

## PRÁCTICA 2

### “Detección de fresas en imágenes: implementación de técnicas de segmentación basadas en color”

#### INTRODUCCIÓN Y PLANTEAMIENTO

**OBJETIVO:** Diseño de algoritmo de detección de fresas en imágenes mediante la implementación de técnicas de segmentación de imágenes basadas en color. El algoritmo recibirá como entrada imágenes del tipo,



y devolverá como salida la segmentación de las fresas rojas detectadas, ofreciendo información sobre su grado de madurez.



Deteccion de Rojo Fresa: en rojo----- Deteccion de Verde Inmaduro: en amarillo

El algoritmo de segmentación se diseñará a partir de las conclusiones alcanzadas tras la realización de las siguientes etapas:

### **ETAPA 1: OBTENCIÓN Y ANÁLISIS DE DATOS: PATRONES DE ENTRENAMIENTO**

**PASO 1.1.:** *Obtención de muestras de los colores representativos de la imagen:* rojo fresa, verde fresa, verde planta y negro lona. Se utilizarán 3 modelos de color: RGB, HSI, YUV y CIE Lab.

**PASO 1.2.:** *Representación de las muestras de color* obtenidas en los diferentes espacios de color considerados.

### **ETAPA 2: DETECCIÓN DE FRESAS EN DOS PASOS. APLICACIÓN DE TÉCNICAS DE SELECCIÓN DE CARACTERÍSTICAS Y DIFERENTES ESTRATEGIAS DE CLASIFICACIÓN**

**PASO 2.1:** *Segmentación del color rojo fresa.* Problema de clasificación binaria de píxeles: píxel rojo fresa o píxel con otro color. Se abordará de la siguiente forma:

- **Selección de descriptores:** análisis de los espacios de color desde un punto de vista de separabilidad de clases y selección de los conjuntos de descriptores matemáticos más prometedores para afrontar la etapa de clasificación.
- Aplicación de distintas técnicas de clasificación, basadas en distancia de Mahalanobis,  $k$  vecinos más próximos, red neuronal y máquinas de vector soporte, con las descripciones matemáticas previamente seleccionadas.
- Comparación y análisis de resultados para seleccionar la configuración más eficiente.

**PASO 2.2.:** *Segmentación del color verde fresa.* Para ello se debe utilizar un vector de características de 3 descriptores y un  $k$ NN como técnica de clasificación.

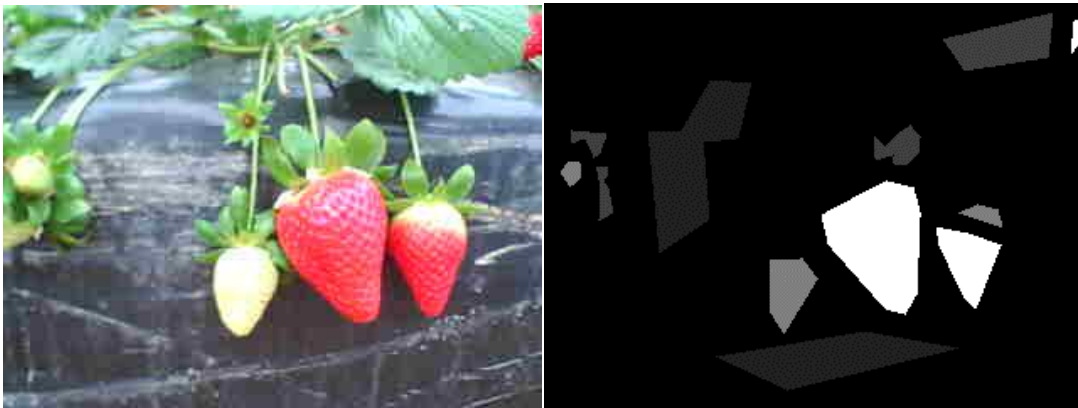
La segmentación de la fresa se realizará a partir de la detección de los colores *rojo* y *verde fresa* de los pasos anteriores

## REALIZACIÓN DE LA PRÁCTICA

### **ETAPA 1: OBTENCIÓN Y ANÁLISIS DE DATOS: PATRONES DE ENTRENAMIENTO.**

**PASO 1.1: Obtención de muestras de los colores representativos de la imagen: rojo fresa, verde fresa, verde planta, negro lona.** Se utilizarán 3 modelos de color: RGB, HSI, YUV y Lab.

**Material facilitado:** Directorio: `Material_Imagenes/01_MuestrasColores`  
(Tres imágenes originales y sus respectivas segmentaciones manuales)



Codificación utilizada para distinguir las distintas muestras de color en las imágenes de segmentaciones manuales:

- Píxeles Rojo Fresa: valor 255.
- Píxeles Verde Fresa: valor 128.
- Píxeles Verde Planta: valor 64.
- Píxeles Negro Lona: valor 32.

1.- Para cada una de las tres imágenes de color de entrada, obtener las matrices **R, G, B, H, S, I, Y, U, V, L, a, b**:

- La obtención de las componentes H y S, puede realizarse utilizando la función de matlab `rgb2hsv`.
- La obtención de las componentes I y YUV, debe implementarse a partir de las transformaciones facilitadas en el apartado 2.2 del Tema 2 de teoría.
- La obtención de las componentes L, a y b, puede realizarse utilizando la función de Matlab `rgb2lab`.

2.- Obtener las matrices anteriores (R, G, B, H, S, I, Y, U, V, L, a, b) normalizadas en el rango 0-1. Para ello, se deben tener en cuenta los rangos de variación teóricos de cada componente (ver teoría).

3.- Generar la matriz `ValoresColores`, compuesta por los valores normalizados R, G, B, H, S, I, Y, U, V, L, a, b, en los píxeles etiquetados de todas las imágenes marcadas manualmente. Además, se debe generar el vector `CodifValoresColores`, que especificará la codificación del color al que corresponden las filas de las matrices de datos anteriores.

Ejemplo de matrices `ValoresColores` y `CodifValoresColores` para dos píxeles representativos de cada color (nota: faltan por añadir las columnas correspondientes al modelo Lab):

`ValoresColores`

R	G	B	H	S	I	Y	U	V
1,000	0,373	0,380	0,998	0,627	0,584	0,561	-0,089	0,385
1,000	0,396	0,471	0,979	0,604	0,622	0,585	-0,056	0,364
1,000	1,000	0,875	0,167	0,125	0,958	0,986	-0,055	0,013
1,000	1,000	0,863	0,167	0,137	0,954	0,984	-0,060	0,014
0,396	0,675	0,376	0,322	0,442	0,482	0,557	-0,089	-0,141
0,396	0,675	0,376	0,322	0,442	0,482	0,557	-0,089	-0,141
0,275	0,314	0,349	0,579	0,213	0,312	0,306	0,021	-0,028
0,278	0,318	0,353	0,579	0,211	0,316	0,310	0,021	-0,028

`CodifValoresColores`

<b>Pixel 1</b>	255
<b>Pixel 2</b>	255
<b>Pixel 3</b>	128
<b>Pixel 4</b>	128
<b>Pixel 5</b>	64
<b>Pixel 6</b>	64
<b>Pixel 7</b>	32
<b>Pixel 8</b>	32

**PASO 1.2.: Representación de las muestras de color obtenidas en los diferentes espacios de color considerados.**

1.- Representa las siguientes gráficas:

- Gráfica RGB (utilizando `plot3`): representar los valores R, G y B de los píxeles de color rojo fresa (mostrarlos en rojo), verde fresa (mostrarlos en verde), verde planta (mostrarlos en azul) y negro lona (mostrarlos en negro).
- Gráfica HS (utilizando `plot`): representar los valores de H y S de los píxeles de color rojo fresa (mostrarlos en rojo), verde fresa (mostrarlos en verde), verde planta (mostrarlos en azul) y negro lona (mostrarlos en negro).
- Gráfica UV (utilizando `plot`): representar los valores de U y V de los píxeles de color rojo fresa (mostrarlos en rojo), verde fresa (mostrarlos en verde), verde planta (mostrarlos en azul) y negro lona (mostrarlos en negro).
- Gráfica ab (utilizando `plot`): representar los valores de las componentes cromáticas a y b del modelo Lab de los píxeles de color rojo fresa (mostrarlos en rojo), verde fresa (mostrarlos en verde), verde planta (mostrarlos en azul) y negro lona (mostrarlos en negro).

2.- Interpretar las gráficas asociando los resultados obtenidos con los valores teóricos representativos de las componentes de color en sus respectivos modelos, especialmente los que hacen referencia a los valores H. ¿Qué problema existe con esta componente de color?

3.- Recalcula los valores de H en `ValoresColores` de acuerdo a la siguiente ecuación:

$$H_{\text{recalculado}} = \begin{cases} 1 - 2 * H & \text{si } H \leq 0.5 \\ 2 * (H - 0.5) & \text{si } H > 0.5 \end{cases}$$

Vuelve a representar la gráfica HS anterior, utilizando los valores actualizados de H y observa las diferencias.

---

**Observación:** a continuación, el resto de la práctica se realizará considerando los valores de H recalculados según el punto anterior.

---

## **ETAPA 2: DETECCIÓN DE FRESAS EN DOS PASOS. APLICACIÓN DE TÉCNICAS DE SELECCIÓN DE CARACTERÍSTICAS Y DIFERENTES ESTRATEGIAS DE CLASIFICACIÓN**

**PASO 2.1: SEGMENTACIÓN DEL COLOR ROJO FRESA.** Problema de **clasificación binaria** de píxeles:

- Clase “1”: píxel tiene color rojo fresa.
- Clase “0”: píxel tiene otro color.

### **2.1.1 Selección de conjuntos de descriptores**

2.1.1.1.- Considerando nuestro problema de clasificación, **cuantificar la separabilidad** que proporcionan los espacios de color RGB, HSI, YUV y Lab mediante CSM - ("Class Scatter Matrix", estimación de las matrices de dispersión entre y dentro de las clases – se facilita función para cálculo).

2.1.1.2.- **Seleccionar los conjuntos de 3, 4, 5 y 6** descriptores que proporcionan mayor separabilidad.

2.1.1.3.- Este paso terminará con la siguiente selección de vectores característica que se utilizarán en la siguiente etapa de clasificación:

- **R G B**
- **L a b**
- **Mejor combinación de 3 descriptores.**
- **La combinación que se considere más adecuada de más descriptores** (4, 5 o 6) atendiendo a los datos de separabilidad.

### **2.1.2. Aplicación de diferentes estrategias de clasificación.**

En este apartado se van a implementar y comparar distintas técnicas de clasificación: basada en **distancia de Mahalanobis**,  $k$  vecinos más próximos, red neuronal y máquinas de vector soporte. Cada una de ellas se aplicará con las descripciones matemáticas más prometedores seleccionadas en el punto anterior. Finalmente, se analizarán y comparará el rendimiento de

cada clasificador con los diferentes conjuntos de descriptores para seleccionar la configuración más eficiente.

Consideraciones al diseño, entrenamiento y aplicación de las estrategias de clasificación consideradas:

- **Clasificación basada en distancia de Mahalanobis:**

Esta técnica se va a plantear como la obtención de una región en el espacio de color considerado que se corresponda con el color rojo fresa. De esta forma, se requiere una etapa previa de diseño para obtener la caracterización de la superficie utilizada para delimitar esta región. De forma específica, el clasificador considerará que un píxel de color desconocido es de color rojo si la distancia Mahalanobis entre su descripción de color y la del centro de la nube de puntos disponibles del color rojo es menor que un determinado valor. Se adoptarán los siguientes criterios para determinar este umbral de corte:

- **Distancia Mahalanobis máxima de la nube de puntos de la clase de interés:** esta configuración supone la detección de la totalidad de los píxeles rojo fresa disponibles.
- Distancia Mahalanobis seleccionada para excluir el 3% de los puntos rojo fresa más alejado del centroide de la nube de puntos.
- **Distancia Mahalanobis mínima de la nube de puntos de la clase fondo:** esta configuración permite no detectar como píxel rojo fresa ningún píxel que sea de otro color.
- Distancia de Mahalanobis que permite la detección como píxeles rojo fresa de únicamente el 0.05% de píxeles de otros colores (los más cercanos a la nube de puntos de interés).

**Ayuda de programación:** representación de los puntos del espacio de características considerado que se incluyen a una distancia de Mahalanobis menor que un determinado valor de umbral  $U_{\text{bral}}$  respecto a un punto dado  $c$ :

```
% Dada la matriz de covarianzas Mcov y el centro C (vector fila) de
la nube de puntos del color rojo fresa:

[X1,X2,X3] = meshgrid(0:1/40:1);

Valores = [X1(:) X2(:) X3(:)];
NumValores temp] = size(Valores);
distancia = [];

for j=1:NumValores
    X = Valores(j,:);
    distancia(j,1)=sqrt((X-Centro)*pinv(Mcov)*(X-Centro)');
end

PosicionesInteres = distancia < Umbral;

datosMahal = Valores(PosicionesInteres,:);

% REPRESENTAMOS

x = datosMahal(:,1); y = datosMahal(:,2); z = datosMahal(:,3);

plot3(x, y, z, '+g')
```

- **Clasificación basada en  $k$  vecinos más próximos ( $k$ NN), Red Neuronal (NN), Máquinas de Vector Soporte (SVM).**

La forma de aplicación del  $k$ NN, NN y SVM debe consultarse en el documento: “AplicacionClasificadores\_DescriptoresColor\_ImagenCompleta.pdf”.

### Procedimiento:

**2.1.2.1- Eliminación de valores anómalos en la clase de interés (ser píxel de color rojo fresa):** eliminar del conjunto de datos todas las muestras de píxeles rojo fresa cuya componente roja sea inferior a 0.95.

### 2.1.2.2- Diseño de clasificadores:

- **Clasificación basada en distancia de Mahalanobis:** en este caso, únicamente se deben considerar las siguientes descripciones matemáticas: RGB, Lab, y mejor combinación de 3 descriptores. En cada caso, se deben obtener los valores del centro de la nube de puntos del color rojo fresa y el umbral óptimo para cada modelo de color



de entre los posibles umbrales de distancia calculados de acuerdo a los criterios explicados anteriormente. Este umbral óptimo será el que presente mayor tasa de acierto en el conjunto de datos disponible. El diseño de este clasificador debe realizarse en un directorio independiente.

- **Entrenamiento de NN:** entrenar una NN para cada una de las descripciones matemáticas seleccionadas (ver tutorial de NN facilitado para el diseño y entrenamiento de la red). El entrenamiento de la red debe llevarse a cabo en un directorio independiente.
- **Diseño de kNN y SVM:** mediante la aplicación de la función de Matlab `fitcknn` y `fitcsvm` explicadas en el documento “AplicacionClasificadores\_DescriptoresColor\_ImagenCompleta.pdf”. Se deben contemplar las descripciones matemáticas seleccionadas. La preparación de los modelos kNN y SVM puede realizarse en el mismo *script* de aplicación de los clasificadores.

**2.1.2.3- Aplicación de clasificadores sobre las imágenes. Entrada:** fichero de imagen; **Salida:** matriz binaria de las mismas dimensiones que índice la detección de píxeles rojo fresa

**Material facilitado:** Directorio: `Material_Imagenes/02_MuestrasRojo` (Dos imágenes originales y sus respectivas segmentaciones manuales del color rojo fresa)



Codificación utilizada para distinguir la muestra de color rojo fresa en las imágenes de segmentaciones manuales:

- Píxeles Rojo Fresa: valor 255.

### Pasos:

1.- Reducir a la mitad la resolución de las imágenes:

```
imresize(ImagenEntrada, 0.5)
```

2.- Aplicar las distintas técnicas de clasificación con las diferentes descripciones matemáticas, sobre la imagen a resolución reducida.

- La aplicación del clasificador basado en distancia de Mahalanobis debe realizarse recorriendo cada píxel de la imagen.
- La aplicación del kNN, NN, y SVM deben aplicarse de forma conjunta a todos los píxeles de la imagen, tal como se indica en el documento

`AplicacionClasificadores_DescriptoresColor_Imagen Completa.pdf`

3.- Re-escalar el resultado de las detecciones  $D$  a las dimensiones originales de la imagen de entrada ( $N$  filas,  $M$  columnas): `round(imresize(D, [N M], 'nearest'))`

4.- Visualizar los resultados de las detecciones aplicando la función `VisualizaColores`. Esta función, dada la imagen en color original y la matriz resultado del tipo anterior, debe representar sobre la imagen en color las siguientes detecciones:

- En rojo: detección 255, rojo fresa.
- En amarillo: detección 128, verde fresa.
- En azul: detección 64, verde planta.
- En negro: detección 32, negro lona, y otros casos.

**2.1.2.4.- Medida del rendimiento de cada modelo** (Descripción Matemática-Clasificador) para detectar el color *rojo fresa*, en términos de sensibilidad, especificidad, falsos positivos y precisión, tomando la imagen marcada manualmente como referencia *gold*:

- Sensibilidad = Número de píxeles *rojo fresa* correctamente detectados / Número total de píxeles rojo fresa presentes en la imagen.
- Especificidad = Número de píxeles de fondo correctamente detectados / Número total de píxeles de fondo.
- Falsos Positivos = Número de píxeles *rojo fresa* erróneamente detectados / Número total de píxeles de fondo.
- Acierto = Número de píxeles correctamente detectados (*rojo fresa* + fondo) / Número total de píxeles de la imagen.

Para ello, utilizar la función facilitada:

```
[Sens Esp Acc FalsosPositivos] = funcion_metricas(Aut, Gold)
```

donde Aut y Gold son las matrices binarias generada de forma automática y manualmente, respectivamente, con 1 pixel de *rojo fresa* y 0 un pixel que no lo es.

Analizar los resultados de sensibilidad, especificidad y acierto, promedios sobre las dos imágenes disponibles, para seleccionar la técnica de clasificación y descripción matemática más eficiente.

## **ETAPA 2: DETECCIÓN DE FRESAS EN DOS PASOS. APLICACIÓN DE TÉCNICAS DE SELECCIÓN DE CARACTERÍSTICAS Y DIFERENTES ESTRATEGIAS DE CLASIFICACIÓN**

**PASO 2.2: SEGMENTACIÓN DEL COLOR VERDE FRESA.** Problema de clasificación binaria de píxeles:

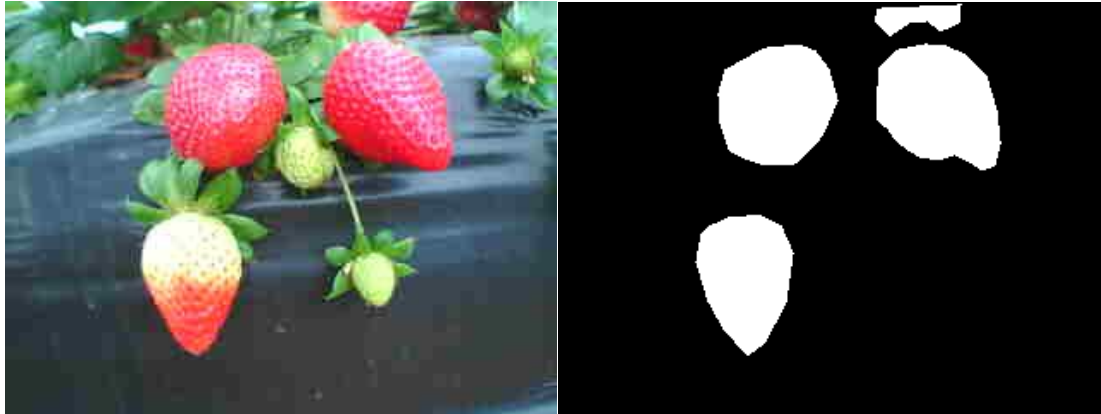
- Clase “1”: píxel tiene color verde fresa.
- Clase “0”: píxel tiene otro color.

**2.2.1 Selección de conjuntos de descriptores:** seleccionar el conjunto de 3 descriptores que proporcionan la mayor separabilidad en nuestro problema de clasificación.

**2.2.2 Clasificación:** aplicar como técnica de clasificación un kNN

**PASO 2.3.: Segmentación de fresas rojas** a partir de la detección del color *rojo* y *verde fresa*.

**Material facilitado:** Directorio: Material\_Imagenes/03\_MuestrasFresas (Tres imágenes originales y las respectivas segmentaciones manuales de las fresas rojas presentes)



Codificación utilizada para distinguir las fresas rojas en las imágenes de segmentaciones manuales:

- Píxeles Fresas Rojas: valor 255.

Para cada una de las tres imágenes disponibles, y utilizando los algoritmos de clasificación rojo fresa y verde fresa:

1.- Detectar los píxeles rojo fresa.

2.- Detectar los píxeles verde fresa.

3.- Visualizar sobre la imagen original los resultados obtenidos (en rojo, la detección del rojo fresa; en amarillo, la detección del verde fresa).

4.- Segmentar las fresas rojas presentes. Para ello se considerará que una fresa roja es el resultado de la conexión de los objetos detectados como *rojo fresa* y aquellos detectados como *verde fresa*.

5.- Visualizar los resultados de la segmentación y cuantificar la efectividad de la misma midiendo la sensibilidad, especificidad, falsos positivos y precisión, tomando la imagen marcada manualmente como referencia *gold*.

6.- Comparar estos resultados son los que se obtienen al realizar una detección en un solo paso del rojo-fresa y verde-fresa, utilizando una descripción de 3 descriptores y kNN.