VISIÓ PER

COMPUTADOR

Informe de laboratori 8

Marc Cervilla Rovira

05/05/2021

**Introducció:**

En aquesta sessió estudiarem els descriptors de forma en objectes binaritzats i el classificador TreeBagger per a predir quin objecte estem observant.

En concret, utilitzarem la imatge que podem veure a continuació, on es mostren els caràcters de les matrícules espanyoles, que seran els objectes que voldrem classificar.



El que voldrem fer és analitzar quins són els millors descriptors dels caràcters per tal de, si li passem una altre imatge on apareixen els caràcters, els quals es poden mostrar lleugerament diferent de la imatge anterior, intentar adivinar de quins caràcters es tracta. Per exemple utilitzarem aquesta altre imatge:



**Implementació:**

Per analitzar i comparar els resultats, he calculat 2 valors que em serviran per veure que tan bé funciona el classificador i voldrem maximitzar-los:

SCORE em permet saber la mitjana dels scores de tots els caràcters:

SCORE = sum(max(score))/30;

Si recorrem la taula, podem calcular quants caràcters hem encertat (positius) i si els dividim entre el total podem calcular la accuracy:

count = 0;

for c = 1:30

if T.Name{c}==T.Label{c}

count=count+1;

end

end

accuracy = count/30;

Un cop tenim aquests valors, realitzarem l’analissis de diferents classificadors que analitzaran diferents propietats i, observant la accuracy i el score podrem comparar els resultats. Com més s’acostin a 1, millors resultats.

En la primera execució, veiem que el classificador del codi que ens han donat acerta bastants caràcters pero altres tenen uns scores molt baixos i les classifica malament.

Per analitzar els resultats fico les propietats que estudia el classificador seguit dels valors que hem calculat anteriorment.

'**Area','MajorAxisLength','MinorAxisLength':**

**SCORE:** 0.4613

**ACCURACY:** 0.8333

Així que analitzem el que estem fent:

Area, funciona bé en aquest exemple pero si variessim el tamany dels caràcters és obvi que no funcionaria gens bé.

Amb MajorAxisLength i MinorAxisLength passa el mateix, ja que si canviem el tamany de la lletra els resultats serien més alts si la lletra és més gran i més baixos en cas contrari.

Això si, amb les rotacions funcionen bé ja que en cas de rotar el caràcter, seguirien tenint la mateixa àrea i longituds en l'el·lipse del objecte.

El que he pensat que podriem fer per millorar els resultats és ficar la relació

r = MinorAxisLength/MajorAxisLength.

relacio = [props.MinorAxisLength]./[props.MajorAxisLength];

**‘Relacio’:**

**SCORE:** 0.5356

**ACCURACY:** 0.5667

Utilitzant aquesta relació els resultats no són gens els esperats, aconseguim millorar una mica les puntuacions dels caràcters, pero veiem que les lletres que acertem són moltes menys, algo que no ens podem permetre.

Això és degut a que només utilitzem una variable i l’algorisme del classificador necessitaria més paràmetres per tal d’estar més segur de quin objecte es tracta, ja que per exemple, si només utilitzem la relació, podria ser que dos caràcters tinguessin un valor molt semblant.

Així que volem afegir noves propietats que ens ajudin a predir quins caràcters són.

Analitzant totes les propietats que ens permet observar la funció regionprops les que més m’agraden i analitzarem els resultats són les següents:

* Circularity
* Eccentricity
* Extent
* Solidity

Totes les demés propietats depenen de la rotació i el tamany del caràcter, per tant, no els tindrem gaire en compte.

**‘Eccentricity’, ‘relacio’, ‘Circularity’:**

**SCORE:** 0.5017

**ACCURACY:** 0.8

Aquí ja hem augmentat bastant el score i acertem el 80% dels caràcters, ens estem centrant molt en la elipse que genera el caràcter (totes les propietats es centren en ella) així que haurem de afegir alguna més que es fixi en altres coses.

**‘Eccentricity’, ‘relacio’, ‘Circularity’, ’Solidity’:**

**SCORE:** 0.5210

**ACCURACY:** 0.9333

Afegint la propietat de solidity que calcula una proporció de pixels, per tant no depèn del tamany o rotació, veiem que ja obtenim resultats bastant bons.

També he fet proves amb altres propietats que no són de les anteriors, la que millors resultats m’ha donat ha sigut utilitzant també l’orientació.

Aquesta calcula l’angle entre que té l’eix principal de la elipse i he observat que podia apareixer en negatiu i en positiu, per tant he fet el valor absolut.

absol=abs([props.Orientation]);

**‘Eccentricity’, ‘relacio’, ‘Circularity’, ’Solidity’, ‘absol’:**

**SCORE:** 0.4910

**ACCURACY:** 0.9667

Aquesta execució és la que m’ha donat el millor resultat d’accuracy, però penso que és per aquest exemple en concret, ja que com que calcula l’angle que esta posicionat l’objecte, els resultats si els caràcters estiguessin molt rotats no serien bons i, per tant, no és una bona propietat a utilitzar si esperem rotacions.

**‘Eccentricity’, ‘relacio’, ‘Circularity’, ’Solidity’, ‘Extent’:**

**SCORE:** 0.5073

**ACCURACY:** 0.9333

Penso que la propietat Extent també és molt interessant, ja que mira la relació de píxels de la regió amb la del quadrat delimitador, però els resultats són molt semblants a l’execució que habiem fet sense l’extent, per tant trobo que ja costa molt millorar-los.

També he probat la 2a imatge que ens donen on ens complica bastant la predicció:



**‘Eccentricity’, ‘relacio’, ‘Circularity’, ’Solidity’, ‘Extent’:**

**SCORE:** 0.1497

**ACCURACY:** 0.1333

En aquesta 2a imatge obtenim uns resultats molt pobres, ja que les lletres varien molt més.

Treient la solidity aconseguim millorar els scores pero segueix predint fatal:

**‘Eccentricity’, ‘relacio’, ‘Circularity’, ‘Extent’:**

**SCORE:** 0.2850

**ACCURACY:** 0.2

Finalment, he provat d’utilitzar una imatge real amb una matrícula per veure com funcionava. La imatge binaritzada de la matrícula és la següent:



He hagut de canviar la següent línia de codi que trobo que hi havia un error i faltava transpossar score:

T = table(label,max(score')','VariableNames',{'Label','Score'});

**‘Eccentricity’, ‘relacio’, ‘Circularity’, ‘Extent’:**

**SCORE:** 0.2850

**ACCURACY:** 0.2857

He provat altres propietats, inclus propietats que he calculat que he pensat que podrien ser bones. Per exemple la relació entre els diàmetres mínim i màxim de Feret:

relacio2 = [props.MinFeretDiameter]./[props.MaxFeretDiameter];

També he provat diferents combinacions de les propietats anteriors però no he aconseguit millorar els resultats que he mostrat anteriorment.

**Conclusió:**

Per tant, la conclusió que trec d’aquesta sessió és que un cop enteses com funcionen totes les propietats que ens permet observar regionprops i com reaccionen a les rotacions i als canvis de mida, encara és molt complicat predir amb exactitud els caràcters que canvien molt respecte la imatge original.

Hem vist que els resultats un cop variem molt els caràcters són molt pobres i penso que és degut a que hi ha molts caràcters a predir, si féssim aquesta sessió intentant predir un conjunt d’objectes més petit obtindríem millors resultats.

Per últim, penso que també és degut a que els nombres i lletres de les matrícules, com que tenen un tamany i unes característiques molt similars és molt fàcil confondre’s al predir-ho i necessitaríem extreure característiques molt més específiques, com per exemple dividir el caràcter en regions i analitzar la forma de cadascuna.