

Memoria P2

Mateo Amado Ares

El trabajo de segmentación y evaluación de crecimiento de plantones presenta 3 objetivos principales. El primero de ellos es separar y segmentar cada una de las plantas, el segundo segmentar cada una de las hojas por separado y el tercero proporcionar una evaluación adecuada de las segmentaciones anteriores. No consideré metodologías basadas en machine learning o deep learning al tener un dataset reducido (6 imágenes).

Para el primer objetivo, separé los objetos de interés del fondo utilizando el color. Apoyándome en el formato HSV, que facilita seleccionar los verdes con el hue, obtuve los rangos de color que correspondían a las plantas. Encontré estos valores ayudándome de páginas como <https://colorizer.org/> y mediante prueba y error (hue 0.1-0.5, saturation 0.35-1.0, value 0.25-1). Con estos valores genero una máscara binaria sobre la que aplico las operaciones morfológicas de `remove_small_holes` y `remove_small_objects` para deshacerme de algo de ruido y suavizar los bordes. ^[1]

Para segmentar cada una de las plantas por separado intenté encontrar bounding boxes mediante conectividad, pero los resultados no eran consistentes (las plantas se sobreponían o partes de ellas estaban desconectadas). Abandoné esta aproximación a favor de una más sencilla basada en la posición de las plantas, ya que todas las imágenes son un grid 4x5 de plantas en lugares similares. En función al número de píxeles "verdes" en cada una de estas celdas determino si esta está ocupada por una planta y su tamaño. En algún caso la planta ha crecido por fuera de su celda, esto lo reconozco mirando si alguno de los bordes de la celda tiene píxeles de planta y en ese caso amplío la celda en todas direcciones para que abarque la planta entera. Las celdas ampliadas tienden a contener partes de otras plantas, por lo que filtro de forma que solo me quede con el elemento conectado más grande. ^[2]

Para el segundo problema (segmentar las hojas) decidí centrarme en las plantas más pequeñas, ya que son más fácilmente diferenciables. Denomino planta grande aquellas cuyas hojas se salen de la maceta donde fue plantada, pequeñas el resto.

En una primera aproximación intenté determinar las regiones correspondientes a cada hoja a partir de los bordes obtenidos mediante Canny, pero aún jugando con diferentes alteraciones de la imagen (ecualización de histograma, contraste, filtro gaussiano...) y parámetros, los bordes no son lo suficientemente completos para determinar una región por sí solos. Para intentar mejorar la visibilidad de los bordes probé a hacer PCA (Análisis de componentes principales) sobre el espacio RGB para determinar el vector de colores que me interesaban y poder maximizar el contraste en él, pero no debí implementarlo bien y no dió buenos resultados. Otra opción que probé fue a completar los bordes usando el algoritmo de Hough con elipses, que es la forma más parecida a una hoja que encontré, pero con bastantes malos resultados también (además de ser computacionalmente bastante intensivo).

Tras tener poco éxito con los métodos anteriores, probé a crear ventanas más pequeñas con hojas fácilmente diferenciables o grupos de hojas más complicados de determinar para poder procesarlas por separado. Para ellos utilicé la técnica de [medial axis transform \(MAT\)](#), que realiza erosiones reduciendo los objetos a líneas que mantienen la longitud y la conectividad del objeto original. Los componentes conectados restantes coincidirán aproximadamente con cada hoja (o varias hojas juntas) que podré analizar en mayor detalle^[3]. Esta técnica dió algunos resultados con las imágenes de plantas más pequeñas, pero con las grandes no es útil. Escogí MAT sobre otras técnicas similares como skeletonize o thinning, ya que entre otras cosas me daba el atributo de distancia y generalmente funcionaba mejor. Además de este filtrado, modifiqué la imagen (eliminación de objetos pequeños y agujeros pequeños mediante operaciones morfológicas) en otras ocasiones a lo largo del procesado para intentar evitar problemas por ruido, al que este algoritmo es bastante sensible.

Por cada una de estas ventanas que tengo obtengo las regiones conectadas que hay dentro (eliminando las que son muy pequeñas para ser una hoja) y pinto la hoja que le correspondería en la imagen original^[4]. Este método tiene varios problemas, ya que en algunos casos las hojas no están al completo dentro de la ventana escogida lo que lleva a que acaben cortadas o separadas en dos. Otro problema es que las hojas de tamaño muy pequeño, que se suelen encontrar en el centro de la planta, no serán nunca reconocidas por separado^[5]. De la misma forma este método no da buenos resultados con las imágenes de plantas grandes. Finalmente, guardo toda la información recogida en diferentes estructuras de datos donde sería sencillo acceder a la información para trabajar sobre ella o exportarla. En `size_grid` almaceno el tamaño de cada planta en función a su posición y en `cell_grid` almaceno cada una de la planta con cada hoja pintada de un color distinto. Esta metodología da resultados "pasables" en ocasiones, pero generalmente era poco fiable^[6]. Tras tener poco éxito utilizando técnicas basadas en bordes, ya que estos no son muy explícitos en la imagen original, decidí utilizar watershed, que al estar basado en imágenes binarias me evita esa problemática.

Para esta nueva aproximación calculé la distancia euclídea de cada píxel y localicé los máximos locales, que corresponderán con los marcadores para watershed. Uno de los parámetros que me ayudó a obtener buenos marcadores es el de distancia mínima, que coge picos más separados y evita que se formen regiones muy pequeñas. Para obtener mejores resultados varíó la distancia mínima utilizada en función al tamaño de las plantas previamente calculado. Finalmente aplico el algoritmo watershed y genero la imagen etiquetada.^[7]

Tras la mejora que supuso utilizar watershed frente a los otros métodos en el proceso de segmentación de hojas intenté aplicarlo a la segmentación de las plantas, utilizando una distancia mínima entre regiones mayor (200-250). Da buenos resultados exceptuando los casos de plantas grandes, en donde mi primera aproximación es más precisa a la hora de saber si determinada maceta contiene una planta.^[8]

Para el tercer objetivo (evaluación), aplicado al primer problema (segmentación de plantas), comparo la imagen proporcionada como ground truth con la obtenida por mí en binario. Para ello por un lado genero la matriz de confusión y calculo los valores de precision, f1, recall y accuracy. Con ellos puedo dibujar la curva ROC^[9]. A su vez también calculo el structural similarity index, para poder evaluar la forma de los bordes y no solo cada píxel por separado. Tabla de resultados: [Figure_1]

Evaluar la segmentación de las hojas es más complejo, ya que la imagen proporcionada como ground truth repite colores por lo que la segmentación de las hojas no es trivial. Por conectividad también se encuentran problemas en los casos en los que una hoja no está conectada a la planta o dos hojas de plantas diferentes se superponen. Opté por evaluar de los bordes de las imágenes (obtenidos con Canny) usando las mismas métricas que el primer problema. Tabla de resultados: [Figure_2]

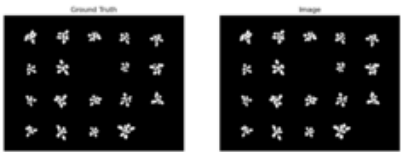
De estas tablas podemos concluir que la segmentación de las plantas es adecuada, mientras que la de las hojas tiene bastante que mejorar, especialmente en las plantas grandes (imágenes 3 y 6). Una posible mejora sería utilizar un sistema de votado como RANSAC para completar los bordes incompletos de la primera aproximación, y utilizarlos para general mejores marcadores para watershed. Otra forma de obtener mejores bordes sería aumentar el contraste en los colores verdes que nos interesan. También sería interesante obtener mejores etiquetas para las hojas del ground_truth y así poder hacer una evaluación más fiable.

image:	1	2	3	4	5	6
ssim:	0.9462	0.9565	0.8998	0.9191	0.9028	0.9174
Precision:	0.7708	0.7429	0.9196	0.8602	0.7983	0.9262
Accuracy:	0.9801	0.9842	0.9626	0.9713	0.9620	0.9712
Recall:	0.9999	.9999	0.9999	0.9992	0.9999	0.9997
F1-score:	0.8705	0.8525	0.9581	0.9245	0.8878	0.9616

Figure_1: tabla evaluacion segmentación de plantas

image:	1	2	3	4	5	6
ssim:	0.9721	0.9791	0.9429	0.9588	0.9429	0.9506
Precision:	0.2397	0.2831	0.2255	0.2992	0.1730	0.2386
Accuracy:	0.9910	0.9931	0.9745	0.9852	0.9822	0.9787
Recall:	0.2378	0.2776	0.1818	0.2818	0.1617	0.1735
F1-score:	0.2387	0.2803	0.2013	0.2903	0.1671	0.2009

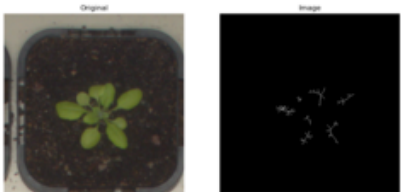
Figure_2: tabla evaluacion segmentación de hojas



1. *comparación segmentación de plantas y ground truth* ↩



2. *ejemplo de segmentación de una planta grande* ↩



3. *ejemplo esqueleto obtenido por primera aproximacion* ↩



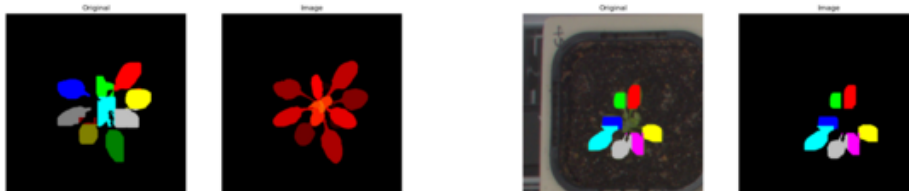
4. obtención de bordes de una ventana de una planta ↵

ejemplo de



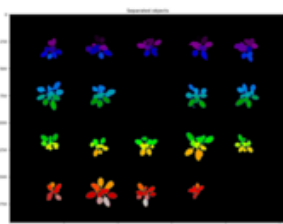
5. obtención de bordes de una ventana de una planta2 ↵

ejemplo de



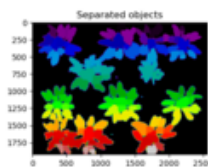
6. segmentación de hojas 1era aproximación ↵

resultados



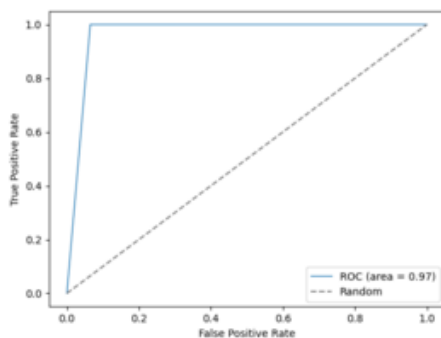
7. resultados waterleaf segmentación de hojas ↵

ejemplos



8. resultados waterleaf segmentación de plantas ↵

ejemplos



9. ejemplo curva ROC ↵