

## Practica 2 Visión Artificial – Javier Fernández Rozas

### Segmentación y evaluación de crecimiento de plantones

#### 1 - Separar y segmentar cada una de las plantas, hasta un máximo de 5 × 4 por bandeja

Para la realización de este apartado, lo primero que se pensó fue utilizar un mecanismo de detección de bordes (Por ejemplo, Canny), para detectar los bordes de las plantas (Esto fue mezclado con operadores de Blur como el de medianas y Gauss, para intentar detectar mejor los bordes) . Sin embargo, esto no funcionaría, ya que también detecta los bordes de la bandeja, así como diversos trozos de tierra sueltos.

Tras este intento, se pasó a intentar a realizar una segmentación basada en regiones (Crecimiento de regiones, en este caso). Sin embargo, esto, aunque al principio llevó a resultados medianamente satisfactorios (pruebas manuales), al intentar generalizarlo a las distintas imágenes, la localización de las semillas óptima era distinta de una imagen a otra, por lo que cuando una imagen se segmentaba decente, la otra daba resultados subóptimos.

Por lo tanto, ya que se puede observar que los colores de las plantas son casi únicos en las fotos, se decidió filtrar la imagen por colores. Antes de este paso, para ver con mas claridad los colores, se intentó hacer una ecualización de histograma en los verdes, pero no lleva a ninguna mejora significativa. Por lo tanto, midiendo los valores tanto RGB como HSV de los pixeles, se eliminan aquellos que no cumplen con las características que necesitamos. Los resultados de estos filtros se encuentran en las carpetas 1-final y 2-final, respectivamente.

Al observar el resultado de estos filtros, se observa que las plantas se conservan bastante bien, exceptuando algunas hojas concretas, y se mantiene un poco de ruido. En este punto, se decide realizar una conversión de la imagen a binario, y aplicar filtros morfológicos (Apertura y cierre) para quitar el ruido. Se probaron con varios kernels (Cuadrado, elipse, cruz...), además de diferentes tamaños. La principal problemática aquí es que el ruido en las imágenes 3 y 6 es mayor, por lo que necesitarían un kernel mayor, pero este elimina información importante en los tallos de las imágenes 1 y 4. Los resultados de este paso se pueden ver en 4-final.

Tras esto, ya que aún sigue habiendo ruido, y aplicar más filtros morfológicos dañaría la imagen, se decide eliminar los conjuntos de pixeles aislados que no lleguen a un cierto umbral. Tras esto, tenemos una imagen binaria, sin casi ruido, que podemos utilizar como máscara para segmentar las plantas del resto de la imagen. Estos resultados se encuentran en 5-final (máscaras) y en 6-final (Imágenes con la máscara aplicada).

Al ver los resultados, observamos que, principalmente, las imágenes 1 y 4 no presentan ruido, pero tienen la pérdida de píxeles en los tallos, generados en los pasos de filtros de color y morfológicos. Las imágenes 2 y 5 presentan ruido casi nulo, pero tienen una ligera pérdida de píxeles en algunas hojas puntuales. Las imágenes 3 y 6 son aquellas que presentan una mayor cantidad de ruido, aunque no excesivo, y también alguna pérdida de píxeles.

Algunas posibles soluciones: A las imágenes 1 y 4, para mantener los tallos finos, podría aplicarse filtros morfológicos más pequeños, ya que el proceso de apertura-cierre elimina varios de los tallos, o en los filtros por color ser un poco más laxos, y ser más permisivos, aunque esto provocaría más ruido. Otra forma podría ser realizar una técnica híbrida entre filtros por color más detección de bordes, ya que eliminaríamos el ruido, a su vez preservando los tallos.

Para el resto de las imágenes, también estaría bien ser más laxos con el filtrado por color, ya que se pierden píxeles en algunas hojas. Sin embargo, esto acentuaría el ruido en demasía, por lo que no es una solución óptima. Un método para que esto fuera posible sería aplicar la solución propuesta en <https://www.sciencedirect.com/science/article/pii/S2214317318301562>, que antes de la segmentación, realiza una mejora de la imagen en el plano V de una imagen HSV, lo cual hace que las plantas sean más fáciles de detectar.

## 2 - Para cada una de las plantas, segmentar cada una de sus hojas por separado.

Para la realización de este apartado, se probaron varios métodos:

La primera intención fue realizar una segmentación mediante la detección de bordes de Canny, y dibujar + colorear los bordes en base a esto. Sin embargo, esto no dio resultado, ya que no detectaba los bordes interiores, o para detectarlos, había que bajar tanto el umbral que detectaba mucho ruido. Ni usando filtros de suavizado se pudo solucionar este problema.

En segunda opción, se trato de seguir los pasos mencionados en [https://www.researchgate.net/publication/337561634 Segmentation of single and overlapping leaves by extracting appropriate contours](https://www.researchgate.net/publication/337561634_Segmentation_of_single_and_overlapping_leaves_by_extracting_appropriate_contours). Sin embargo, al no ser capaz de implementar con éxito la estilización, y la proporcionada por OpenCV no ser capaz de extraer los bordes interiores, se tuvo que dejar de lado.

Como opción elegida, aunque no llevada a cabo con efectividad ni en su totalidad, se uso el crecimiento de regiones. Usando 4-conectividad, se codifico un algoritmo para la aplicación de crecimiento de regiones en cada una de las 20 sub-imágenes creadas al dividir las 5x4 macetas para plantas.

Esta opción, si hubiera sido llevada a cabo con más tiempo, podría haber llegado a ser optima, ya que, tras algunos experimentos, se podía observar que si aislaba correctamente las hojas (centrando los puntos manualmente para cada imagen). Sin embargo, por falta de tiempo, los puntos fueron establecidos en las esquinas, y se trata de eliminar la mayor parte del fondo, y separar las hojas en la medida de lo posible. Solo funciona decentemente en las imágenes 1,2,4,5, ya que en las 3 y 6 la colocación de los puntos no es, ni de lejos, la ideal.

Tras extraer las hojas, estas son coloreadas y guardadas.

## 3 - Proporcionar una evaluación cuantitativa adecuada de las segmentaciones 1 y 2.

Segmentación 1:

Se usan las imágenes dadas en materiales, y pasadas a binario, como ground truth, sobre las máscaras creadas en el apartado 1 (carpeta 5-final). Métricas calculadas en el archivo métricas.ipynb. Se usan VP, VN, FP, FN, además de precisión, sensibilidad, especificidad, coeficiente de similitud de Dice y similitud.

Imagen	Numero de pixeles valor 0	Numero de pixeles valor 255
Ground Truth 1	4586194	329006
Segmentada 1	4580973	334227
Ground Truth 2	4051986	863214
Segmentada 2	4064760	850440
Ground Truth 3	3145403	1769797
Segmentada 3	3150968	1764232
Ground Truth 4	4691303	223897
Segmentada 4	4686179	229021
Ground Truth 5	4176561	738639
Segmentada 5	4153606	761594
Ground Truth 6	2817729	2097471
Segmentada 6	2804752	2110448

Imagen 1:

VP: 324617, FP: 9610, VN: 4576584, FN: 4389

Precisión: 97.12470865609302%

Sensibilidad: 98.66598177540835%

Especificidad: 99.79045805737829%

Similitud de Dice: 0.9788927873009937

Similitud: 0.9775869402625511

#### Imagen 2:

VP: 841682, FP: 8758, VN: 4043228, FN: 21532

Precisión: 98.97018014204411%

Sensibilidad: 97.50560116031482%

Especificidad: 99.78385907552494%

Similaridad de Dice: 0.9823243198451963

Similitud: 0.9809178689459707

#### Imagen 3:

VP: 1744077, FP: 20155, VN: 3125248, FN: 25720

Precisión: 98.85757655455745%

Sensibilidad: 98.5467259804373%

Especificidad: 99.35922360346194%

Similaridad de Dice: 0.9870190652085764

Similitud: 0.9869287783573364

#### Imagen 4:

VP: 221064, FP: 7957, VN: 4683346, FN: 2833

Precisión: 96.52564611978815%

Sensibilidad: 98.73468603866957%

Especificidad: 99.83038827379089%

Similaridad de Dice: 0.9761767030676635

Similitud: 0.9738541071048616

#### Imagen 5:

VP: 736285, FP: 25309, VN: 4151252, FN: 2354

Precisión: 96.6768383154279%

Sensibilidad: 99.68130575287792%

Especificidad: 99.39402297727725%

Similaridad de Dice: 0.9815608642124257

Similitud: 0.976393888921556

#### Imagen 6:

VP: 2084986, FP: 25462, VN: 2792267, FN: 12485

Precisión: 98.79352630341994%

Sensibilidad: 99.40475935066564%

Especificidad: 99.09636448359655%

Similaridad de Dice: 0.9909820032182178

Similitud: 0.9904871397278223

### Segmentación 2:

#### Imagen 1:

VP: 68265, FP: 4540, VN: 3783954, FN: 260741

Precision: 93.7641645491381%

Sensibilidad: 20.748861722886513%

Especificidad: 99.88016346337093%

Similaridad de Dice: 0.33978661609562705

Similitud: 0.43787774630475684

#### Imagen 2:

VP: 201771, FP: 11, VN: 3254579, FN: 661139

Precision: 99.9945485722215%

Sensibilidad: 23.382623912111345%

Especificidad: 99.99966201579923%

Similaridad de Dice: 0.37902229001438914  
Similitud: 0.45823333674398914

Imagen 3:

VP: 301618, FP: 1039, VN: 2379765, FN: 1435078  
Precision: 99.65670709747337%  
Sensibilidad: 17.36734581066577%  
Especificidad: 99.95635928031035%  
Similaridad de Dice: 0.2957977358505369  
Similitud: 0.4156938564249465

Imagen 4:

VP: 63072, FP: 4505, VN: 3889098, FN: 160825  
Precision: 93.33353063912277%  
Sensibilidad: 28.170096070961204%  
Especificidad: 99.88429739755183%  
Similaridad de Dice: 0.432779596121781  
Similitud: 0.4899031017446559

Imagen 5:

VP: 235304, FP: 5180, VN: 3373681, FN: 503335  
Precision: 97.8460105454001%  
Sensibilidad: 31.856427835519106%  
Especificidad: 99.84669390069612%  
Similaridad de Dice: 0.48064237077466265  
Similitud: 0.5179115175662075

Imagen 6:

VP: 148416, FP: 133, VN: 2079748, FN: 1889203  
Precision: 99.91046725322957%  
Sensibilidad: 7.283795449492766%  
Especificidad: 99.99360540338606%  
Similaridad de Dice: 0.13577730531230903  
Similitud: 0.3443971246873515

Sin embargo, estas estadísticas, en este caso, solo sirven para comparar la “mascara” creada al segmentar las hojas, y no la segmentación de las hojas en sí. Para esta evaluación, se necesitaría coger las imágenes de ground truth, detectar el número de hojas (al ser colores distintos no sería tan complicado), y compararlos con el numero de hojas obtenido en nuestro método de segmentación. Además, se podría ir hoja a hoja, comparando diferentes parámetros (área usando cv2.area, los VP, VN... hoja a hoja...), medidas las cuales darían unas medidas mucho mas apropiadas para la evaluación de este algoritmo.

#### 4-Estructura de los ficheros y funcionamiento:

- final, final2 y métricas son los notebooks correspondientes a los ejercicios 1,2 y 3
- (1-9)-final son las carpetas donde se guardan los resultados de la ejecución del código
- ej2\_1, y ej2\_2 son intentos no satisfactorios del ejercicio 2.
- En intendo\_dos\_fallos están los resultados de ej2\_1

-Para el funcionamiento de la práctica, las imágenes que se usan de input han de ser descargadas, renombradas con nombres del 1 al 6, ya sean PNG o JPG, y guardadas en una carpeta llamada inputs, al mismo nivel que el resto de las carpetas. Esto es así porque estas imágenes pesan demasiado (38 Mb), y el campus virtual solo deja la entrada de archivos de hasta tamaño de 50Mb, por lo que, en conjunto con otros archivos, no se podría entregar.