta. Per aquest motiu hem volgut detectar cercles en la imatge per ajudar-nos a diferenciar les senyals circulars de la resta. Per

detectar aquests cercles hem fet servir la funció integrada en matlab `imfindcircles()`. Aquesta funció aplica la transformada de Hough per a detectar els cercles. En cas de detectar un cercle en l'imatge s'afegeix un 1 al vector de features, altrament s'afegeix un 0.



Fig 4. Aplicació de la transformada de Hough en una imatge

Vertexos (Harris)

Finalment una de les últimes característiques que vam pensar que podia ser d'utilitat, era la detecció de vèrtexs mitjançant l'algorisme de Harris. Per a calcular aquest descriptor el que fèiem era calcular quants vèrtexs s'hi detectaven. Donat que la segmentació de la imatge no ha estat gaire bona, no ha estat possible fer-los servir, ja que aleshores el soroll del fons afegeix vèrtexs que no ens interessen. A continuació veiem un exemple de com es detecten els vèrtexs en un senyal usant aquest algorisme:

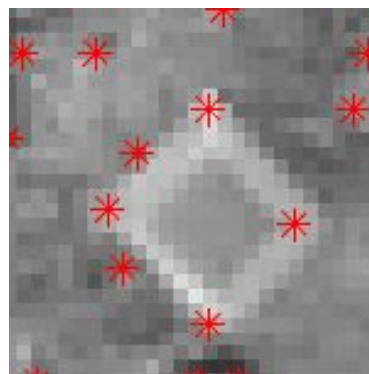


Fig 5. Detecció de vèrtex usant l'algorisme de Harris

Com podem observar, inicialment semblava una bona idea però entre el soroll del fons i la resolució de les imatges, no sembla que aquest descriptor ens pugui ajudar a l'hora de classificar.

Classificadors

Un cop hem obtingut el dataset compostat pels vectors de features de les imatges, hem utilitzat l'aplicació Classification Learner de Matlab per veure quin model s'ajusta millor a les nostres dades. Per a obtenir una puntuació i per tant poder comparar els models entre si, hem fet servir cross validation amb 5 particions.

Donat que estem fent servir els HOGs descriptors, es bona idea fixarse amb la família de les SVM i KNN ja que funcionen molt bé per aquest tipus de característiques.

Experimentació

Característiques

A l'hora d'experimentar amb les diferents característiques que hem trobat en l'apartat anterior, ens hem adonat que algunes d'elles no ajudaven a millorar la predicció del model, per tant han estat descartades del model final. A més a més, donat que la gran majoria d'imatges tenien fons i soroll, va decidir aplicar un filtre de mitjana a les senyals per a suavitzar-les una mica. Gràcies a això, el càlcul de característiques és una mica millor.

Primer de tot vam començar amb els Hogs, ja que creiem que la forma era una molt bona manera de distingir senyals de trànsit. Per aquesta raó va ser la primera característica que vam incorporar al vector de features. Vam provar el classificador només amb aquesta feature i ens vam sorprendre amb els resultats. Utilitzant una SVM amb kernel quadràtic, vam aconseguir un accuracy del 89%. Donat aquest resultat vam decidir partir d'aquesta base i intentar que els features que afegíssim al vector de característiques servissin per a millorar aquesta puntuació.

La següent característica que vam intentar afegir al vector de característiques va ser el color, ja que al costat de la forma són les característiques més importants (a priori) per a diferenciar senyals. En aquest cas, vam veure com el resultat de la classificació era lleugerament

superior, arribant al 91% . Cal destacar que de tots els possibles descriptors de colors comentats a l'apartat de colors, el que millor resultat ens ha donat ha estat el d'agafar els histogrames RGB. Tots els altres empitjoraven el resultat

Després del color vam intentar afegir la resta de característiques que creiem que eren menys importants (Ràtio i Cercles). A l'hora d'afegir-los vam veure que la puntuació seguia sent la mateixa i el programa es ralentia degut al càlcul d'aquestes característiques. Per tant vam arribar a la conclusió que realment l'única característica útil de les que havíem escollit eren els HOGs i el Color.

Segmentació

Hem intentat fer una segmentació amb la funció de matlab *activecountourn*. Aquesta funció separa el senyal del fons basant-se en un algorisme de region growing que utilitza la tècnica de contorns actius. En vista dels resultats, però, hem decidit experimentar per altres vies que no impliquin segmentació. Degut a la irregularitat de les imatges, la qualitat d'algunes així com la il·luminació o borrossitat que tenien, s'ha decidit deixar de banda la segmentació.

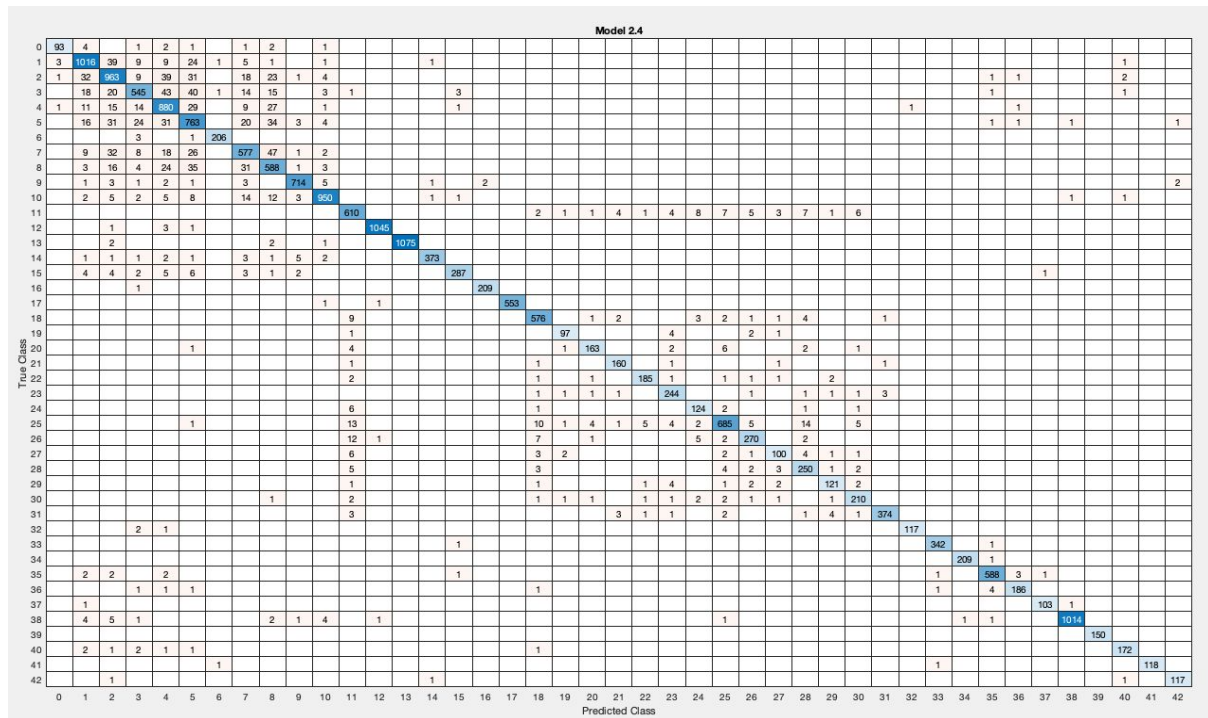
Resultats (Matriu confusió)

Gràcies a l'App Classification Learner, hem pogut provar i testejar bastants models de classificació d'una manera bastant senzilla. De tots els models ens hem quedat amb els 4 millors, que són els següents:

Model	Accuracy
Fine KNN	92,4%
Quadratic SVM	91,0%
Medium KNN	86,7
Cosine KNN	86,4

Com podem veure, el millor model que hem pogut treure és el de **Fine KNN**, amb un accuracy del **92,4%**.

La matriu de confusió per al Fine KNN és la següent:



Funcions i Llibreries usades

- **getImAreaRatio**: Retorna el descriptor de l'àrea de la senyal.
- **getImageFeatures**: Retorna el vector de característiques d'una imatge.
- **getIfCircle**: Retorna el descriptor del cercle.
- **getHogs**: Retorna el descriptor de HOGs.
- **getColorFeatures**: Retorna el descriptor de color.
- **filterImage**: Retorna una imatge filtrada amb un filtre de mitjana
- **findCircle**: Retorna 1 si troba un cercle i 0 si no en troba.
- **image2palette [1]**: Funció extreta de Matlab File Exchange. Agafa una imatge, la converteix a $L*a*b$, i aplica un kmeans per a trobar els k colors que més apareixen en la imatge. També retorna el nom del color i el percentatge que representa en la imatge.
- **hog_feature_vector**: Funció extreta de Matlab File Exchange. Donada una imatge calcula el vector de HOGs.
- **minSize**: Calcula la mida de l'imatge més petita del dataset
- **saveTrainImages**: Importa les imatges de Train en una estructura del tipus Cell.
- **predictTrafficSignal**: Demana una imatge i en retorna la predicció de la senyal que hi ha a la imatge.

References

- [1] HioBeen, H., 2018. *Image2palette: Simple K-Means Color Clustering*. [online] Mathworks.com. Available at:
<<https://www.mathworks.com/matlabcentral/fileexchange/69538-image2palette-simple-k-means-color-clustering>>
- Apunts de l'assignatura de Visió per Computador. FIB-UPC.
- Mathworks.com. 2021. *Extract Histogram Of Oriented Gradients (HOG) Features - MATLAB Extracthogfeatures*. [online] Available at:
<<https://www.mathworks.com/help/vision/ref/extracthogfeatures.html>>
- Mathworks.com. 2021. Classifier Learner App. [online] Available at:
<https://www.mathworks.com/support/search.html/videos/classify-data-using-the-classification-learner-app-106171.html?fq=asset_type_name:video%20category:stats/classification&page=1>
- Mathworks.com. 2021. *Active Countourn* - [online] Available at:
<<https://www.mathworks.com/help/images/ref/activecontour.html>>
- Mathworks.com. 2021. *Image Segmentation*. [online] Available at:
<https://www.mathworks.com/discovery/image-segmentation.html?s_tid=srchtitle>
- Mathworks.com. 2021. *Hough Transform*. [online] Available at:
<<https://www.mathworks.com/help/images/ref/hough.html>>