

Problema 3: Classificació d'imatges

Arey Ferrero Ramos

12 de juny del 2022

Índex

Especificacions	3
Disseny i implementació	3
Decisions de disseny, problemàtiques i solucions proposades	3
Pseudocodi	5
Equalització adaptativa i binarització de llindar global	5
Binarització de llindar local	6
Equalització adaptativa i binarització de llindar local	7
Implementacions.....	8
Equalització adaptativa i binarització de llindar global	8
Binarització de llindar local	9
Equalització adaptativa i binarització de llindar local	10
Exemple d'evolució del processament de les imatges.....	10
Equalització adaptativa i binarització de llindar global	11
Binarització de llindar local	13
Equalització adaptativa i binarització de llindar local	15
Jocs de proves	17
Fonts documentals.....	18

Especificacions

Es disposa d'una carpeta que conté un conjunt d'imatges de cucs intestinals pressos amb un microscopi. En algunes d'aquestes imatges tots els cucs estan vius, en altres tots els cucs estan morts i en les restants hi ha cucs vius i cucs morts. Els cucs vius es caracteritzen per tenir una forma curvilínia, mentre que els cucs morts tenen una forma rectilínia. S'ha de desenvolupar un programa que classifiqui les imatges en dues categories, en funció de a quina hi pertanyen la majoria de cucs de la imatge. Les categories són: cucs vius i cucs morts. Aquest programa retornarà la precisió, és a dir, quantes imatges del total han estat classificades correctament.

Disseny i implementació

Decisions de disseny, problemàtiques i solucions proposades

Per tal d'entendre com s'ha plantejat la resolució d'aquest problema de visió per computador, primer s'ha d'entendre la naturalesa del problema. Dit d'un altra manera, s'ha d'entendre en base a quin element del problema es vol portar a terme la classificació. Per tant, tot i que pugui semblar contra intuïtiu, el plantejament del problema s'ha de fer des de les etapes finals a les etapes inicials. Dit d'un altre manera, primer s'estableix un mecanisme de classificació que doni una precisió acceptable i després es fan proves amb diferents tècniques de preprocessament per tal de millorar la precisió.

En el nostre cas, els cucs es classifiquen com a cucs vius si tenen una forma rectilínia o com a cucs morts si tenen una forma curvilínia. Per tant, la classificació es fa segons una característica concreta del cuc. No s'han d'utilitzar ni mètodes estructurals ni mètodes sintàctics perquè aquests mètodes es basen en l'ús de descriptors de regió i/o descriptors de frontera, que descriuen el contingut i la frontera dels objectes respectivament. Aquests dos elements no permeten distingir els cucs vius dels cucs morts. El mètode de reconeixement que s'ha d'utilitzar per dur a terme la classificació és un **mètode de decisió teòrica** com, per exemple, un **arbre de decisió**. Aquests mètodes porten a terme la classificació en base a una o varies característiques dels objectes.

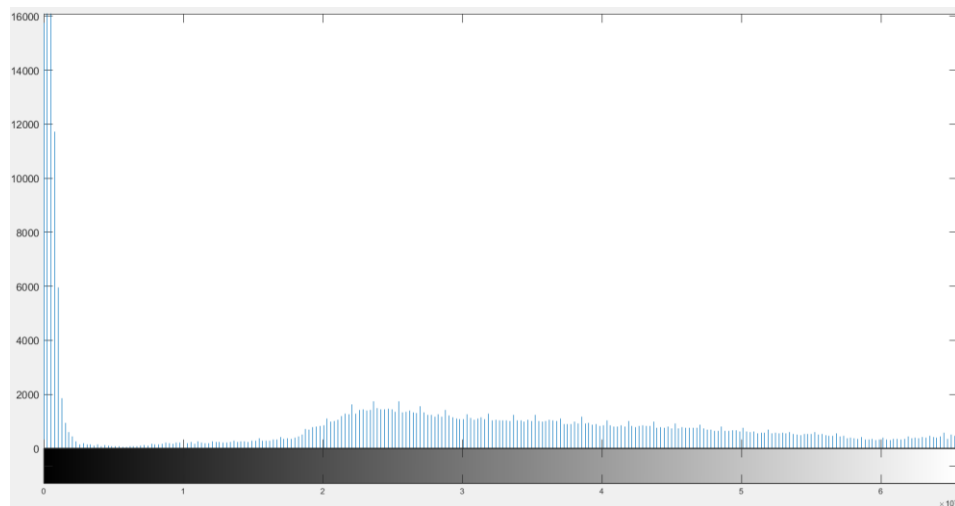
Per tal d'extreure les característiques s'utilitzaran dues funcions que estan explicades al laboratori d'identificació d'objectes (sessió 5). La primera és `bwconncomp()`, la qual rep per paràmetre una imatge i retorna una estructura de dades que conté informació sobre els objectes de la imatge. La segona funció és `regionprops()`, la qual rep per paràmetre l'estructura retornada per la funció anterior i el nom de la característica que es vol utilitzar per classificar l'objecte i retorna un vector de mida equivalent al número d'objectes, a cada posició del qual hi ha el valor numèric d'aquesta característica per cada objecte de la imatge. Amb aquesta informació es podrà realitzar la classificació.

Per tant, el següent pas ha estat seleccionar la característica més adequada. Ni el color ni l'esquelet de la regió permeten distingir els cucs vius dels cucs morts. La característica ha de ser una **característica geomètrica**. D'entre les característiques geomètriques explicades a classe de teoria sembla que hi ha més d'una opció, però la que jo considero més adequada és la **relació**

d'aspecte, que es calcula com el quocient del costat gran entre el costat petit del rectangle englobant (rectangle d'àrea mínima que conté l'objecte). S'ha de tenir en compte que la funció `regionprops()` no reconeix aquesta característica a l'ús ni tampoc el costat major i el costat menor del rectangle englobant, però sí que reconeix l'eix major i l'eix menor de l'el·lipse englobant (el·lipse d'àrea mínima que conté l'objecte), el quocient dels quals també és la relació d'aspecte.

Degut a que la característica que s'utilitzarà per a fer la classificació és una característica geomètrica i no d'altres com el color, per facilitar-ne l'extracció es pot fer una **binarització**. De fet, és molt recomanable. Analitzant la manera en que estan il·luminades les imatges, sembla que utilitzar una **binarització de llindar global** no donarà resultats suficientment bons. Per tant, caldrà millorar la qualitat de la imatge utilitzant certes tècniques de preprocessament o utilitzar una **binarització de llindar local**. La funció per fer binaritzacions, que és mencionada en diversos temes dels explicat en les sessions de laboratori, és `imbinarize()`. Si l'únic paràmetre que si li passa és la imatge, farà una binarització de llindar global. En canvi, si s'especifica el paràmetre `'adaptive'`, farà una binarització de llindar local. En aquest últim cas s'haurà d'especificar el llindar amb el paràmetre `'Sensitivity'`.

Les tècniques de processament que s'han avaluat son les explicades a classe de teoria y a laboratori de preprocessament (sessió 2): el suavitzat, la normalització i l'equalització. S'han provat tres tipus de **suavitzat**: El **suavitzat de la mitjana**, el **suavitzat de la mediana** i el **suavitzat gaussià**. Cap dels tres tipus de suavitzat ha resultat en una millora de la qualitat de la imatge. La **normalització** no cal ni tan sols provar-la, ja que si s'observa l'histograma, s'aprecia clarament que tots els nivells de color estan representats en la imatge i, per tant, no hi haurà cap canvi.



L'equalització, en canvi, sí que sembla que pugui donar bons resultats ja que la equiprobabilitat dels píxels de l'histograma és molt diferent en funció del nivell de color que representen. No obstant, si després s'aplica la binarització de llindar global tots els cucs desapareixen de la imatge. Tot i que existeix la possibilitat de que es pogués evitar aplicant una binarització de llindar local, s'ha decidit utilitzar una variant de l'equalització anomenada **equalització adaptativa**, la qual aplica l'equalització a nivell de regions de la imatge. Per a implementar aquesta tècnica s'utilitza la funció `adapthisteq()`.

Tot i que la binarització i, en els cassos en que s'ha utilitzat, l'equalització adaptativa de la imatge resulten en una imatge en la que es molt més fàcil fer el tractament dels objectes, també han generat certs inconvenients que caldria resoldre abans d'arribar al pas d'extracció de les característiques. En alguns cassos, s'han produït forats en l'interior d'alguns cucs. Per reomplir aquests cucs, cal utilitzar la tècnica de **closing**, que consisteix en una dilatació seguida d'una erosió. Aquesta tècnica s'implementa amb la funció `imdilate()` seguida de la funció `imerode()`. En altres cassos, s'haurà produït un augment del soroll o hi haurà cucs que estaran en contacte, de manera que es podrien detectar com un mateix objecte. Per reduir el soroll i per separar alguns d'aquests cucs, cal utilitzar la tècnica d'**opening**, que consisteix en una erosió seguida d'una dilatació. Aquesta tècnica s'implementa amb la funció `imerode()` seguida de la funció `imdilate()`. S'ha de destacar que les funcions de Matlab que implementen operacions morfològiques treballen sobre els píxels blanc i, per tant, la imatge binaritzada s'haurà d'invertir. Depèn del tipus de binarització utilitzada y de si s'ha utilitzat o no l'equalització adaptativa, caldrà utilitzar l'opening, el closing o les dues tècniques.

Finalment, després de les operacions morfològiques mencionades, s'ha utilitzat una funció anomenada `bwareafilt()`, que rep com a paràmetre la imatge processada i un rang, la qual permet descartar tots aquells elements l'àrea dels quals sigui inferior o superior al rang. Aquesta funció és imprescindible per tal d'acabar de filtrar el soroll que no s'ha aconseguit eliminar amb la tècnica d'opening i per treure el marc de la imatge, el qual també es tractaria com un objecte més.

Pseudocodi

S'inclou el pseudocodi de tres dels dissenys amb els que s'ha aconseguit una precisió del 100%.

Equalització adaptativa i binarització de llindar global

funció `ClassificarImatges(nom_carpeta, nom_fitxer_verificació)` retorna precisió és

```
carpeta <- llegird(nom_carpeta);
verificador <- llegirf(nom_fitxer_verificació);
cont_imatges_correctes <- 0;
cont_imatges <- 0;
per (num_imatge <- 3; num_imatge < mida(carpeta); num_imatge <-
num_imatge + 1) fer
    ima=llegir_imatge(carpeta(num_imatge));
    ima_equal <- equalització_adaptativa(ima);
    ima_bin <- binarització_global(ima_equal);
    es <- element_estructurant('diamant', 2);
    ima_closed <- dilatació(erosió(ima_bin, se), se);
    ima_opened <- erosió(dilatació(ima_closed, se), se);
```

```

cucs <- filtrar_àrea(ima_opened, [300 3000]);
característiques <- obtenir_característiques(cucs);
mesures <- obtenir_mesures(característiques, 'eix_major',
'eix_menor');
num_vius <- 0;
num_morts <- 0;
per (i <- 0; i < característiques.num_objectes; i <- i + 1) fer
    si (mesures(i).eix_major / mesures(i).eix_menor < 8)
        llavors
            num_vius <- num_vius + 1;
        sino
            num_morts <- num_morts + 1;
    fsi
fper
si (num_vius > num_morts) llavors
    classificar <- 'vius';
sino
    classificar <- 'morts';
fsi
si (classificar = verificador(i).Status) llavors
    cont_imatges_correctes <- cont_imatges_correctes + 1;
fsi
cont_imatges <- cont_imatges + 1;
fper
precisió <- (cont_imatges_correctes / cont_imatges) * 100;

```

algorisme

Binàrització de l'indar local

funció ClassificarImatges(nom_carpeta, nom_fitxer_verificació) retorna precisió és

```

carpeta <- llegir(nom_carpeta);
verificador <- llegirf(nom_fitxer_verificació);
cont_imatges_correctes <- 0;
cont_imatges <- 0;
per (num_imatge <- 3; num_imatge < mida(carpeta); num_imatge <-
num_imatge + 1) fer
    ima=llegir_imatge(carpeta(num_imatge));

```

```

ima_bin <- binarització_local(ima, 0.8);
es <- element_estructurant('diamant', 2);
ima_opened <- erosió(dilatació(ima_bin, se), se);
cucs <- filtrar_àrea(ima_opened, [200 3000]);
característiques <- obtenir_característiques(cucs);
  mesures <- obtenir_mesures(característiques, 'eix_major',
    'eix_menor');
num_vius <- 0;
num_morts <- 0;
per (i <- 0; i < característiques.num_objectes; i <- i + 1) fer
  si (mesures(i).eix_major / mesures(i).eix_menor < 8)
    llavors
      num_vius <- num_vius + 1;
    sino
      num_morts <- num_morts + 1;
  fsi
fper
si (num_vius > num_morts) llavors
  classificar <- 'vius';
sino
  classificar <- 'morts';
fsi
si (classificar = verificador(i).Status) llavors
  cont_imatges_correctes <- cont_imatges_correctes + 1;
fsi
cont_imatges <- cont_imatges + 1;
fper
precisió <- (cont_imatges_correctes / cont_imatges) * 100;
falgorisme

```

Equalització adaptativa i binarització de llindar local

funció ClassificarImatges(nom_carpeta, nom_fitxer_verificació) retorna precisió és

```

carpeta <- llegird(nom_carpeta);
verificador <- llegirf(nom_fitxer_verificació);
cont_imatges_correctes <- 0;
cont_imatges <- 0;

```

```

per (num_imatge <- 3; num_imatge < mida(carpeta); num_imatge <-
num_imatge + 1) fer
    ima=llegir_imatge(carpeta(num_imatge));
    ima_equal <- equalització_adaptativa(ima);
    ima_bin <- binarització_local(ima_equal, 1.0);
    es <- element_estructurant('diamant', 2);
    ima_opened <- erosió(dilatació(ima_bin, se), se);
    cucs <- filtrar_àrea(ima_opened, [300 3000]);
    característiques <- obtenir_característiques(cucs);
    mesures <- obtenir_mesures(característiques, 'eix_major',
'eix_menor');
    num_vius <- 0;
    num_morts <- 0;
per (i <- 0; i < característiques.num_objectes; i <- i + 1) fer
    si (mesures(i).eix_major / mesures(i).eix_menor < 8)
        llavors
            num_vius <- num_vius + 1;
        sino
            num_morts <- num_morts + 1;
    fsi
fper
si (num_vius > num_morts) llavors
    classificar <- 'vius';
sino
    classificar <- 'morts';
fsi
si (classificar = verificador(i).Status) llavors
    cont_imatges_correctes <- cont_imatges_correctes + 1;
fsi
cont_imatges <- cont_imatges + 1;
fper
precisió <- (cont_imatges_correctes / cont_imatges) * 100;
falgorisme

```

Implementacions

Equalització adaptativa i binarització de llindar global


```

function precision=ClassificarImatges(name_folder, name_check_file)
    folder=dir(name_folder);
    checker=readtable(name_check_file);
    cont_correct_images=0;
    cont_images=0;
    for num_image=3:length(folder)
        ima=imread(append(name_folder, folder(num_image).name));
        ima_equal=adapthisteq(ima);
        ima_bin=~imbinarize(ima_equal);
        se=strel('diamond', 2);
        ima_closed=imerode(imdilate(ima_bin, se), se);
        ima_opened=imdilate(imerode(ima_closed, se), se);
        worms=bwareafilt(ima_opened, [300 3000]);
        features=bwconncomp(worms);
        measures=regionprops(features, 'MajorAxisLength', 'MinorAxisLength');
        num_live=0;
        num_dead=0;
        for i=1:features.NumObjects
            if (measures(i).MajorAxisLength/measures(i).MinorAxisLength < 8)
                num_live=num_live+1;
            else
                num_dead=num_dead+1;
            end
        end
        if (num_live > num_dead)
            classify='alive';
        else
            classify='dead';
        end
        if (strcmp(classify, checker{num_image-2, "Status"}))
            cont_correct_images=cont_correct_images+1;
        end
        cont_images=cont_images+1;
    end
    precision=(cont_correct_images/cont_images)*100;
end

```

Binarització de llindar local

```

function precision=ClassificarImatges(name_folder, name_check_file)
    folder=dir(name_folder);
    checker=readtable(name_check_file);
    cont_correct_images=0;
    cont_images=0;
    for num_image=3:length(folder)
        ima=imread(append(name_folder, folder(num_image).name));
        ima_bin=~imbinarize(ima, 'adaptive', 'Sensitivity', 0.8);
        se=strel('diamond', 2);
        ima_opened=imdilate(imerode(ima_bin, se), se);
        worms=bwareafilt(ima_opened, [200 3000]);
        features=bwconncomp(worms);
        measures=regionprops(features, 'MajorAxisLength', 'MinorAxisLength');
        num_live=0;
        num_dead=0;
        for i=1:features.NumObjects
            if (measures(i).MajorAxisLength/measures(i).MinorAxisLength < 8)
                num_live=num_live+1;
            else
                num_dead=num_dead+1;
            end
        end
        if (num_live > num_dead)
            classify='alive';
        else
            classify='dead';
        end
    end
end

```

```

end
if (strcmp(classify, checker{num_image-2, "Status"}))
    cont_correct_images=cont_correct_images+1;
end
cont_images=cont_images+1;
end
precision=(cont_correct_images/cont_images)*100;
end

```

Equalització adaptativa i binarització de llindar local

```

function precision=ClassificarImatges(name_folder, name_check_file)
    folder=dir(name_folder);
    checker=readtable(name_check_file);
    cont_correct_images=0;
    cont_images=0;
    for num_image=3:length(folder)
        ima=imread(append(name_folder, folder(num_image).name));
        ima_equal=adapthisteq(ima);
        ima_bin=~imbinarize(ima_equal, 'adaptive', 'Sensitivity', 1.0);
        se=strel('diamond', 2);
        ima_opened=imdilate(imerode(ima_bin, se), se);
        worms=bwareafilt(ima_opened, [200 3000]);
        features=bwconncomp(worms);
        measures=regionprops(features, 'MajorAxisLength', 'MinorAxisLength');
        num_live=0;
        num_dead=0;
        for i=1:features.NumObjects
            if (measures(i).MajorAxisLength/measures(i).MinorAxisLength < 8)
                num_live=num_live+1;
            else
                num_dead=num_dead+1;
            end
        end
        if (num_live > num_dead)
            classify='alive';
        else
            classify='dead';
        end
        if (strcmp(classify, checker{num_image-2, "Status"}))
            cont_correct_images=cont_correct_images+1;
        end
        cont_images=cont_images+1;
    end
    precision=(cont_correct_images/cont_images)*100;
end

```

Exemple d'evolució del processament de les imatges

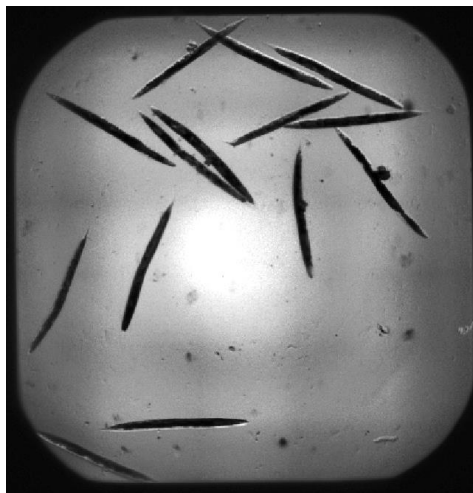
La imatge utilitzada com a model de l'evolució del processament de les imatges és la imatge 'WormA24.tif', ja que ha estat la imatge utilitzada per a fer proves. Primer de tot, es mostra la imatge original.



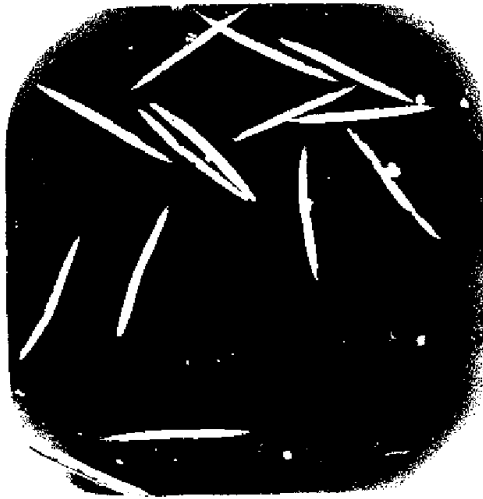
Ara es mostrarà el progrés del processament de la imatge en cada una de les tres implementacions proporcionades.

Equalització adaptativa i binarització de llindar global

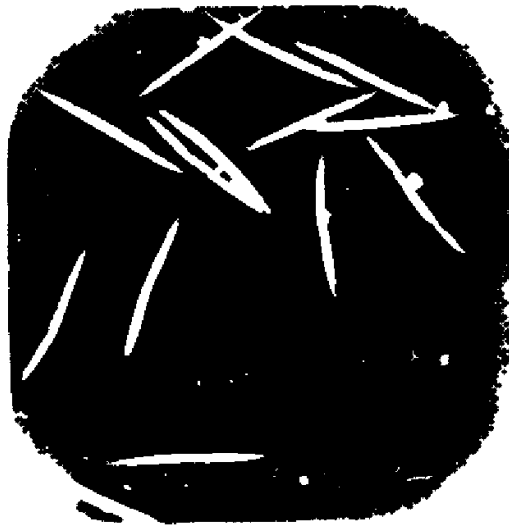
El primer pas és aplicar una equalització adaptativa sobre la imatge. S'observa una millora significativa del contrast de la imatge.



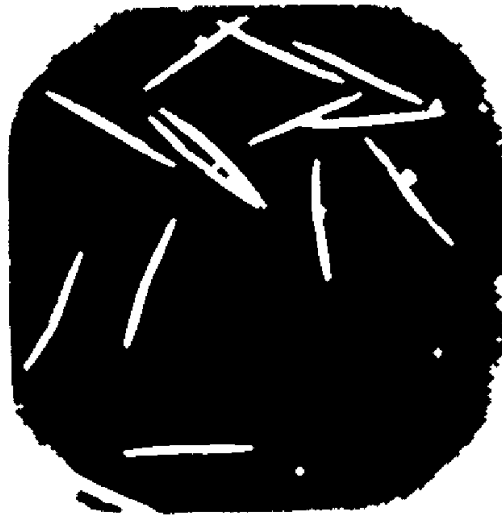
Seguidament, s'aplica una binarització de llindar global sobre la imatge anterior. Tot i la millora del contrast aconseguida anteriorment, no es suficient per a segmentar el cuc inferior esquerra (no s'aconsegueix separar del fons de la imatge).



Posteriorment, s'aplica la tècnica de closing. En aquest exemple en concret, els objectes no tenen gaires forats que es puguin omplir i, per tant, l'únic que s'aconsegueix es augmentar la dimensió d'algunes connexions existents o, fins i tot, generar-ne de noves. Per tant, en el cas d'aquesta imatge en concret, no s'hauria d'aplicar.



A continuació, s'aplica la tècnica d'opening. En aquest exemple en concret, només s'aconsegueix compensar els efectes provocats pel closing: Es trenca la connexió generada en l'etapa anterior. En aquest cas concret, aquesta tècnica hagués donat millors resultats si no s'hagués aplicat el closing anterior, ja que s'haguessin separat dos cucs que romanen junts.

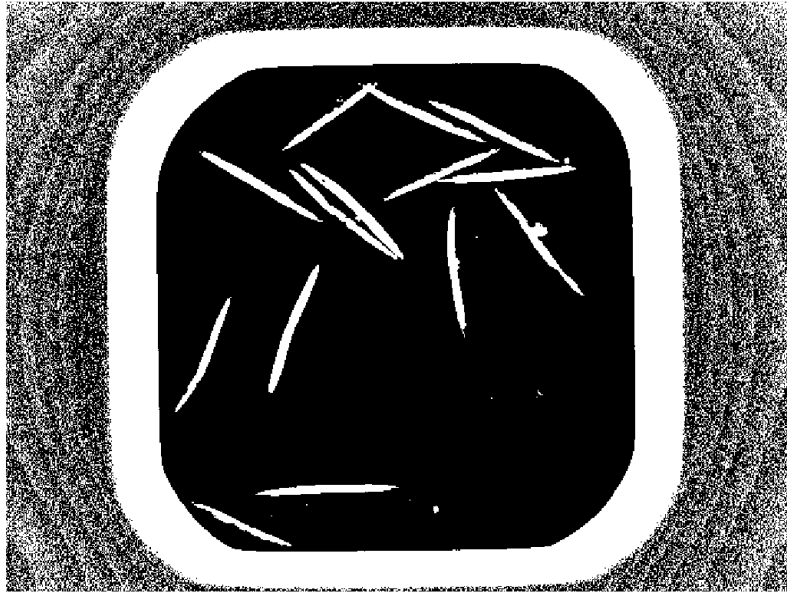


L'últim pas és l'aplicació del filtratge en funció de l'àrea. S'elimina el marc i tot el soroll restant, ja que son tractat com a objectes amb una àrea molt gran i una àrea petita, respectivament. No obstant, degut a que el cuc inferior esquerra està superposat amb el marc, també és eliminat.



Binarització de llindar local

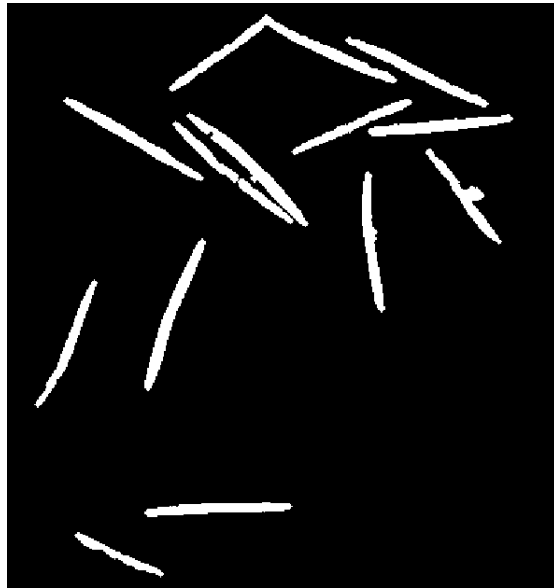
Primer de tot, s'aplica una binarització de llindar local sobre la imatge anterior amb un llindar de 0.8. A diferència d'en l'exemple anterior, el cuc inferior esquerra ha quedat perfectament separat del marc de la imatge.



A continuació, s'aplica la tècnica d'opening. Com que en aquest exemple no s'ha aplicat cap closing prèviament, s'han aconseguit trencar algunes connexions que unien alguns cucs amb els altres, de manera que es reconeixeran com a objectes independents. Per altra banda, s'han trencat dos cucs i, per tant, cada un es tractarà com a dos objectes separats quan s'haurien d'haver tractat com un de sol.

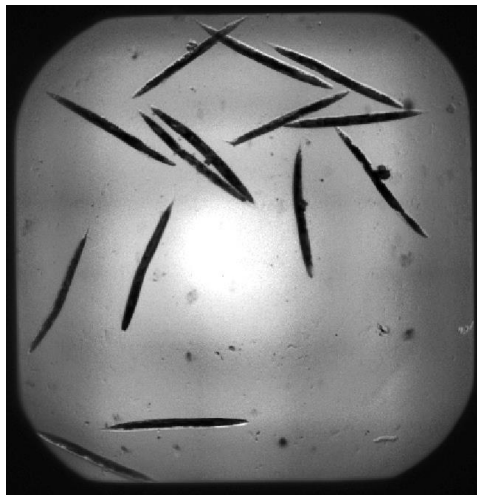


Finalment, s'aplica el filtratge en funció de l'àrea. S'eliminen els marcs i el soroll restant, ja que son tractat com a objectes amb una àrea molt gran i una àrea petita, respectivament. Com que el cuc inferior esquerra no està superposat amb el marc, no s'ha eliminat. Si que s'ha eliminat una dels dos fragments en que havia quedat dividit el cuc de més a la dreta perquè tenia una àrea més petita.

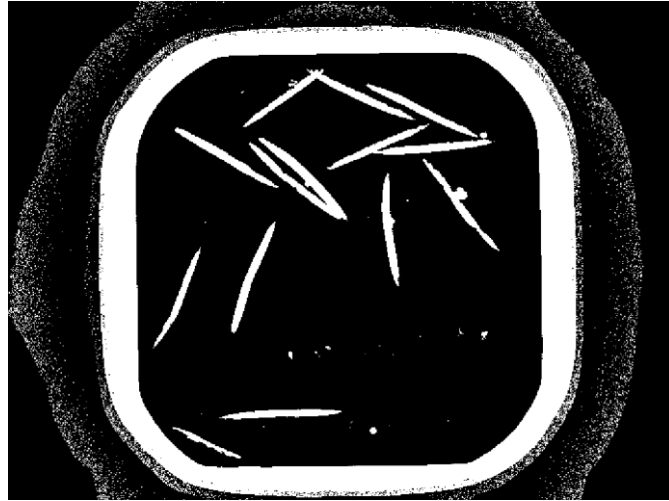


Equalització adaptativa i binarització de llindar local

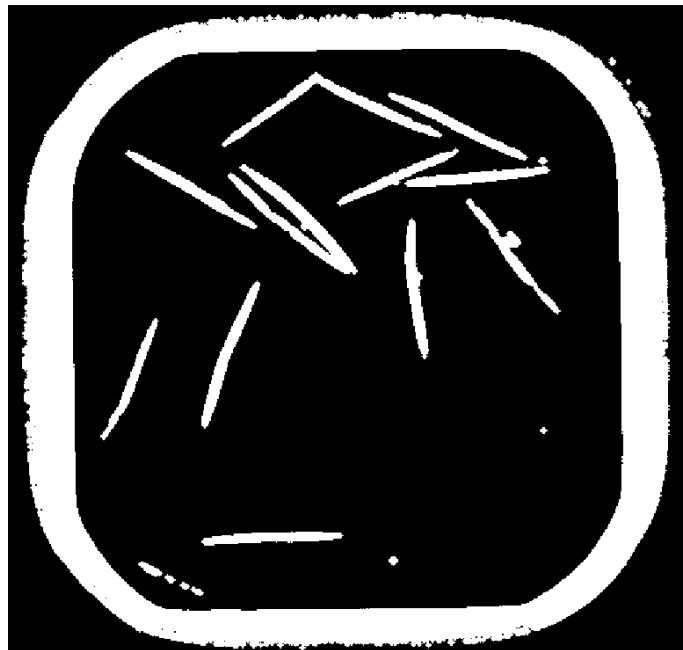
El primer pas és aplicar una equalització adaptativa sobre la imatge. S'observa la millora ja vista en el contrast de la imatge.



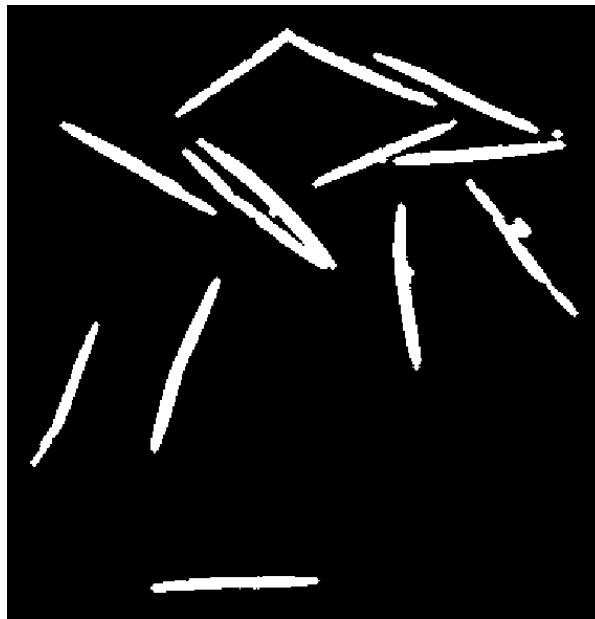
Seguidament, s'aplica una binarització de llindar local sobre la imatge anterior amb un llindar de 1.0. Igual que en el segon exemple i a diferencia del primer, el cuc inferior esquerra ha quedat perfectament separat del marc de la imatge. El principal problema que té l'aplicació de l'equalització adaptativa abans d'una binarització de llindar local, és que s'accentua el soroll present en la imatge, cosa que fa que es mantingui després d'aplicar la binarització.



A continuació, s'aplica la tècnica d'opening. Com que en aquest exemple no s'ha aplicat cap closing prèviament, s'han aconseguit trencar una connexió que unia dos dels cucs, de manera que es reconeixeran com a objectes independents. També s'ha aconseguit reduir part del soroll provocat per l'equalització adaptativa. Per altra banda, el cuc inferior esquerra que s'havia aconseguit separar del marc gràcies a l'aplicació de la binarització de llindar local apareix completament fragmentat.



Finalment, s'aplica el filtratge en funció de l'àrea. S'eliminen els marcs i el soroll restant, ja que son tractat com a objectes amb una àrea molt gran i una àrea petita, respectivament. Tot i que el cuc inferior esquerra ha quedat separat del marc, es eliminat degut a que ha estat fragmentat en trossos inferiors al valor mínim del rang. Tot i així, el resultat es millor que en la primera implementació perquè s'han trencat les connexions de dos cucs que no s'havien aconseguit separar.



Es complicat discernir quina de les tres implementacions dona un millor resultat a l'hora de classificar la imatge ja que totes tres generen problemàtiques a tenir en compte. Tot i aquestes problemàtiques, totes tres han classificat la imatge de manera correcta. Si hagués de triar, em quedaria amb la segona implementació, que sembla ser la que genera els errors menys greus. No obstant, tenint en compte que totes tres tenen una precisió del 100% en la correcta classificació del conjunt d'imatges, es podria dir que les tres son prou bones.

Jocs de proves

En el següent jocs de proves, en cada prova (entrada de la taula) es fixa un conjunt de característiques de preprocessament i processament (definides en la columna destinada a la descripció). A continuació, es varia la dimensió de l'element estructurant, l'àrea mínim de filtratge d'objectes i el valor de relació d'aspecte segons el que es fa la classificació fins arribar a la màxima precisió possible.

Prova	Descripció	Valors	Precisió màxima
1	Equalització adaptativa, binarització de llindar global, closing i opening.	Mida diamant = 2. Àrea mínima de filtratge = 300. Relació d'aspecte llindar = 8.	100%
2	Binarització de llindar local (llindar = 1.0) i closing.	Mida diamant = 2. Àrea mínima de filtratge = 300. Relació d'aspecte llindar = 8.	91'67%
3	Binarització de llindar local (llindar = 0.9) i closing.	Mida diamant = 2. Àrea mínima de filtratge = 300.	95'83%

		Relació d'aspecte llindar = 8.	
4	Binarització de llindar local (llindar = 0.8) i opening.	Mida diamant = 2. Àrea mínima de filtratge = 200. Relació d'aspecte llindar = 8.	100%
5	Binarització de llindar local (llindar = 0.7) i opening.	Mida diamant = 2. Àrea mínima de filtratge = 200. Relació d'aspecte llindar = 8'5.	100%
6	Equalització adaptativa, binarització de llindar local (llindar = 1.0) i opening.	Mida diamant = 2. Àrea mínima de filtratge = 300. Relació d'aspecte llindar = 8.	100%
7	Equalització adaptativa, binarització de llindar local (llindar = 0.9) i opening.	Mida diamant = 2. Àrea mínima de filtratge = 300. Relació d'aspecte llindar = 8.	95'83%
8	Equalització adaptativa, binarització de llindar local (llindar = 0.8) i opening.	Mida diamant = 2. Àrea mínima de filtratge = 200/300. Relació d'aspecte llindar = 8.	100%

Fonts documentals

- Documentació de les sessions de laboratori de l'assignatura.
- Eina *help*. Aquesta eina serveix per obtenir informació de les funcions de la llibreria de Matlab.
- <https://es.mathworks.com/help/images/ref/regionprops.html>