

C.1 - Projecte Final de Visió per Ordenador

Joan Bernaus Casadesús i Joan Bennàssar Martín

VO - GIA - FIB - UPC

Barcelona, a gener de 2025

Enunciat

En el camp de la biologia vegetal, un tema d'investigació recurrent consisteix a monitoritzar el creixement de diferents plantes sota diversos tractaments. Per garantir que els resultats observats són estadísticament significatius i no es deuen únicament a la inherent variabilitat biològica, s'empren múltiples rèpliques biològiques (és a dir, plantes) per a cada tractament. Tradicionalment, les tècniques de monitoratge consisteixen a extreure les plantes tractades de la terra on estan plantades i pesar-les, comparant el pes mesurat amb el d'una planta control (sense tractar). No obstant això, aquest mètode és destructiu, ja que l'extracció i el pesatge causen la mort de les plantes. Per aquesta raó, s'estan explorant estratègies basades en visió per ordinador, que permeten fer un seguiment i estimar el creixement de les plantes de manera no destructiva mitjançant l'ús d'imatges.

En aquest exercici es proporciona una imatge de referència anomenada Model.jpg, que representa una planta control. A més, a cada grup se li han assignat diverses imatges de plantes tractades sota diferents condicions. Utilitzant aquestes imatges, es demana desenvolupar un script en MATLAB que realitzi les següents tasques:

- Obtenir una estimació de l'àrea de cada planta a les imatges assignades.
- Calcular la convex hull (envolupant convexa) de cada planta i dibuixar aquesta envolupant sobre la planta a les imatges originals.
- Calcular l'àrea de l'envolupant convexa de cada planta i comparar-la amb l'àrea de la planta. El quotient entre l'àrea de l'envolupant convexa i l'àrea de la planta proporciona una estimació de la compacitat de la planta, que és diferent però relacionada amb la definició de compacitat vista a la sessió teòrica 7. Analitza la compacitat de cada planta i identifica les plantes amb major i menor compacitat dins de les imatges assignades. Reflexiona: quina informació ens proporciona la compacitat en aquest context?
- Estudiar el creixement o decreixement relatiu de les plantes assignades en comparació amb la planta control.
- Utilitzant el color predominant de la planta control com a referència, identificar les plantes el color predominant de les quals sigui significativament diferent. Això podria indicar que algunes plantes estan enveïllint prematurament (possiblement a causa del tractament) o presenten signes de toxicitat. Aquestes plantes podrien necessitar ser retirades per evitar afectar les altres.

1. Carregar les imatges

En primer lloc hem carregat les quatre imatges assignades en la pràctica des del directori específic a l'ordinador o es troben utilitzant les rutes completes generades amb la funció fullfile. Les imatges es

carreguen a través de la funció `imread`, es posen en una cel·la i s'assignen noms a cada una d'elles. Finalment, es mostra un collage de les imatges carregades utilitzant la funció `montage`.

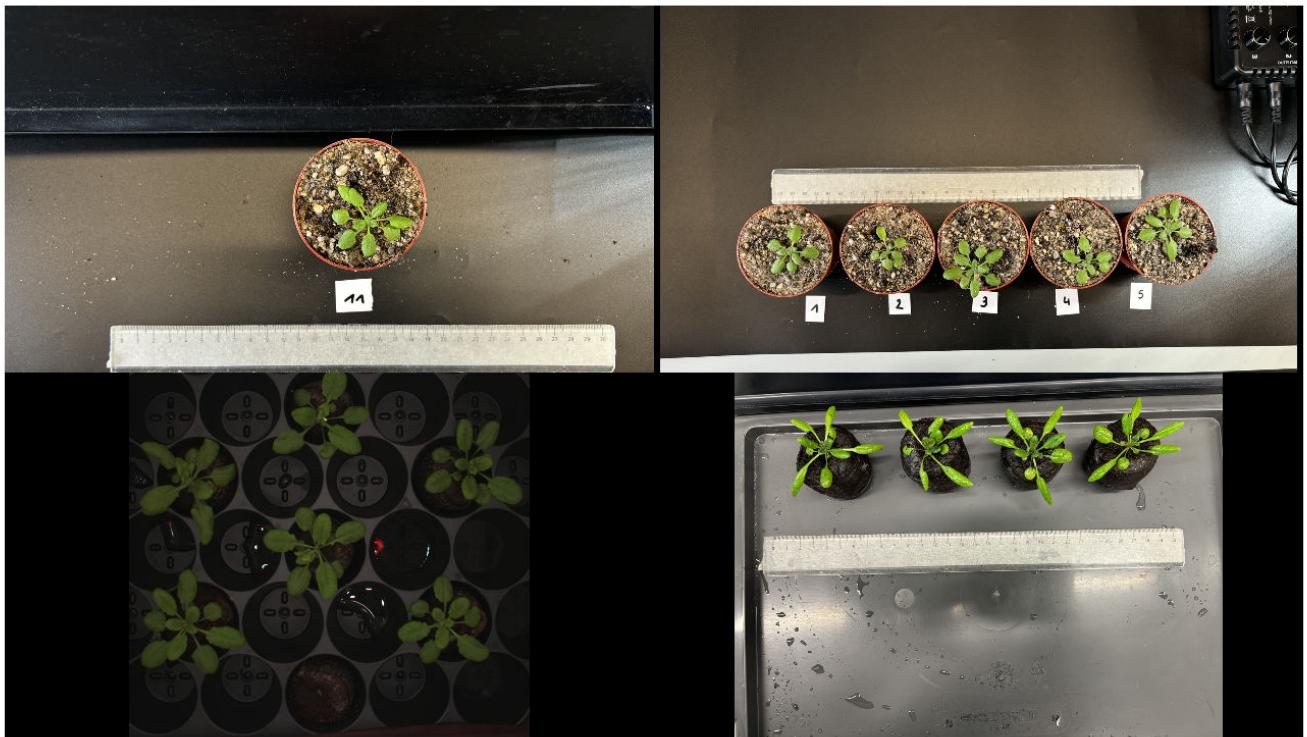
```
% Directori on es troben les imatges
dir = pwd;

% Ruta de les imatges
model_path = fullfile(dir, 'plants\Model.png');
image1_path = fullfile(dir, 'plants\22.png');
image2_path = fullfile(dir, 'plants\22.jpg');
image3_path = fullfile(dir, 'plants\22_22.png');

% Carregar les imatges
model = imread(model_path);
image1 = imread(image1_path);
image2 = imread(image2_path);
image3 = imread(image3_path);

% Llista d'imatges
imatges = {model, image1, image2, image3};
noms_imatges = {'Model', 'Imatge 1', 'Imatge 2', 'Imatge 3'};

% Mostrar les imatges en un collage
montage({model, image1, image2, image3});
```



2. Detecció i identificació de les plantes en les imatges

Una vegada hem carregat les imatges, hem de procedir amb la detecció de les plantes en les diferents imatges. Per fer-ho hem provat diferents mètodes.

2.1 Prova amb model preentrenat Mask R-CNN

En un primer moment vam creure que seria una bona idea fer ús de Mask R-CNN, que vam veure en el lab 10, per a la identificació, reconeixement i detecció de plantes en les nostres imatges gràcies a la capacitat del model per combinar detecció d'objectes i segmentació semàntica. Gràcies al codi podem veure que l'índex 59 es correspon a la classe *pottedplant* (planta en test), la qual cosa significa que el model preentrenat *resnet50-coco* ja és capaç de detectar i identificar plantes com a una de les seves classes predeterminades.

Els resultats, però, van ser decebedors com podem veure en les imatges després de fer la inferència.

```
% Cargar modelo.
maskr_cnn = maskrcnn('resnet50-coco');

% Obtener los nombres de las clases del modelo
clases_maskrcnn = maskr_cnn.ClassNames;

% Mostrar los índices y los nombres de las clases
for i = 1:length(clases_maskrcnn)
    fprintf('Índice: %d - Clase: %s\n', i, clases_maskrcnn{i});
end
```

Índice: 1 - Clase: person
Índice: 2 - Clase: bicycle
Índice: 3 - Clase: car
Índice: 4 - Clase: motorbike
Índice: 5 - Clase: aeroplane
Índice: 6 - Clase: bus
Índice: 7 - Clase: train
Índice: 8 - Clase: truck
Índice: 9 - Clase: boat
Índice: 10 - Clase: traffic light
Índice: 11 - Clase: fire hydrant
Índice: 12 - Clase: stop sign
Índice: 13 - Clase: parking meter
Índice: 14 - Clase: bench
Índice: 15 - Clase: bird
Índice: 16 - Clase: cat
Índice: 17 - Clase: dog
Índice: 18 - Clase: horse
Índice: 19 - Clase: sheep
Índice: 20 - Clase: cow
Índice: 21 - Clase: elephant
Índice: 22 - Clase: bear
Índice: 23 - Clase: zebra
Índice: 24 - Clase: giraffe
Índice: 25 - Clase: backpack
Índice: 26 - Clase: umbrella
Índice: 27 - Clase: handbag
Índice: 28 - Clase: tie
Índice: 29 - Clase: suitcase
Índice: 30 - Clase: frisbee
Índice: 31 - Clase: skis
Índice: 32 - Clase: snowboard

Índice: 33 - Clase: sports ball
Índice: 34 - Clase: kite
Índice: 35 - Clase: baseball bat
Índice: 36 - Clase: baseball glove
Índice: 37 - Clase: skateboard
Índice: 38 - Clase: surfboard
Índice: 39 - Clase: tennis racket
Índice: 40 - Clase: bottle
Índice: 41 - Clase: wine glass
Índice: 42 - Clase: cup
Índice: 43 - Clase: fork
Índice: 44 - Clase: knife
Índice: 45 - Clase: spoon
Índice: 46 - Clase: bowl
Índice: 47 - Clase: banana
Índice: 48 - Clase: apple
Índice: 49 - Clase: sandwich
Índice: 50 - Clase: orange
Índice: 51 - Clase: broccoli
Índice: 52 - Clase: carrot
Índice: 53 - Clase: hot dog
Índice: 54 - Clase: pizza
Índice: 55 - Clase: donut
Índice: 56 - Clase: cake
Índice: 57 - Clase: chair
Índice: 58 - Clase: sofa
Índice: 59 - Clase: pottedplant
Índice: 60 - Clase: bed
Índice: 61 - Clase: diningtable
Índice: 62 - Clase: toilet
Índice: 63 - Clase: tvmonitor
Índice: 64 - Clase: laptop
Índice: 65 - Clase: mouse
Índice: 66 - Clase: remote
Índice: 67 - Clase: keyboard
Índice: 68 - Clase: cell phone
Índice: 69 - Clase: microwave
Índice: 70 - Clase: oven
Índice: 71 - Clase: toaster
Índice: 72 - Clase: sink
Índice: 73 - Clase: refrigerator
Índice: 74 - Clase: book
Índice: 75 - Clase: clock
Índice: 76 - Clase: vase
Índice: 77 - Clase: scissors
Índice: 78 - Clase: teddy bear
Índice: 79 - Clase: hair drier
Índice: 80 - Clase: toothbrush

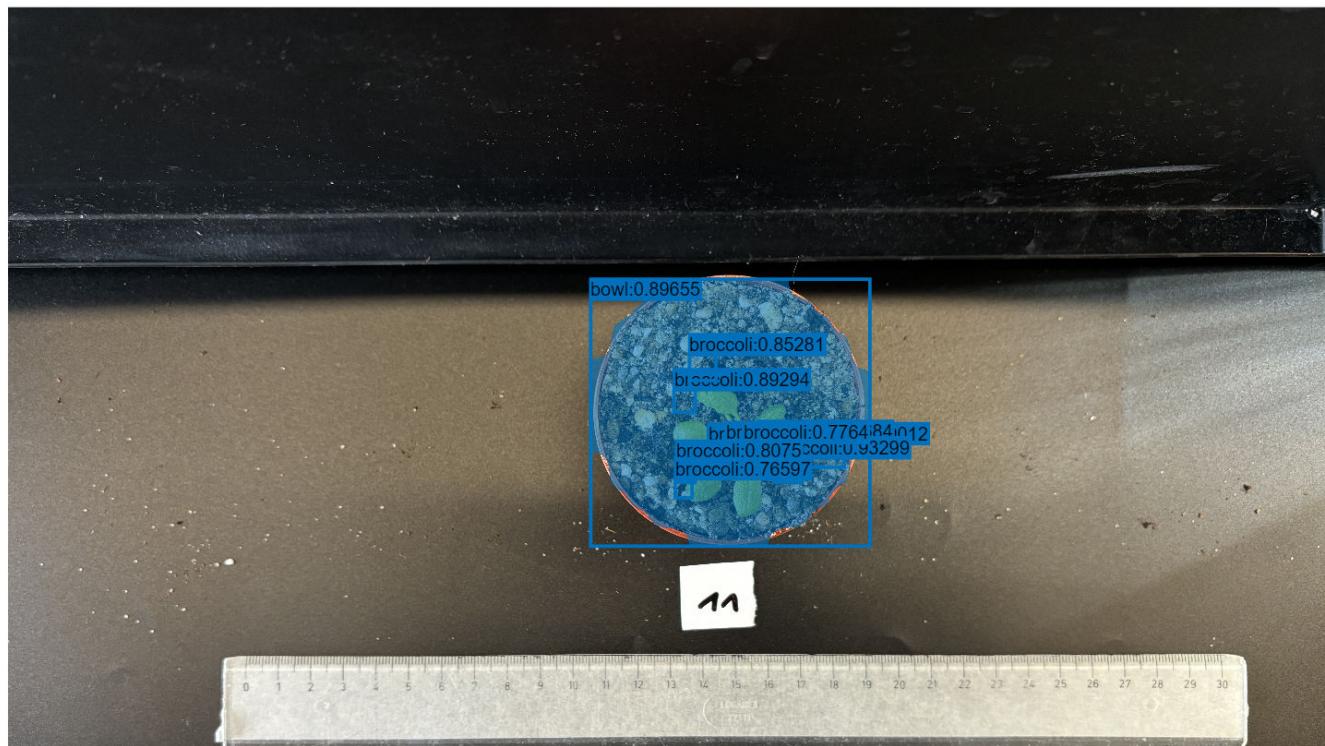
```
for i = 1:1:length(imatges)

[mascaras, etiquetas_maskrcnn, scores_maskrcnn, bb_maskrcnn] =
segmentObjects(maskr_cnn, imatges{i});

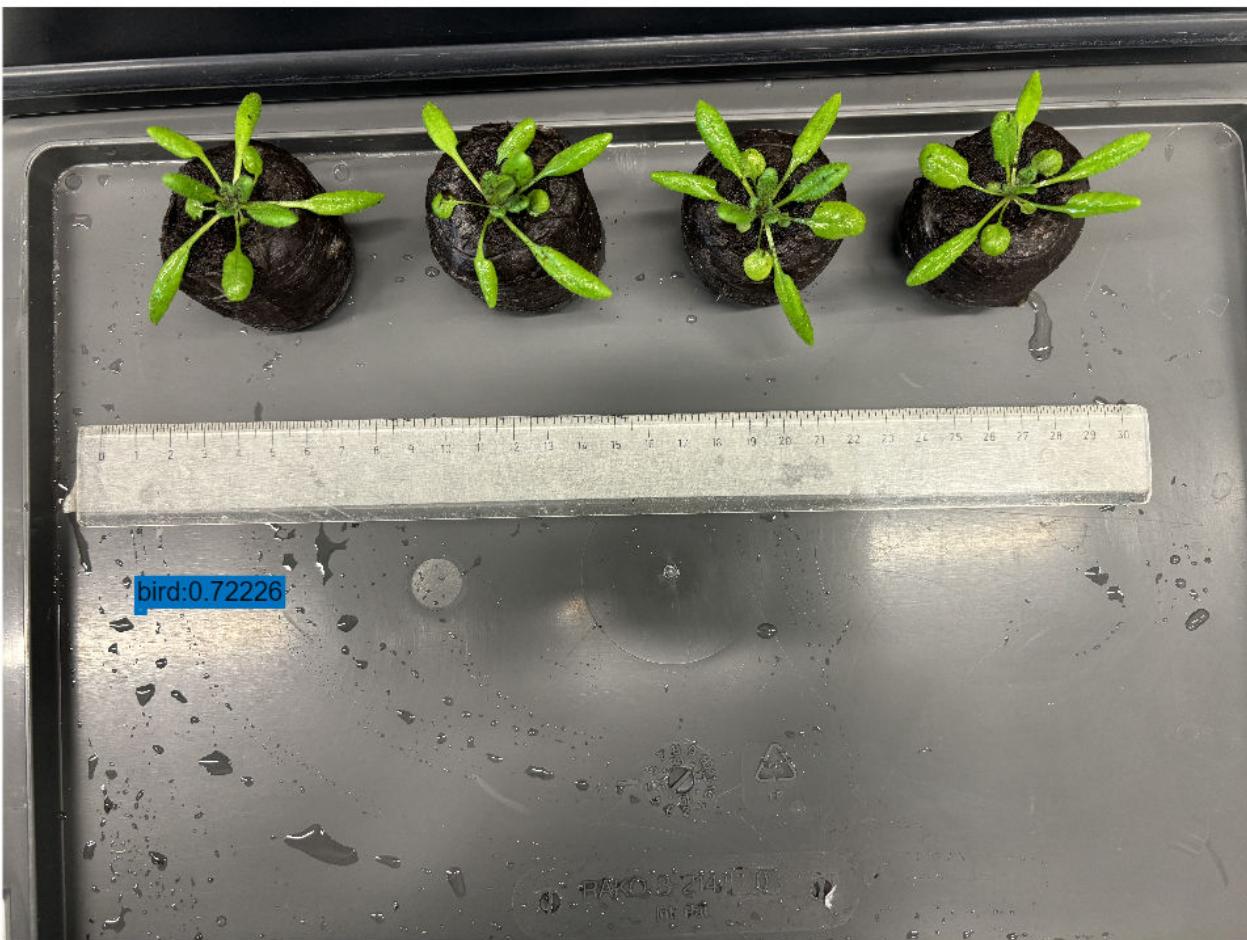
imagen_con_mascara = insertObjectMask(imatges{i},mascaras);

figure;
imshow(imagen_con_mascara);
showShape("rectangle", bb_maskrcnn,"Label",string(etiquetas_maskrcnn)
+ ':'+string(scores_maskrcnn));
```

end







2.2 Identificació amb màscara de color verd

Davant dels mals resultats presentats per l'ús del model preentrenat vam optar per una altra via. No podíem binaritzar les imatges ja què el que diferenciava les plantes en aquestes era el seu color i és així com vam optar per la següent via. Identificariem les plantes amb una màscara de color sobre les imatges originals. Primer però, ens cal determinar els umbrals de color per aplicar aquesta màscara.

2.2.1 Trobar els umbrals de color

En primer lloc, com hem dit, hem de determinar els umbrals de color. Per fer-ho usem la funció `getpts`, aquesta ens permet seleccionar píxels clicant a sobre de la imatge. Nosaltre hem clicat sobre els punts de les diferents plantes en les diferents imatges. A continuació, el codi realitzat converteix la imatge seleccionada a l'espai de color HSV i obté els valors de tonalitat (*hue*), saturació (*saturation*) i valor (*value*) dels píxels seleccionats. Els valors s'emmagatzemen en arrays globals per a ser utilitzats posteriorment.

Després de processar totes les imatges, el codi calcula els umbrals míнимs i màxims per a cada component HSV basant-se en els valors seleccionats, els de les plantes. D'aquesta manera podem coneixer ja els intervals de color que caracteritzin les plantes en les nostres imatges i procedir amb el procés d'identificació.

```
% Inicialitzar arrays per emmagatzemar els valors seleccionats
```

```

valors_hue_totals = [];
valors_saturation_totals = [];
valors_value_totals = [];

% Bucle per iterar sobre les imatges
for i = 1:length(imatges)
    % Mostrar la imatge actual
    figure;
    imshow(imatges{i});
    title(['Selecciona píxels de la ', noms_imatges{i}]);

    % Convertir la imatge actual a HSV
    imatge_hsv = rgb2hsv(imatges{i});

    % Selecció interactiva de píxels
    disp(['Selecciona píxels de la ', noms_imatges{i}, '. Pulsa Enter per
finalitzar.']);
    [x, y] = getpts; % Obtenir coordenades seleccionades

    % Convertir les coordenades a índexs enters
    x = round(x);
    y = round(y);

    % Comprovar que els índexs estan dins dels límits de la imatge
    [height, width, ~] = size(imatges{i});
    valid_points = x > 0 & x <= width & y > 0 & y <= height; % Índexs vàlids
    x = x(valid_points);
    y = y(valid_points);

    % Extreure els valors HSV dels píxels seleccionats vàlids
    for j = 1:length(x)
        valors_hue_totals = [valors_hue_totals; imatge_hsv(y(j), x(j), 1)];
        valors_saturation_totals = [valors_saturation_totals; imatge_hsv(y(j),
x(j), 2)];
        valors_value_totals = [valors_value_totals; imatge_hsv(y(j), x(j), 3)];
    end
end

```

Selecciona píxels de la Model. Pulsa Enter per finalitzar.

Selecciona píxels de la Model



Selecciona píxels de la Imatge 1. Pulsa Enter per finalitzar.

Selecciona píxels de la Imatge 1



Selecciona píxels de la Imatge 2. Pulsa Enter per finalitzar.

Selecciona píxels de la Imatge 2



Selecciona píxels de la Imatge 3. Pulsa Enter per finalitzar.

Selecciona píxels de la Imatge 3



```
% Calcular els umbrals globals (mínim i màxim de cada component HSV)
umbral_min_hue = min(valors_hue_totals);
umbral_max_hue = max(valors_hue_totals);
umbral_min_saturation = min(valors_saturation_totals);
umbral_max_saturation = max(valors_saturation_totals);
umbral_min_value = min(valors_value_totals);
umbral_max_value = max(valors_value_totals);

% Mostrar els resultats
tabla_hsv = table(valors_hue_totals, valors_saturation_totals, valors_value_totals,
...
    'VariableNames', {'Hue', 'Saturation', 'Value'});
disp(tabla_hsv);
```

Hue	Saturation	Value
0.20513	0.72589	0.77255
0.20567	0.62389	0.88627
0.19904	0.84242	0.64706
0.20865	0.65174	0.78824

0.17123	0.56154	0.5098
0.20297	0.66013	0.6
0.20357	0.69307	0.79216
0.20567	0.73822	0.74902
0.20044	0.80526	0.7451
0.20728	0.64674	0.72157
0.2058	0.71875	0.62745
0.20606	0.76923	0.56078
0.20833	0.58537	0.80392
0.20079	0.72159	0.6902
0.21404	0.45238	0.82353
0.20238	0.67879	0.64706
0.22421	0.45652	0.72157
0.21212	0.5906	0.58431
0.22	0.41899	0.70196
0.19746	0.56442	0.63922
0.20926	0.5625	0.62745
0.21099	0.47236	0.78039
0.21875	0.29358	0.8549
0.20351	0.56213	0.66275
0.20792	0.50249	0.78824
0.19599	0.59341	0.71373
0.19945	0.74847	0.63922
0.20952	0.64417	0.63922
0.21242	0.60714	0.65882
0.21099	0.65278	0.56471
0.25	0.36364	0.77647
0.21131	0.69565	0.63137
0.21333	0.54945	0.71373
0.20748	0.50256	0.76471
0.215	0.4878	0.80392
0.21515	0.57592	0.74902
0.22222	0.42056	0.83922
0.2	0.61644	0.28627
0.20667	0.625	0.31373
0.19608	0.48571	0.27451
0.2125	0.54054	0.2902
0.2093	0.59722	0.28235
0.20202	0.49254	0.26275
0.22222	0.52941	0.26667
0.21014	0.61333	0.29412
0.2027	0.55224	0.26275
0.2043	0.5082	0.23922
0.21429	0.62821	0.30588
0.21591	0.57895	0.29804
0.19919	0.56164	0.28627
0.20202	0.51562	0.25098
0.19136	0.48214	0.21961
0.2043	0.49206	0.24706
0.2043	0.53448	0.22745
0.1875	0.57143	0.27451
0.2093	0.67188	0.25098
0.21014	0.64789	0.27843
0.20098	0.5	0.26667
0.24006	0.51174	0.83529
0.19904	0.5451	1
0.23077	0.82184	0.68235
0.23288	0.82022	0.69804
0.20747	0.90995	0.82745
0.20305	0.94258	0.81961
0.18039	0.33333	1
0.21212	0.8871	0.72941
0.21642	0.91156	0.57647
0.21569	0.78462	0.76471

```

0.19883  0.22353      1
0.2114   0.9596       0.77647
0.22109   1           0.57647
0.21593   0.88827     0.70196
0.21948   0.74737     0.7451
0.22436   0.58036     0.87843
0.22574   0.90286     0.68627
0.21984   0.95882     0.66667

```

```
disp('Umbrals globals calculats:');
```

Umbrals globals calculats:

```
fprintf('Hue: Min = %.3f, Max = %.3f\n', umbral_min_hue, umbral_max_hue);
```

Hue: Min = 0.171, Max = 0.250

```
fprintf('Saturation: Min = %.3f, Max = %.3f\n', umbral_min_saturation,
umbral_max_saturation);
```

Saturation: Min = 0.224, Max = 1.000

```
fprintf('Value: Min = %.3f, Max = %.3f\n', umbral_min_value, umbral_max_value);
```

Value: Min = 0.220, Max = 1.000

2.2.2 Aplicar la màscara a les imatges originals - binarització

Així, a partir de la informació anterior respecte als llindars de les plantes definits en l'espai de color HSV (*Hue*, *Saturation*, *Value*) hem procedit a determinar els umbrals que delimiten el rang del què és planta i el què no ho és: el valor de *hue* oscil·la entre 0.17 i 0.3 (aproximadament entre 90 i 160 graus en el cercle cromàtic, associats al verd). A més, hem definit un llindar mínim de *saturation* (0.2) per excloure colors apagats i un *value* (0.15) per eliminar píxels massa foscós. Aquestes restriccions permeten aïllar els píxels que representen les plantes en les diferents imatges, excloent píxels amb colors o condicions de brillantor que podrien causar confusions.

En cada iteració sobre les imatges, el codi segueix aquests passos: primer, converteix la imatge de l'espai RGB a HSV, separant els canals (*hue*, *saturation* i *value*). A continuació, crea una màscara binària que identifica els píxels que compleixen els criteris dels umbrals definits per a la identificació de plantes. Aquesta màscara es millora amb operacions morfològiques: es fa un tancament (*closing*) per omplir petits forats dins de les regions verdes, i una obertura (*opening*) per eliminar soroll petit. Després, la màscara s'utilitza per ressaltar les àrees verdes de la imatge original, les plantes, posant a negre els píxels que no són verds. Finalment, es guarden tant la màscara millorada com la imatge amb les àrees verdes ressaltades. Al final, les màscares millorades i les imatges processades es visualitzen mitjançant un *montage*, mostrant clarament els resultats per a cada imatge del conjunt.

```
% Definir els umbrals per detectar el color verd en HSV
umbral_min_hue = 0.17; % Aproximadament 90 graus al cercle de color (verd)
umbral_max_hue = 0.3; % Aproximadament 160 graus al cercle de color (verd)
umbral_min_saturation = 0.2; % Per excloure colors apagats
umbral_min_value = 0.15; % Excloure píxels massa foscós
```

```

% Inicialitzar les llistes per emmagatzemar els resultats
mascares_millorades = cell(1, length(imatges));
imatges_verd_resaltades = cell(1, length(imatges));

% Iterar sobre les imatges
for i = 1:length(imatges)
    % Convertir la imatge actual a HSV
    imatge_actual = imatges{i};
    imatge_hsv = rgb2hsv(imatge_actual);

    % Separar els canals HSV
    hue = imatge_hsv(:, :, 1);
    saturation = imatge_hsv(:, :, 2);
    value = imatge_hsv(:, :, 3);

    % Crear una màscara binària per als píxels verds
    mascara_verde = (hue >= umbral_min_hue & hue <= umbral_max_hue) & ...
                    (saturation >= umbral_min_saturation) & ...
                    (value >= umbral_min_value);

    % Millorar la màscara amb operacions morfològiques
    mascara_mejorada = imclose(mascara_verde, strel('disk', 5)); % Tanca petits
    forats
    mascara_mejorada = imopen(mascara_mejorada, strel('disk', 3)); % Elimina soroll
    petit

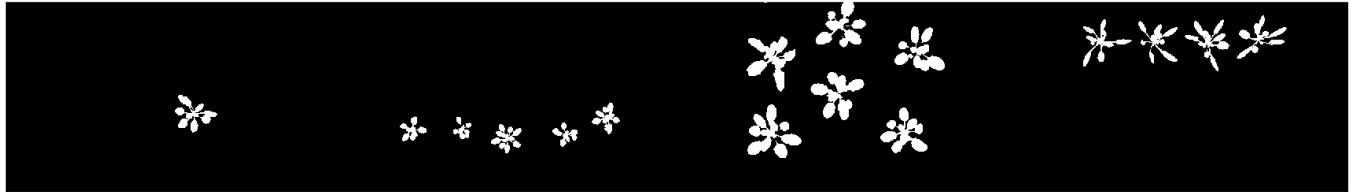
    % Guardar la màscara millorada a la llista
    mascares_millorades{i} = mascara_mejorada;

    % Aplicar la màscara a la imatge original per ressaltar les àrees verdes
    imatge_verde = imatge_actual;
    imatge_verde(repmat(~mascara_mejorada, [1, 1, 3])) = 0; % Posar a negre els
    píxels no verds

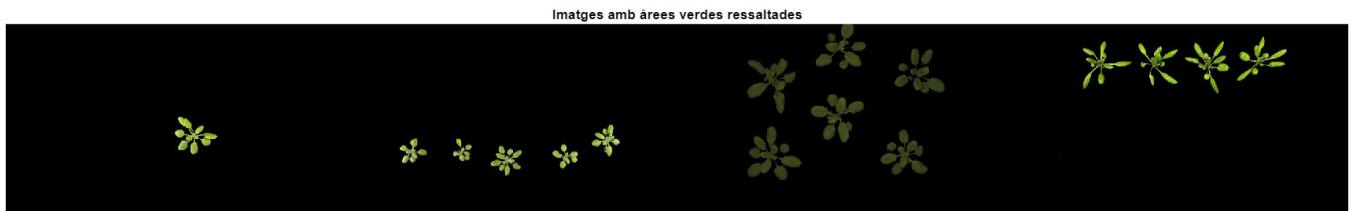
    % Guardar la imatge amb píxels verds ressaltats a la llista
    imatges_verd_resaltades{i} = imatge_verde;
end

% Visualitzar totes les màscares millorades amb montage
figure;
montage(mascares_millorades, 'Size', [1, length(imatges)]);
title('Màscares millorades per a totes les imatges');

```



```
% Visualitzar totes les imatges amb àrees verdes ressaltades amb montage
figure;
montage(imatges_verd_resaltades, 'Size', [1, length(imatges)]);
title('Imatges amb àrees verdes ressaltades');
```



2.2.3 Identificació de les plantes en cada imatge

Una vegada hem acoseguit aïllar les plantes del fons de la imatge, ara ens toca identificar les diferents plantes en cadascuna de les imatges i agrupar aquestes. El codi següent a cada iteració, converteix la imatge amb les àrees verdes destacades a una imatge binària on els píxels no negres representen les zones de plantes (verdes). Posteriorment, s'etiqueten les regions connexes utilitzant bwlabel, i es calculen les seves propietats mitjançant regionprops, incloent el seu centreoide, àrea i la llista d'índexs dels píxels que les formen.

El codi agrupa les regions detectades en funció de les distàncies entre els seus centroides, considerant que pertanyen al mateix grup si es troben a menys de 300 píxels de distància. Això ho fem ja que no totes les plantes han resultat esdevenir en regions connexes i dues fulles de la mateixa planta s'han d'identificar com a tal i no com a dues plantes diferents. Es descarten aquells grups amb una àrea inferior a 500 píxels, ja que aquestes són massa petites per ser rellevants com a plantes. Els píxels de les regions restants s'agrupen per planta dins de cada imatge. Finalment, els píxels de les plantes agrupades es guarden en una estructura de dades organitzada per imatge, permetent analitzar posteriorment les característiques de cada planta detectada en funció dels píxels que la componen.

```
% Inicialitzar les llistes per emmagatzemar els resultats
pixels_planta = cell(1, length(imatges)); % Emmagatzemem els píxels per a cada
planta per imatge

% Iterar sobre les imatges per identificar plantes i obtenir els píxels
for i = 1:length(imatges_verd_resaltades)
    % Obtenir la imatge processada amb només les àrees verdes ressaltades
    imatge_actual = imatges_verd_resaltades{i};

    % Convertir a escala de grisos i aplicar la màscara per obtenir les àrees verdes
```

```

imatge_gris = rgb2gray(imatge_actual);
imatge_binaria = imatge_gris > 0; % Considerem els píxels no negres com a verds

% Etiquetar els components connexos
[L, num] = bwlabel(imatge_binaria); % L = matriu de etiquetatge, num = nombre
de components

% Obtenir les propietats de les regions (centroides, àrees, índexs de píxels)
stats = regionprops(L, 'Centroid', 'Area', 'PixelIdxList');

% Calcular les distàncies entre els centroides i agrupar components a menys de
300 píxels
centroids = cat(1, stats.Centroid); % Obtenir els centroides de totes les
plantes
distances = pdist2(centroids, centroids); % Matriu de distàncies entre
centroides
agrupacions = (1:num)'; % Inicialment cada component és un grup separat

% Agrupar les components que estan a menys de 300 píxels de distància
for j = 1:num
    for k = j+1:num
        if distances(j, k) <= 300 % Si la distància entre els centroides és <=
300 píxels
            agrupacions(agrupacions == agrupacions(k)) = agrupacions(j); %
Agrupar les components
            end
    end
end

% Inicialitzar una cel·la per emmagatzemar els píxels de cada planta
pixels_planta_actual = cell(1, max(agrupacions)); % Crear una cel·la per cada
grup

% Agrupar les components i eliminar aquelles amb àrea menor de 500 píxels
for j = 1:num
    group_id = agrupacions(j); % Obtenir el grup de la component
    area = stats(j).Area; % Obtenir l'àrea de la component

    % Si l'àrea de la planta és >= 500 píxels, mantenir la planta
    if area >= 500
        % Afegir els píxels d'aquesta component a la llista de pixels del grup
        if isempty(pixels_planta_actual{group_id})
            pixels_planta_actual{group_id} = stats(j).PixelIdxList; % Si és el
primer, inicialitzem
        else
            % Si ja existeixen altres components associats a aquest grup,
afegim els pixels
            pixels_planta_actual{group_id} = [pixels_planta_actual{group_id};
stats(j).PixelIdxList];
        end
    end
end

```

```

    end
end

% Eliminar aquells grups que no tinguin píxels
pixels_planta_actual = pixels_planta_actual(~cellfun('isempty',
pixels_planta_actual));

% Guardar els píxels de les plantes agrupades per la imatge actual
pixels_planta{i} = pixels_planta_actual;
end

```

A continuació podem observar les plantes detectades a cada imatge ressaltant-les amb colors diferents. Primer, per a cada imatge, es crea una matriu de zeros (labels) per assignar una etiqueta única a cada planta. Les etiquetes es basen en els píxels agrupats corresponents a cada planta (emmagatzemats a planta_pixels). Després, la funció label2rgb s'utilitza per convertir les etiquetes en una imatge RGB, on cada planta apareix pintada amb un color diferent, seleccionat del colormap 'jet', amb un fons negre ('k').

```

% Inicialitzar les llistes per emmagatzemar les imatges identificades
imatges_identificades = cell(1, length(imatges));

% Iterar sobre cada imatge
for i = 1:length(imatges)
    % Obtenir la imatge processada amb només les àrees verdes ressaltades
    imatge_actual = imatges_verd_resaltades{i};

    % Crear una matriu de zeros per etiquetar les plantes
    labels = zeros(size(imatge_actual, 1), size(imatge_actual, 2));

    % Obtenir els píxels de les plantes per la imatge actual
    planta_pixels = pixels_planta{i};

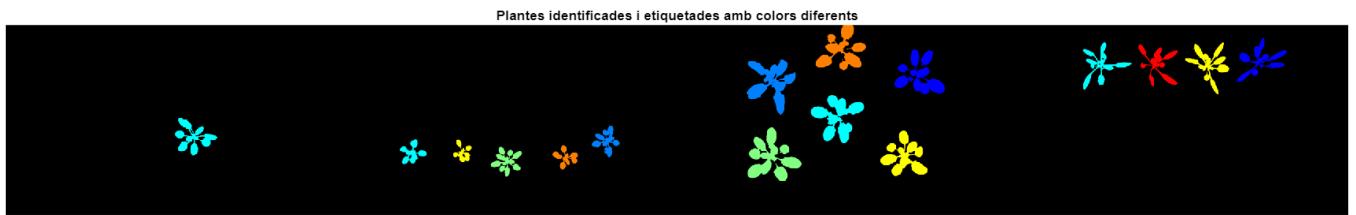
    % Assignar una etiqueta única per cada planta
    for j = 1:length(planta_pixels)
        labels(planta_pixels{j}) = j; % Assignem l'etiqueta j a la planta j
    end

    % Utilitzar label2rgb per pintar cada etiqueta amb un color diferent
    imatge_identificada = label2rgb(labels, 'jet', 'k', 'shuffle');

    % Guardar la imatge identificada amb colors a la llista
    imatges_identificades{i} = imatge_identificada;
end

% Visualitzar les imatges amb les plantes identificades
figure;
montage(imatges_identificades, 'Size', [1, length(imatges)]);
title('Plantes identificades i etiquetades amb colors diferents');

```



3. Estimació de l'àrea de cada planta

Una vegada ja tenim totes les plantes identificades en cadascuna de les imatges procedim a la primera tasca de l'enunciat, que no era altra que calcular l'àrea de les plantes. Com que no disposem dels mecanismes per escalar les mesures, assumim que totes les imatges s'han pres des-de la mateixa distància i amb la mateixa càmera, resultant així una bona mesura per calcular l'àrea els píxels que ocupa en les diferents imatges.

Assumint això, el codi següent calcula les àrees de les plantes detectades en les diferents imatges i també calcula l'àrea mitjana de les plantes per cada imatge. Per a cada imatge, es recullen els píxels de cada planta emmagatzemats a `planta_pixels`, i per cada planta, es calcula la seva àrea com el nombre de píxels que la componen. Aquestes àrees es guarden en un vector `areas_actualls` per a cada imatge. Un cop calculades les àrees de totes les plantes d'una imatge, es guarda el resultat en una llista anomenada `areas_plantes`. A més, es calcula l'àrea mitjana de les plantes de cada imatge, utilitzant la funció `mean`. Finalment, es mostren per pantalla les àrees de cada planta per cada imatge, així com l'àrea mitjana per cada una, amb dos decimals de precisió.

```
% Inicialitzar les llistes per emmagatzemar les àrees
areas_plantes = cell(1, length(imatges)); % Àrees per a cada planta per imatge
area_mitjana_plantes = zeros(1, length(imatges)); % Àrea mitjana per cada imatge

% Iterar sobre les imatges per calcular les àrees de les plantes
for i = 1:length(imatges_verd_resaltades)
    % Obtenir els píxels de les plantes per a la imatge actual
    planta_pixels = pixels_planta{i}; % Píxels de cada planta per aquesta imatge

    % Inicialitzar un vector per emmagatzemar les àrees de les plantes
    areas_actualls = [];

    % Iterar sobre cada planta
    for j = 1:length(planta_pixels)
        % Obtenir els píxels de la planta j
        pixels_planta_j = planta_pixels{j};

        % Calcular l'àrea com el nombre de píxels
        area_planta_j = length(pixels_planta_j); % Nombre de píxels
        areas_actualls = [areas_actualls, area_planta_j]; % Afegir l'àrea al vector
    end

    % Guardar les àrees de les plantes per a la imatge actual
    areas_plantes{i} = areas_actualls;
```

```

% Calcular l'àrea mitjana per a la imatge actual (si hi ha plantes)
if ~isempty(areas_actualls)
    area_mitjana_plantes(i) = mean(areas_actualls); % Àrea mitjana
else
    area_mitjana_plantes(i) = 0; % Si no hi ha plantes, l'àrea mitjana és 0
end
end

% Mostrar les àrees de les plantes i l'àrea mitjana per cada imatge
for i = 1:length(imatges)
    disp(['Àrees de les plantes per a la imatge ', noms_imatges{i}, ':']);
    disp(areas_plantes{i}); % Mostrar les àrees de les plantes
    disp(['Àrea mitjana de les plantes: ', num2str(area_mitjana_plantes(i),
'%.2f'), ' pixels']); % Mostrar l'àrea mitjana amb 2 decimals
end

```

Àrees de les plantes per a la imatge Model:
162990
Àrea mitjana de les plantes: 162990.00 pixels
Àrees de les plantes per a la imatge Imatge 1:
79961 53101 115217 69440 98801
Àrea mitjana de les plantes: 83304.00 pixels
Àrees de les plantes per a la imatge Imatge 2:
86979 98755 86836 75111 73968 76204
Àrea mitjana de les plantes: 82975.50 pixels
Àrees de les plantes per a la imatge Imatge 3:
256147 234900 300056 280865
Àrea mitjana de les plantes: 267992.00 pixels

El codi següent itera sobre el conjunt d'imatges amb àrees verdes ressaltades, es a dir amb només les plantes, i per cada imatge, identifica les plantes a través dels píxels associats a cada planta emmagatzemats a pixels_planta. A partir d'aquests píxels, el codi utilitzà la funció ind2sub per convertir els índexs de píxels en coordenades espacials (x, y). Un cop obtingudes aquestes coordenades, es calcula el **centroide** de cada planta com la mitjana de les coordenades x i y dels seus píxels. Finalment, sobre cada imatge, es mostra l'àrea de la planta al seu **centroide** utilitzant la funció text per dibuixar el text amb el nombre de píxels, i tot això es visualitza sobre la imatge original amb el centroide de cada planta com a referència.

```

% Iterar sobre les imatges per mostrar l'àrea de cada planta al seu centreide
for i = 1:length(imatges_verd_resaltades)
    % Obtenir la imatge actual
    imatge_actual = imatges_verd_resaltades{i};

    % Obtenir els píxels de les plantes per la imatge actual
    planta_pixels = pixels_planta{i}; % Píxels de cada planta per aquesta imatge

    % Crear una figura per mostrar la imatge
    figure;
    imshow(imatge_actual);
    hold on;

    % Iterar sobre cada planta per mostrar el seu àrea al centreide

```

```

for j = 1:length(planta_pixels)
    % Obtenir els píxels de la planta j
    pixels_planta_j = planta_pixels{j};

    % Convertir els índexs de píxels a coordenades (x, y)
    [y_coords, x_coords] = ind2sub(size(imatge_actual), pixels_planta_j);  %
    Convertir índexs a coordenades (x, y)

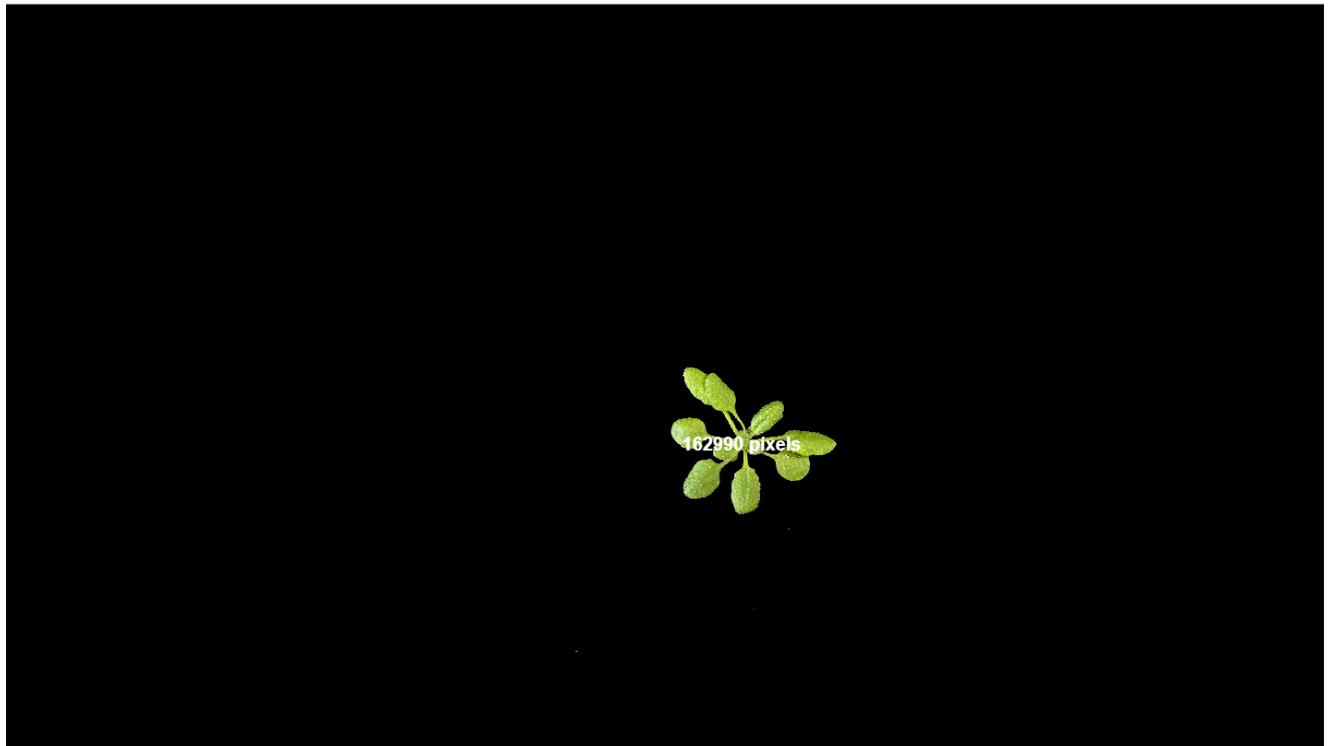
    % Calcular el centroid com la mitjana de les coordenades x i y
    centroid_x = mean(x_coords);  % Mitjana de les coordenades x
    centroid_y = mean(y_coords);  % Mitjana de les coordenades y

    % Calcular l'àrea de la planta com el nombre de píxels
    area_planta_j = length(pixels_planta_j);  % Nombre de píxels

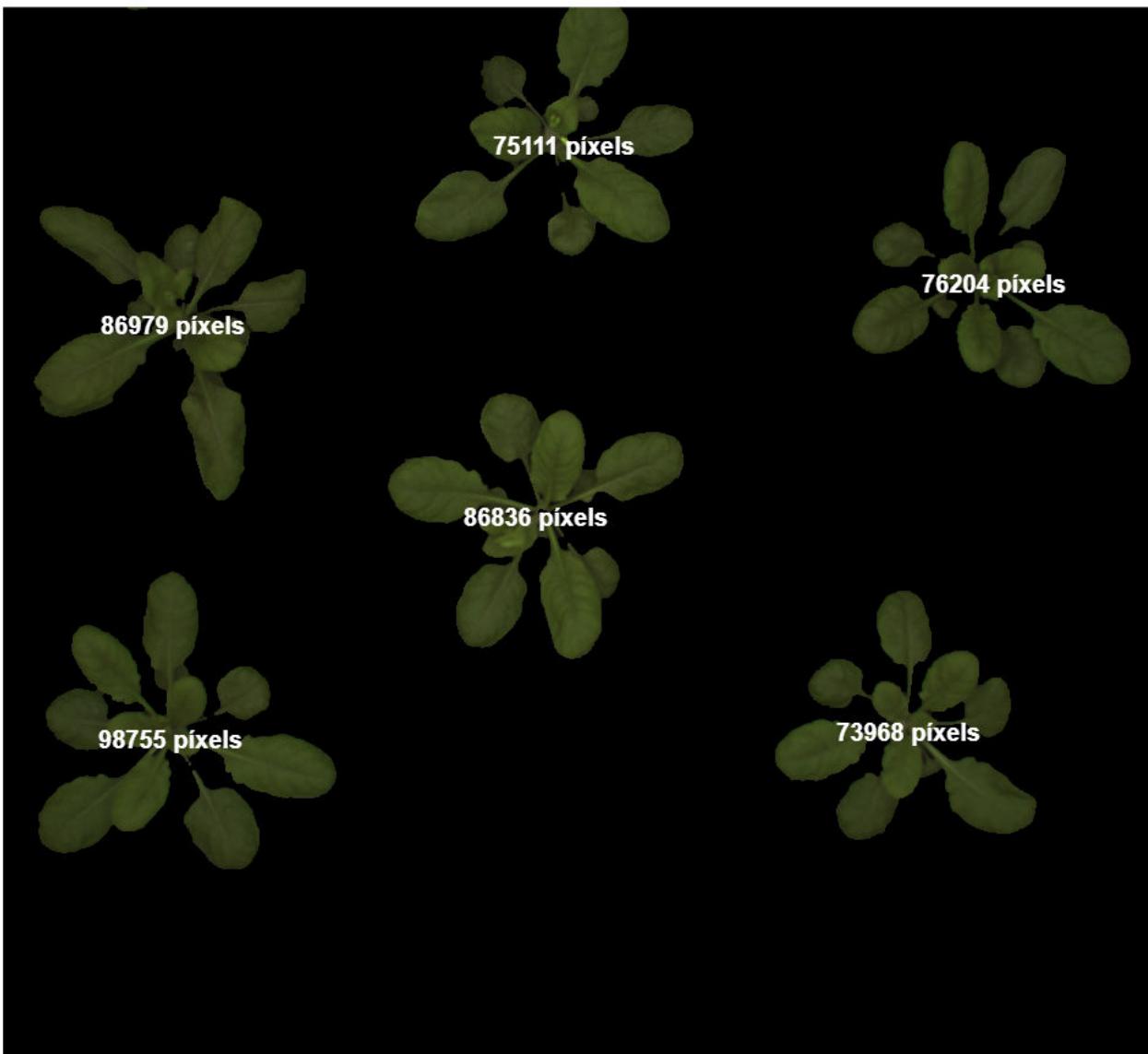
    % Mostrar l'àrea al centroid
    text(centroid_x, centroid_y, sprintf('%d pixels', area_planta_j), ...
        'Color', 'white', 'FontSize', 12, 'FontWeight', 'bold',
        'HorizontalAlignment', 'center');
end

hold off;

```









4. Càcul envoltura convexa

El següent punt de l'enunciat consisteix en el càlcul de la envoltura convexa de cada planta i dibuixar aquesta en les imatges originals. Per fer-ho, iterem sobre el conjunt d'imatges i, per a cada imatge, calculem i dibuixem l'envolupant convexa de cada planta identificada en l'imatge. Primer, el codi obté la imatge original i els píxels associats a les plantes a través de la variable `pixels_planta`. Per a cada planta, es converteixen els índexs de píxels a coordenades (x, y) mitjançant la funció `ind2sub`, com abans. Després, es calcula la **convex hull**, o envolupant convexa, dels píxels de la planta utilitzant la funció `convhull`, la qual retorna els índexs dels punts que formen aquesta envolupant. A continuació, es dibuixa la línia de l'envolupant convexa sobre la imatge original amb un color vermell, i es pinta l'interior d'aquesta envolupant amb un color vermell semitransparent mitjançant la funció `fill`. Finalment, es mostra la imatge amb les envoltures convexes dibuixades per a cada planta.

```
% Iterar sobre les imatges per calcular la envoltura convexa i dibuixar-la
for i = 1:length(imatges_verd_resaltades)
    % Obtenir la imatge original
    imatge_original = imatges{i};
```

```

% Obtenir els píxels de les plantes per a la imatge actual
planta_pixels = pixels_planta{i}; % Píxels de cada planta per aquesta imatge

% Crear una nova figura per cada imatge per dibuixar les envoltures convexes
figure;
imshow(imatge_original); % Mostrar la imatge original
hold on;

% Iterar sobre cada planta per calcular la seva envoltura convexa i dibuixar-la
for j = 1:length(planta_pixels)
    % Obtenir els píxels de la planta j
    pixels_planta_j = planta_pixels{j};

    % Obtenir les coordenades (x, y) dels píxels
    [y_coords, x_coords] = ind2sub(size(imatge_original), pixels_planta_j); %

    % Convertir índexs de píxels a coordenades (x, y)

    % Calcular la envoltura convexa (convex hull) per a aquests píxels
    k = convhull(x_coords, y_coords); % Obtenir els índexs del convex hull

    % Dibuixar la línia de la envoltura convexa sobre la imatge
    plot(x_coords(k), y_coords(k), 'r-', 'LineWidth', 2); % Dibuixar la línia
de l'envoltura en vermell

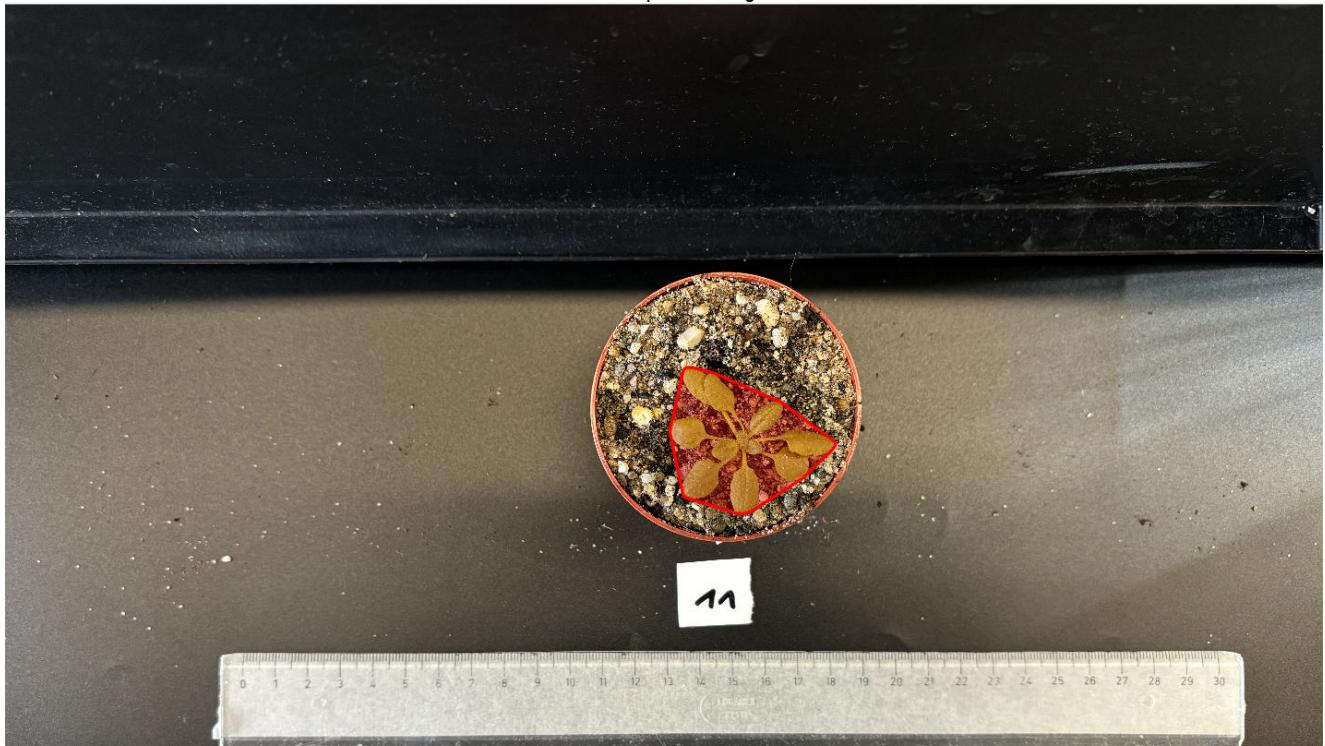
    % Pintar l'interior de l'envoltura convexa amb transparència
    fill(x_coords(k), y_coords(k), 'r', 'FaceAlpha', 0.3, 'EdgeColor',
'none'); % Pintar l'interior amb vermell semitransparent
end

% Afegir títol a la figura
title(['Envoltor convex per a la imatge ', nom_imatges{i}]);

% Mostrar la figura amb les envoltures convexes
hold off;
end

```

Envoltor convex per a la imatge Model



Envoltor convex per a la imatge Imatge 1



Envoltor convex per a la imatge Imatge 2





5. Compacitat

El següent punt en l'enunciat es correspon a calcular la compacitat. La *compacitat* d'un objecte, en termes geomètrics, és una mesura de la seva "eficiència" o "optimització" en l'ús de l'espai. Es defineix com la relació entre l'àrea de l'objecte (en aquest cas, l'àrea de la planta) i l'àrea de la seva envolupant convexa, és a dir, l'àrea mínima que el pot contenir. En el cas de les plantes, una amb poques fulles, o amb aquestes primes i allagades, presentarà una envolupant convexa més gran en comparació amb la seva pròpia àrea, la qual cosa implicarà una baixa compacitat. Per altra banda, si la planta és més compacta, amb més fulles i aquestes més grans i amples, la diferència entre l'àrea de la planta i l'àrea de la seva envolupant convexa serà petita, indicant una alta compacitat.

El següent codi calcula la compacitat de les plantes en les diverses imatges utilitzant l'àrea de la planta i l'àrea de la seva envolupant convexa. En primer lloc, s'obté l'àrea de l'envoltura convexa per cada planta, que es calcula mitjançant l'algoritme de la "convex hull", i es guarda en una llista per a cada imatge. A continuació, es compara l'àrea de la planta amb l'àrea de la seva envolupant convexa per calcular la compacitat, que es defineix com el cocient entre l'àrea de la planta i l'àrea de la seva envolupant convexa. Aquest càlcul es fa per cada planta en cada imatge.

El codi després guarda les compacitats calculades per cada planta en una llista, així com les àrees de les plantes i les envoltures convexes. Finalment, per a cada imatge, es mostren les àrees de les plantes, les envoltures convexes, la compacitat de cada planta, i les mitjanes d'aquests valors. A més, es reporten les plantes amb la major i la menor compacitat, proporcionant una visió de quines plantes tenen una forma més compacta o més irregular.

```
% Inicialitzar les llistes per emmagatzemar les àrees de les envoltures convexes
areas_envoltura = cell(1, length(imatges)); % Àrees de les envoltures convexes
area_envoltura_mitjana = zeros(1, length(imatges)); % Àrea mitjana de les
envoltures convexes

% Iterar sobre les imatges per calcular la envoltura convexa i dibuixar-la
for i = 1:length(imatges_verd_resaltades)
    % Obtenir la imatge original
    imatge_original = imatges{i};

    % Obtenir els píxels de les plantes per a la imatge actual
    planta_pixels = pixels_planta{i}; % Píxels de cada planta per aquesta imatge

    % Inicialitzar una llista per emmagatzemar les àrees de les envoltures convexes
    % per a aquesta imatge
    areas_actuals = [];

    % Iterar sobre cada planta per calcular la seva envoltura convexa i dibuixar-la
    for j = 1:length(planta_pixels)
        % Obtenir els píxels de la planta j
        pixels_planta_j = planta_pixels{j};

        % Obtenir les coordenades (x, y) dels píxels
        [y_coords, x_coords] = ind2sub(size(imatge_original), pixels_planta_j); % Convertir índexs de píxels a coordenades (x, y)

        % Calcular la envoltura convexa (convex hull) per a aquests píxels
        k = convhull(x_coords, y_coords); % Obtenir els índexs del convex hull

        % Crear una màscara binària per calcular l'àrea
        mask = poly2mask(x_coords(k), y_coords(k), size(imatge_original, 1),
        size(imatge_original, 2));
        area_envoltura = sum(mask(:)); % Comptar el nombre de píxels dins de la
        màscara

        % Afegir l'àrea calculada a la llista d'àrees per aquesta imatge
        areas_actuals = [areas_actuals, area_envoltura];
    end

    % Afegir les àrees de les envoltures convexes de la imatge actual a la llista
    % global
    areas_envoltura{i} = areas_actuals;
```

```

% Calcular l'àrea mitjana de les envoltures convexes per a aquesta imatge
area_envoltura_mitjana(i) = mean(areas_actuals);
end

% Inicialitzar les llistes per emmagatzemar les compacitats
compacitats_plantes = cell(1, length(imatges)); % Compacitat de cada planta per a
cada imatge
compacitat_mitjana = zeros(1, length(imatges)); % Compacitat mitjana per a cada
imatge

% Iterar sobre les imatges per calcular la compacitat de cada planta
for i = 1:length(imatges_verd_resaltades)
    % Obtenir els píxels de les plantes per a la imatge actual
    planta_pixels = pixels_planta{i}; % Píxels de cada planta per aquesta imatge

    % Inicialitzar un vector per emmagatzemar les compacitats de les plantes
    compacitats_actuals = [];
    areas_envoltura_actuals = areas_envoltura{i}; % Per emmagatzemar les àrees de
les envoltures convexes
    areas_plantes_actuals = areas_plantes{i}; % Àrees de les plantes per a aquesta
imatge

    % Iterar sobre cada planta per calcular la seva compacitat
    for j = 1:length(planta_pixels)

        % Area de l'envoltura convexa
        area_envoltura = areas_envoltura_actuals(j);

        % Calcular l'àrea de la planta
        area_planta = areas_plantes_actuals(j); % Obtenir l'àrea de la planta (ja
calculada)

        % Calcular la compacitat
        compacitat = area_planta/area_envoltura;

        % Afegir les àrees de l'envoltura convexa i la compacitat
        areas_envoltura_actuals = [areas_envoltura_actuals, area_envoltura];
        compacitats_actuals = [compacitats_actuals, compacitat];
    end
    % Guardar les compacitats de les plantes per a la imatge actual
    compacitats_plantes{i} = compacitats_actuals;

    % Calcular la compacitat mitjana per a la imatge actual
    compacitat_mitjana(i) = mean(compacitats_actuals); % Compacitat mitjana
end

% Mostrar les compacitats de les plantes, l'àrea de la planta, l'àrea de la
envoltura convexa,
% i les àrees mitjanes per a cada imatge
for i = 1:length(imatges)

```

```

    disp(['Compacitats de les plantes per a la imatge ', nomes_imatges{i}, ':']);

    % Mostrar les àrees de la planta, l'àrea de la envoltura convexa i la
    compacitat per planta
    for j = 1:length(compacitats_plantes{i})
        disp(['Planta ', num2str(j), ' - Àrea de la planta: ',
        num2str(areas_plantes{i}(j)), ' pixels']);
        disp(['Planta ', num2str(j), ' - Àrea de l''envoltura convexa: ',
        num2str(areas_envoltura{i}(j)), ' pixels']);
        disp(['Planta ', num2str(j), ' - Compacitat: ',
        num2str(compacitats_plantes{i}(j) * 100, '%.2f'), '%']); % Compacitat en
        percentatge
        disp('-----');
    end

    % Mostrar les mitjanes de les àrees i la compacitat
    disp(['Àrea mitjana de les plantes: ', num2str(area_mitjana_plantes(i),
    '%.2f'), ' pixels']);
    disp(['Àrea mitjana de les envoltures convexes: ',
    num2str(area_envoltura_mitjana(i), '%.2f'), ' pixels']);
    disp(['Compacitat mitjana de les plantes: ', num2str(compacitat_mitjana(i) *
    100, '%.2f'), '%']); % Compacitat mitjana en percentatge

    % Identificar la planta amb la major i menor compacitat
    [max_compacitat, idx_max] = max(compacitats_plantes{i});
    [min_compacitat, idx_min] = min(compacitats_plantes{i});

    disp(['Planta amb la major compacitat: ', num2str(max_compacitat * 100,
    '%.2f'), '% (planta ', num2str(idx_max), ')']);
    disp(['Planta amb la menor compacitat: ', num2str(min_compacitat * 100,
    '%.2f'), '% (planta ', num2str(idx_min), ')']);
    disp('=====');
end

```

Compacitats de les plantes per a la imatge Model:

Planta 1 - Àrea de la planta: 162990 pixels

Planta 1 - Àrea de l'envoltura convexa: 292950 pixels

Planta 1 - Compacitat: 55.64%

Àrea mitjana de les plantes: 162990.00 pixels

Àrea mitjana de les envoltures convexes: 292950.00 pixels

Compacitat mitjana de les plantes: 55.64%

Planta amb la major compacitat: 55.64% (planta 1)

Planta amb la menor compacitat: 55.64% (planta 1)

=====

Compacitats de les plantes per a la imatge Imatge 1:

Planta 1 - Àrea de la planta: 79961 pixels

Planta 1 - Àrea de l'envoltura convexa: 135728 pixels

Planta 1 - Compacitat: 58.91%

Planta 2 - Àrea de la planta: 53101 pixels

Planta 2 - Àrea de l'envoltura convexa: 90274 pixels

Planta 2 - Compacitat: 58.82%

Planta 3 - Àrea de la planta: 115217 pixels

Planta 3 - Àrea de l'envoltura convexa: 196904 píxels
Planta 3 - Compacitat: 58.51%

Planta 4 - Àrea de la planta: 69440 píxels
Planta 4 - Àrea de l'envoltura convexa: 118393 píxels
Planta 4 - Compacitat: 58.65%

Planta 5 - Àrea de la planta: 98801 píxels
Planta 5 - Àrea de l'envoltura convexa: 161920 píxels
Planta 5 - Compacitat: 61.02%

Àrea mitjana de les plantes: 83304.00 píxels
Àrea mitjana de les envoltures convexes: 140643.80 pixels
Compacitat mitjana de les plantes: 59.18%
Planta amb la major compacitat: 61.02% (planta 5)
Planta amb la menor compacitat: 58.51% (planta 3)

=====

Compacitats de les plantes per a la imatge Imatge 2:

Planta 1 - Àrea de la planta: 86979 píxels
Planta 1 - Àrea de l'envoltura convexa: 166740 píxels
Planta 1 - Compacitat: 52.16%

Planta 2 - Àrea de la planta: 98755 píxels
Planta 2 - Àrea de l'envoltura convexa: 162616 píxels
Planta 2 - Compacitat: 60.73%

Planta 3 - Àrea de la planta: 86836 píxels
Planta 3 - Àrea de l'envoltura convexa: 142811 pixels
Planta 3 - Compacitat: 60.80%

Planta 4 - Àrea de la planta: 75111 píxels
Planta 4 - Àrea de l'envoltura convexa: 133012 píxels
Planta 4 - Compacitat: 56.47%

Planta 5 - Àrea de la planta: 73968 píxels
Planta 5 - Àrea de l'envoltura convexa: 119967 píxels
Planta 5 - Compacitat: 61.66%

Planta 6 - Àrea de la planta: 76204 píxels
Planta 6 - Àrea de l'envoltura convexa: 134325 píxels
Planta 6 - Compacitat: 56.73%

Àrea mitjana de les plantes: 82975.50 píxels
Àrea mitjana de les envoltures convexes: 143245.17 pixels
Compacitat mitjana de les plantes: 58.09%
Planta amb la major compacitat: 61.66% (planta 5)
Planta amb la menor compacitat: 52.16% (planta 1)

=====

Compacitats de les plantes per a la imatge Imatge 3:

Planta 1 - Àrea de la planta: 256147 píxels
Planta 1 - Àrea de l'envoltura convexa: 742716 píxels
Planta 1 - Compacitat: 34.49%

Planta 2 - Àrea de la planta: 234900 píxels
Planta 2 - Àrea de l'envoltura convexa: 676067 píxels
Planta 2 - Compacitat: 34.75%

Planta 3 - Àrea de la planta: 300056 píxels
Planta 3 - Àrea de l'envoltura convexa: 720207 píxels
Planta 3 - Compacitat: 41.66%

Planta 4 - Àrea de la planta: 280865 píxels
Planta 4 - Àrea de l'envoltura convexa: 694549 píxels
Planta 4 - Compacitat: 40.44%

```
-----  
Àrea mitjana de les plantes: 267992.00 píxels  
Àrea mitjana de les envoltures convexes: 708384.75 píxels  
Compacitat mitjana de les plantes: 37.83%  
Planta amb la major compacitat: 41.66% (planta 3)  
Planta amb la menor compacitat: 34.49% (planta 1)  
=====
```

6. Estudi de creixement en els diferents ambients

En aquest punt, amb totes les dades anteriors recollides procedim a estudiar el creixement i decreixement relatiu de les plantes en els seus diferents ambients. El codi següent compara el creixement i la variació en compacitat de les plantes a través de diverses imatges, utilitzant la planta model. Primer es recuperen l'àrea mitjana i la compacitat mitjana de la planta model. A continuació, s'inicialitzen dos vectors per emmagatzemar el creixement relatiu en àrea i la diferència en compacitat per a les imatges posteriors. Es calcula el creixement relatiu de l'àrea de cada planta en comparació amb la planta model, així com la diferència en la compacitat mitjana. El creixement relatiu es calcula com el canvi percentual de l'àrea, mentre que la diferència en compacitat es calcula en percentatge respecte al valor de la planta model. Finalment, el codi mostra els resultats de la comparació, incloent les variacions en àrea i compacitat per a cada ambient, indicant si són positives o negatives. Això ens permet veure com canvia l'estructura i la forma de les plantes en diferents condicions.

```
% Obtenir les dades de la planta model (primera imatge)  
area_model = area_mitjana_plantes(1); % Àrea mitjana de la planta model  
compacitat_model = compacitat_mitjana(1) * 100; % Compacitat mitjana de la planta  
model (en percentatge)  
  
% Inicialitzar els resultats  
creixement_relatiu = zeros(1, length(imatges) - 1); % Creixement relatiu en àrea  
comparacio_compacitat = zeros(1, length(imatges) - 1); % Diferència en compacitat  
mitjana  
  
% Iterar sobre les altres imatges per calcular el creixement relatiu i la  
diferència de compacitats  
for i = 2:length(imatges)  
    % Calcular el creixement relatiu en àrea respecte a la planta model  
    creixement_relatiu(i - 1) = ((area_mitjana_plantes(i) - area_model) /  
    area_model) * 100; % En percentatge  
  
    % Calcular la diferència en compacitat mitjana respecte a la planta model  
    comparacio_compacitat(i - 1) = (compacitat_mitjana(i) * 100) -  
    compacitat_model; % En percentatge  
end  
  
% Mostrar els resultats  
disp('Comparació amb la planta model:');
```

Comparació amb la planta model:

```
disp(['Àrea mitjana de la planta model: ', num2str(area_model, '%.2f'), ' pixels']);
```

Àrea mitjana de la planta model: 162990.00 pixels

```
disp(['Compacitat mitjana de la planta model: ', num2str(compacitat_model, '%.2f'), '%']);
```

Compacitat mitjana de la planta model: 55.64%

```
for i = 2:length(imatges)
    % Obtenir el signe positiu o negatiu per al creixement relatiu
    if creixement_relatiu(i - 1) >= 0
        signe_creixement = '+';
    else
        signe_creixement = '';
    end

    % Obtenir el signe positiu o negatiu per a la diferència de compacitats
    if comparacio_compacitat(i - 1) >= 0
        signe_compacitat = '+';
    else
        signe_compacitat = '';
    end

    % Mostrar els resultats per a cada imatge
    disp(['Ambient ', noms_imatges{i}, ':']);
    disp([' Creixement relatiu en àrea: ', signe_creixement,
num2str(creixement_relatiu(i - 1), '%.2f'), '%']);
    disp([' Diferència en compacitat mitjana: ', signe_compacitat,
num2str(comparacio_compacitat(i - 1), '%.2f'), '%']);
end
```

Ambient Imatge 1:

Creixement relatiu en àrea: -48.89%
Diferència en compacitat mitjana: +3.55%

Ambient Imatge 2:

Creixement relatiu en àrea: -49.09%
Diferència en compacitat mitjana: +2.46%

Ambient Imatge 3:

Creixement relatiu en àrea: +64.42%
Diferència en compacitat mitjana: -17.80%

Amb aquestes dades, podem veure com la compacitat de les plantes no varia significativament en els ambients de les Imatges 1 i 2, experimentant un lleu creixement. Això ens indica que en quant a la forma de les fulles, a la seva estructura general o a la quantitat de fulles, aquestes es presenten similars que en la planta Model. Pel que fa també en els dos primers ambients, es veu una devallada significativa en la mida de les plantes, presentant decreixements en ambdos casos propers al 50%. Això ens indica que en aquests ambients la mida de les plantes s'ha reduït a la meitat respecte a la planta model. Tenim en compte que hem mesurat totes les plantes prenent com a mesura els píxels que ocupen en la imatge, assumint que totes aquestes s'han realitzat pel mateix dispositiu i a la mateixa distància.

Finalment, podem observar que les plantes de l'ambient 3 han experimentat un fort creixement respecte a la planta model, pròxim al 65%. Per la seva banda, però, podem veure com han reduït la seva compacitat

lleugerament, al voltant d'un 20%. D'aquesta manera, podem deduir que s'ha reduït el seu numero de fulles o s'han aprimat aquestes, tot i augmentar de forma considerable en tamany, és a dir, s'han allargat.

7. Estudi del color

L'últim apartat fa referència a l'estudi del color de les plantes. Els següent codi comparar els colors de les plantes en les diferents imatges (ambients), normalitzant-les prèviament per ajustar les diferències de llum existents en les imatges i calcular la distància de color en funció de la planta model. Primerament, inicialitza diverses variables per emmagatzemar els colors mitjans de les plantes i de l'ambient. Per cada imatge, es calcula la mitjana i la desviació estàndard de cada canal RGB de la imatge per a normalitzar la imatge i ajustar les condicions de llum abans de realitzar la comparació. Això ajuda a fer que les comparacions siguin més consistentes, independentment de les diferències en la il·luminació entre les imatges.

Després de normalitzar la imatge, el codi extreu els píxels de les plantes definits prèviament i calcula el color mitjà de cada planta. Per cada planta, els píxels es converteixen a coordenades de la imatge, i es recuperen els valors RGB per aquests píxels. Llavors, es calcula el color mitjà de la planta, emmagatzemant-lo per cada imatge. També es calcula el color mitjà de tot l'ambient (mitjana dels colors de totes les plantes) per cada imatge, i aquest color s'emmagatzema per comparar-lo posteriorment amb el color de la planta model.

Finalment, el codi compara el color mitjà de cada ambient amb el color de la planta model mitjançant la distància euclidiana entre els components RGB (roig, verd, blau) de cada imatge i la model. Es calcula la diferència total de color i es mostra la distància en cada component RGB, així com un percentatge que indica la diferència total de color respecte a la planta model. Aquesta informació es visualitza per cada ambient, ajudant a avaluar com es compara el color de les plantes en les diferents imatges amb el color de la planta model.

```
% Inicialitzar les variables per emmagatzemar els colors mitjans
colors_mitjans = cell(1, length(imatges)); % Color mitjà de cada planta a cada
imatge
color_mitja_ambient = zeros(length(imatges), 3); % Color mitjà de totes les
plantes per ambient
color_model = zeros(1, 3); % Color de la planta model

% Definir el valor de normalització per a cada imatge (potser es vol utilitzar la
mitjana i la desviació)
for i = 1:length(imatges)
    % Obtenir la imatge original
    imatge_actual = imatges{i};

    % Convertir la imatge a tipus 'double' per evitar errors en les funcions de
    mitjana i desviació
    imatge_actual = double(imatge_actual); % Convertir la imatge a double per
    càlculs

    % Normalitzar la imatge per ajustar les condicions de llum
    % Calcular la mitjana i la desviació estàndard de la imatge per cada canal de
    color
    mitjana_imatge = mean(imatge_actual, [1 2]);
    desviacio_imatge = std(imatge_actual, 0, [1 2]);
```

```

% Normalitzar la imatge (subtraiem la mitjana i dividim per la desviació
estàndard)
imatge_normalitzada = (imatge_actual - mitjana_imatge) ./ desviacio_imatge;

% Obtenir els píxels de les plantes per a la imatge normalitzada
planta_pixels = pixels_planta{i}; % Píxels de cada planta

% Inicialitzar una llista per emmagatzemar els colors mitjans de les plantes
colors_mitjans_plantes = zeros(length(planta_pixels), 3);

for j = 1:length(planta_pixels)
    % Obtenir els índexs dels píxels de la planta j
    pixels_planta_j = planta_pixels{j};

    % Obtenir les coordenades (x, y) dels píxels
    [y_coords, x_coords] = ind2sub(size(imatge_normalitzada(:, :, 1)),
pixels_planta_j); % Convertir índexs a coordenades

    % Extreure els colors RGB de la planta
    rgb_valors = zeros(length(pixels_planta_j), 3); % Valors RGB per a cada
pixel
    for k = 1:length(pixels_planta_j)
        rgb_valors(k, :) = imatge_normalitzada(y_coords(k), x_coords(k), :); % 
Valors RGB del píxel
    end

    % Calcular el color mitjà de la planta
    colors_mitjans_plantes(j, :) = mean(rgb_valors, 1);
end

% Guardar el color mitjà de les plantes per a la imatge normalitzada
colors_mitjans{i} = colors_mitjans_plantes;

% Calcular el color mitjà per a tot l'ambient (mitjana de totes les plantes)
color_mitja_ambient(i, :) = mean(colors_mitjans_plantes, 1);
end

% Guardar el color mitjà de la planta model (primera imatge)
color_model = color_mitja_ambient(1, :);

for i = 1:length(imatges)
    % Obtenir els components R, G i B per al color mitjà de l'ambient
    color_ambient = color_mitja_ambient(i, :);
    R_ambient = color_ambient(1);
    G_ambient = color_ambient(2);
    B_ambient = color_ambient(3);

    % Obtenir els components R, G i B per al color model (planta model)
    R_model = color_model(1);

```

```

G_model = color_model(2);
B_model = color_model(3);

% Calcular la distància euclidiana per a cada component RGB
diferencia_R = R_ambient - R_model;
diferencia_G = G_ambient - G_model;
diferencia_B = B_ambient - B_model;

% Calcular la distància euclidiana total
diferencia_color = sqrt(diferencia_R^2 + diferencia_G^2 + diferencia_B^2);

% Calcular el percentatge de diferència de color amb la planta model
percentatge_diferencia_color = (diferencia_color / 433) * 100;

% Mostrar la distància de cada component i la diferència total en percentatge
disp(['Ambient ', nom_imatges{i}, ':']);
disp([' Color mitjà de l''ambient - R: ', num2str(R_ambient, '%.2f'), ', G: ', num2str(G_ambient, '%.2f'), ', B: ', num2str(B_ambient, '%.2f')]);
disp([' Color model - R: ', num2str(R_model, '%.2f'), ', G: ', num2str(G_model, '%.2f'), ', B: ', num2str(B_model, '%.2f')]);
disp([' Diferència en R: ', num2str(diferencia_R, '%.2f')]);
disp([' Diferència en G: ', num2str(diferencia_G, '%.2f')]);
disp([' Diferència en B: ', num2str(diferencia_B, '%.2f')]);
disp([' Diferència total de color (distància euclidiana): ', num2str(diferencia_color, '%.2f')]);
disp([' Diferència de color amb la planta model: ', num2str(percentatge_diferencia_color, '%')]);
end

```

Ambient Model:

Color mitjà de l'ambient - R: 0.86, G: 1.39, B: -0.21
Color model - R: 0.86, G: 1.39, B: -0.21
Diferència en R: 0.00
Diferència en G: 0.00
Diferència en B: 0.00
Diferència total de color (distància euclidiana): 0.00
Diferència de color amb la planta model: 0.00%

Ambient Imatge 1:

Color mitjà de l'ambient - R: 0.64, G: 1.11, B: -0.09
Color model - R: 0.86, G: 1.39, B: -0.21
Diferència en R: -0.22
Diferència en G: -0.28
Diferència en B: 0.12
Diferència total de color (distància euclidiana): 0.38
Diferència de color amb la planta model: 0.09%

Ambient Imatge 2:

Color mitjà de l'ambient - R: 1.38, G: 1.70, B: 0.30
Color model - R: 0.86, G: 1.39, B: -0.21
Diferència en R: 0.51
Diferència en G: 0.31
Diferència en B: 0.51
Diferència total de color (distància euclidiana): 0.79
Diferència de color amb la planta model: 0.18%

Ambient Imatge 3:

Color mitjà de l'ambient - R: 0.07, G: 0.72, B: -1.33
Color model - R: 0.86, G: 1.39, B: -0.21

Diferència en R: -0.79
Diferència en G: -0.67
Diferència en B: -1.12
Diferència total de color (distància euclidiana): 1.53
Diferència de color amb la planta model: 0.35%

Podem observar que cap ambient varia de forma destacada sobre la planta model. L'ambient 3 és el que presenta més variacions destacant la reducció que pateixen les seves plantes en el canal blau.

Finalment, per poder visualitzar si alguna planta, en algun ambient té un color predominant significativament diferent a la model apliquem el següent codi, on podem veure que no és el cas de les nostres plantes, on la que més distanciada es troba de la model és la planta 1 de la imatge 3 amb una distància euclidiana de 1.56 respecte a la planta model.

```
% Iterar sobre totes les plantes i identificar les que es diferencien més de 1
unitat del model
for i = 1:length(imatges)
    % Comprovar que hi ha plantes a la imatge
    if ~isempty(colors_mitjans{i})
        % Iterar sobre totes les plantes d'aquesta imatge
        for j = 1:size(colors_mitjans{i}, 1) % Comprovar les dimensions abans
d'accendir a l'element
            % Obtenir el color mitjà de la planta j
            color_planta = colors_mitjans{i}(j, :);

            % Calcular la distància euclidiana entre el color de la planta i el
model
            diferencia_planta = sqrt((color_planta(1) - color_model(1))^2 + ...
                (color_planta(2) - color_model(2))^2 + ...
                (color_planta(3) - color_model(3))^2);

            % Si la distància és major que el llindar (1 unitat), imprimir el
resultat
            if diferencia_planta > 1
                disp(['Planta ', num2str(j), ' a l''ambient ', noms_imatges{i}, ...
                    ' es diferencia més de 1 unitat (distància euclidiana: ', ...
                    num2str(diferencia_planta, '%.2f'), ').']);
            end
        end
    end
end
```

Planta 3 a l'ambient Imatge 2 es diferencia més de 1 unitat (distància euclidiana: 1.21).
Planta 4 a l'ambient Imatge 2 es diferencia més de 1 unitat (distància euclidiana: 1.04).
Planta 1 a l'ambient Imatge 3 es diferencia més de 1 unitat (distància euclidiana: 1.56).
Planta 2 a l'ambient Imatge 3 es diferencia més de 1 unitat (distància euclidiana: 1.50).
Planta 3 a l'ambient Imatge 3 es diferencia més de 1 unitat (distància euclidiana: 1.55).
Planta 4 a l'ambient Imatge 3 es diferencia més de 1 unitat (distància euclidiana: 1.50).