

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería Electrónica de Comunicaciones



Trabajo Fin de Grado

Modelo analítico de calidad de fruto en explotaciones agrícolas dedicadas al cultivo del tomate a partir de visión artificial

ESCUELA POLITECNICA
SUPERIOR

Autor: Álvaro Sola Soria

Tutor: Juan Manuel Miguel Jiménez

2020

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

GRADO EN INGENIERÍA ELECTRÓNICA DE COMUNICACIONES

Trabajo de fin de grado

Modelo analítico de calidad de fruto en explotaciones agrícolas dedicadas al cultivo del tomate a partir de visión artificial

Autor: Álvaro Sola Soria

Tutor/es: Juan Manuel Miguel Jiménez

Tribunal:

Presidente: Cristina Losada Gutiérrez

Vocal 1º: Marta Marrón Romera

Vocal 2º: Juan Manuel Miguel Jiménez

Fecha: 02/07/2020

RESUMEN

En este proyecto se realiza el diseño de un sistema de reconocimiento de frutos (tomates en este caso) localizados en su planta. La finalidad del modelo consistirá en la realización de un conteo, calibrado y estado de madurez del fruto en la planta. Para ello se ha realizado una instalación de una cámara recogiendo imágenes diarias y poder observar su evolución.

El modelo se diseñará empleando un filtrado, por capas, de las componentes que los representan.

SUMMARY

In this project, the design of a recognition system for fruits (tomatoes in this case) located in their plant is carried out. The purpose of the model will be to carry out a count, calibration and state of maturity of the fruit in the plant. For this, a camera installation has been carried out collecting daily images and observing their evolution.

The model will be designed using a filtering, by layers, of the components that represent them.

PALABRAS CLAVE

Umbral, HSV, Segmentación, Medidor de objetos.

Índice

RESUMEN.....	3
SUMMARY	3
PALABRAS CLAVE	3
INTRODUCCIÓN	6
Presentación	6
Estado del arte.....	9
Organización de la memoria.....	11
OBJETIVOS	12
METODOLOGÍA.....	13
Principios básicos de óptica.....	13
Lente	14
Ángulo de visión.....	15
Píxel.....	18
Color	19
Comportamiento del color en espacios vegetales	20
Técnicas de segmentación.....	23
Segmentación basada en el Histograma.....	24
Segmentación basada en Bordes.....	29
Segmentación basada en Regiones	42
HARDWARE.....	44
Ubicación de las cámaras.....	45
SOFTWARE	46
Matlab.....	46
ImageJ	48
DESARROLLO DEL TRABAJO	50
Detección y conteo	50
Resultado	55
Reconocimiento de estado de madurez.....	57
Resultado	59
Calibrado del fruto.....	61
Medición de objetos	61
Resultado	68
Visualización de resultados	70
EXPERIMENTOS Y RESULTADOS	71
Ejecución del modelo definitivo	71
Detecciones erróneas	74

Vista global de resultados	76
Mejora con recorte de la imagen	79
Ensayos no llevados al proyecto.....	81
Detector de bordes	81
RCNN	85
INTERFAZ DE USUARIO	88
DISCUSIÓN Y CONCLUSIONES.....	91
Conclusiones.....	91
LINEAS FUTURAS.....	92
PLANIFICACIÓN TEMPORAL Y PRESUPUESTO	93
Planificación temporal.....	93
Presupuesto.....	93
ANEXOS.....	95
Código de aplicación para interfaz de usuario MATLAB	95
Algoritmo MATLAB	106
Tabla de bandas salariales.....	111
BIBLIOGRAFÍA	112

INTRODUCCIÓN

Presentación

En el sector agrícola se tienen que tomar muchas decisiones que afectan al rendimiento y calidad del producto sembrado. Actualmente, muchas de ellas se toman en base a observaciones humanas dependientes de la vista y experiencia de la persona que las recoge. Estos procedimientos hacen que todas ellas sean lentas y costosas y, además, requiere de largos procesos de formación.

Una de las decisiones más estratégicas que tiene que tomar un agricultor es el momento de cosecha. Antes de proceder, hay que tener en cuenta ciertos criterios de madurez y calibre que definirán finalmente la calidad del producto [1]. Ahí entra en juego el obtener la mayor rentabilidad, minimizando los costes de explotación que se tienen, empleando los métodos tradicionales.

La tecnología ofrece técnicas de visión, a través de imágenes sometidas a procesos computacionales, capaces de detectar, identificar y analizar objetos específicos que se encuentran en su campo de visión. De esta forma, el agricultor puede disponer de información muy precisa de la cantidad y calidad de recolección antes de realizarlo y de forma remota. Además, ayudará a determinar el momento idóneo para planificar la cosecha.

La visión artificial, al igual que la inteligencia artificial, no es nueva en las operaciones artificiales. De hecho, existen numerosos sistemas desarrollados para la industria manufacturera con unos beneficios notables.

Los avances que han ido dando en los últimos años las tecnologías relacionadas con la visión artificial la han hecho mejorar considerablemente en su calidad y precisión de obtener mejores resultados, haciéndolo más factible su introducción por parte de los agricultores [2].

Existen casos de éxito en la aplicación de esta tecnología en el campo como los que está realizando la empresa HISPATEC con empresas como CAMPOSOL (Chile) y su aplicación CAMPOGEST. Esta herramienta permite el reconocimiento facial de los trabajadores que se encuentran en el lugar de trabajo [3].



Figura 1: Usuario realizando el procedimiento de reconocimiento facial al personal de campo.

Además del caso mostrado, existen diversos modelos de clasificación de frutos en cadenas de manufactura y empaquetado. La mayoría de ellos se basan en técnicas de clasificación del fruto según características (color, calibre, ...) pero aplicando instrumentos de contacto e incluso humanos con las ratios de error tan elevados que aportan.



Figura 2: Planta industrial para la selección y manufactura de fruta.

Las tecnologías de visión se están introduciendo desde hace ya tiempo, pero aún entra muy tímidamente, de forma que los proyectos que emplean esa tecnología están orientados con fines, mayoritariamente, experimentales.

Existen técnicas de reconocimiento de frutos, aunque se realizan en condiciones ideales sin un entorno que introduzca elementos de ruido.



Figura 3: Método de clasificación empleado en proyecto de sistema de reconocimiento de frutas [4].

También existen numerosos experimentos tecnológicos que están diseñando robots autómatas que extraen los frutos y, como se puede esperar, emplean esta tecnología para su localización y recolección con resultados excelentes.



Figura 4: Brazo robótico recolector de fruta.

Estado del arte

La visión artificial, o visión por computador, es la disciplina que estudia cómo reconstruir, interpretar y entender una escena en 3 dimensiones a partir de imágenes en 2 dimensiones. El objetivo es dotar al sistema de capacidad para poder identificar y clasificar objetos determinados a partir de un modelo que exceda las limitaciones de visión humana a partir de un hardware y un software determinado.

Mientras que la visión humana es mejor para la interpretación cualitativa de una forma desestructurada compleja, la visión artificial destaca en la medida cuantitativa de una escena estructurada debido a su velocidad, precisión y repetibilidad. Un sistema de visión artificial puede inspeccionar cientos, o incluso miles, de piezas por minuto.

Esta tecnología, además, permite eliminar el contacto físico entre el sistema de prueba y las piezas que se van a verificar, evitando daños en las mismas, tiempo y mantenimiento de los elementos mecánicos necesarios. También aporta beneficios operativos y de seguridad al reducir la participación humana evitando la contaminación en salas limpias y protegiéndolas frente a entornos peligrosos.

Para llevar a cabo el proceso de crear un proceso computacional de visión artificial, se deben seguir los siguientes pasos fundamentales:

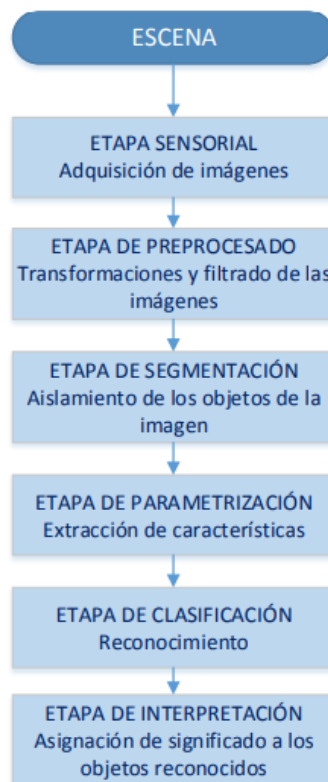


Figura 5: Pasos fundamentales para crear un proceso de visión artificial.

En la primera etapa, se recoge la imagen digitalizada. Para esto se puede emplear una cámara fotográfica normal, aunque también se podría emplear una cámara térmica, infrarroja, ...

En la etapa de preprocesado se adapta la imagen con el fin de facilitar su posterior tratamiento.

En la siguiente etapa se realiza la segmentación de la imagen. Este proceso trata de obtener una imagen reducida, formada por objetos más sencillos, contornos de estos o puntos interiores. El sistema devolverá una serie de imágenes que descompongan la imagen en objetos con diferentes características. Esto aportará una información extra a la siguiente etapa para realizar un tratamiento de la imagen más eficiente ya que las imágenes son muy complejas de tratar.

En la siguiente etapa de parametrización se trata de extraer las características que permiten distinguir unos objetos de otros.

En la etapa de clasificación se trata de agrupar todos los objetos con las mismas características, separándolas de los que sean diferentes.

Finalmente, en la etapa de interpretación, se determina el significado de la imagen analizada.

Estas etapas son las que, básicamente, debe seguir cualquier sistema de visión artificial, sea cual sea su propósito.

Organización de la memoria

- Introducción

En este apartado se hace una presentación general del proyecto presentado y una introducción sobre los campos que se van a tratar. También se realiza una introducción al estado actual del arte donde se explica la teoría básica de la visión artificial y los procesos comunes de todos los sistemas que se basen en esta tecnología.

- Objetivos

Presentación de los objetivos principales a los que se quiere llegar en este proyecto.

- Metodología

Exposición de la teoría a partir de la cual se fundamenta todo lo desarrollado en este proyecto.

- Hardware

Características de todos los elementos físicos utilizados en el proyecto. Especialmente la estación instalada para la captura de las imágenes.

- Software

Se describen las herramientas que se han empleado para el análisis y el tratamiento de imágenes al igual que el desarrollo del modelo empleado para la detección de los frutos.

- Desarrollo del Trabajo

Definición del trabajo realizado para el desarrollo del modelo de detección. Se va a dividir en varios subapartados: Detección y conteo, reconocimiento del estado de madurez y calibrado del fruto.

- Experimentos y resultados

En este apartado se muestran las pruebas realizadas paralelamente al desarrollo. Estas han sido útiles para determinar el camino a seguir y los pasos futuros para mejorar el modelo.

- Discusión y conclusiones

Se discuten los resultados obtenidos y se destacan las conclusiones alcanzadas,

- Líneas Futuras

Análisis de posibles vías de investigación y mejoras del trabajo realizado.

- Planificación temporal y presupuesto

Detalle de las actividades realizadas durante el proyecto y presupuesto de este.

- Anexos
- Bibliografía

OBJETIVOS

Los objetivos principales de este proyecto consisten en la obtención de parámetros en plantas ornamentales (tomateras en este caso) tales como el conteo de frutos, parámetros y el estado de estos.

Para ello, se ha dispuesto de imágenes recibidas desde un invernadero localizado en Almería donde realizan el cultivo de este fruto en la campaña de invierno (octubre-abril) de la variedad *Sugarino* en rama.

Esto puede aportar información, en tiempo real, del estado del cultivo y disponer de un histórico de evolución de este para posteriores análisis que aporten valor a modelos de predicción para cosecha, calidad y muchos otros.

Se pretende demostrar la capacidad que puede tener un sistema de visión de reconocer frutos (tomates en este caso) dentro de un ambiente con numerosos elementos que interfieren en el reconocimiento de estos como sucede con la vegetación del entorno y el suelo.

METODOLOGÍA

Principios básicos de óptica

La motivación para realizar un apartado para explicar los principios de óptica es porque, en este proyecto, se trabaja con un elemento captador de imágenes (cámara) que está dotada de, al menos, una lente. Por ello, es importante conocer sus principios con el fin de poder entender y aportar cierto valor añadido al producto final.

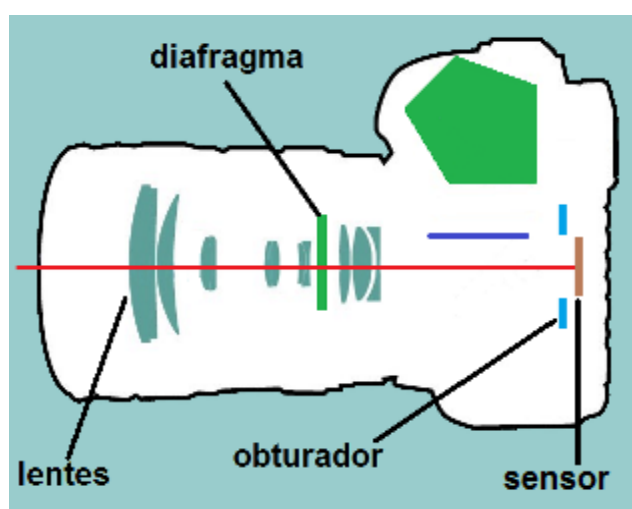


Figura 6: Arquitectura básica de una cámara fotográfica.

Como se puede observar en la imagen, una cámara está compuesta, principalmente, por una, o varias, lentes, un diafragma, un obturador y un sensor (entre otros elementos). En este caso, lo que más nos interesa conocer es el principio de funcionamiento de la lente y el sensor.

Lente

La lente es un objeto transparente (normalmente de vidrio), limitada por dos superficies, de la que, al menos, una es curva. La función de esta consiste en proyectar la luz del exterior formando una imagen en el sensor.

Una de las características principales de la misma es su distancia focal (f). Esta es la distancia entre el centro óptico de la lente y el punto focal cuando enfocamos al infinito.

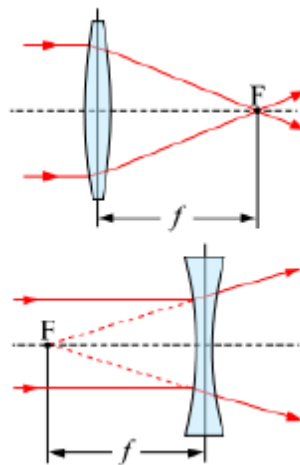


Figura 7: Representación del punto focal en lente convergente (arriba) y divergente (abajo).

Existen dos tipos de lentes, la lente convergente (arriba) y divergente (abajo). En las cámaras fotográficas, en caso de tener solo una lente (como es el caso actual), sería convergente. De forma que la distancia focal siempre será positiva.

Una relación básica y muy importante entre la imagen y la distancia focal es la que se muestra en la ecuación de lentes delgadas:

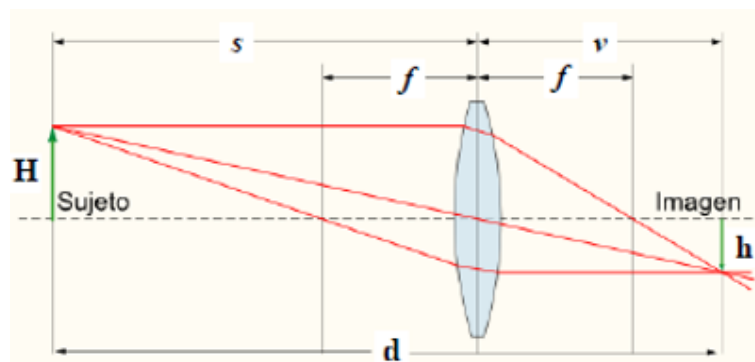


Figura 8: Relación de la distancia focal y el tamaño del sujeto-imagen.

La relación entre la distancia focal respecto a la imagen se puede representar a partir de la ecuación fundamental de las lentes delgadas:

$$\frac{1}{s} + \frac{1}{v} = \frac{1}{f} \quad (ec. 1)$$

$$m = \frac{h}{H} = \frac{v}{s} \quad (ec. 2)$$

Esta ecuación nos muestra la relación existente entre la distancia del objeto capturado y la imagen reflejada en el sensor. De forma que se puede obtener de una manera sencilla el tamaño de la imagen en el sensor respecto del tamaño real.

Esto visto anteriormente, aplicando alguna relación trigonométrica básica, nos va a permitir obtener el ángulo de visión.

Ángulo de visión

La información de métrica en una imagen es importante para poder disponer de la información de tamaño y calibrado de los objetos que componen la imagen. La técnica que facilita la obtención de este tipo de información empieza por el conocimiento del ángulo de visión en función de la distancia focal y el tamaño del sensor (datos que disponemos actualmente).

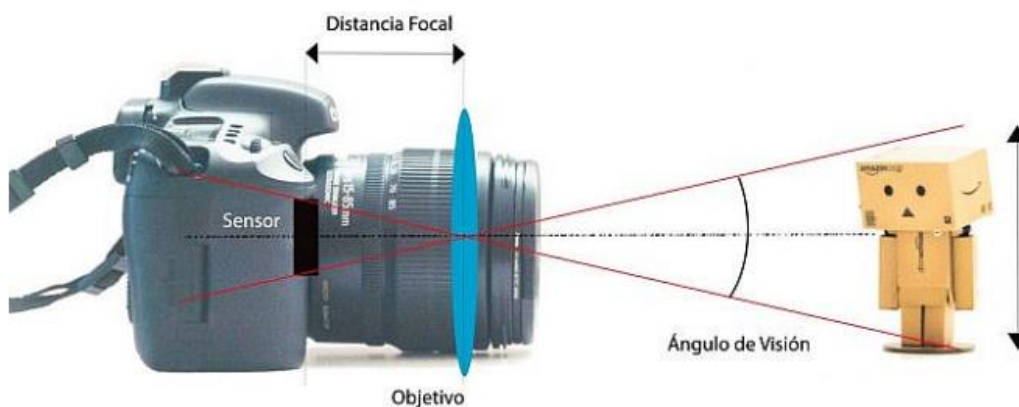


Figura 9: Visualización del ángulo de visión aplicado en una cámara fotográfica.

Como se ha descrito anteriormente, el ángulo de visión es función de la distancia focal y el tamaño del sensor. Viene dada por la siguiente ecuación:

$$\alpha = 2 \arctan \frac{2h}{2f(m+1)} \quad (ec. 3)$$

Relación del ángulo de visión con el tamaño del sensor [5]

Aunque, en este caso, al tratarse de distancia con enfoque grande ($D \gg d$; $m \approx 0$), podemos aproximarlo con la siguiente ecuación:

$$\alpha = 2 \arctan \frac{2h}{2f} \quad (ec. 4)$$

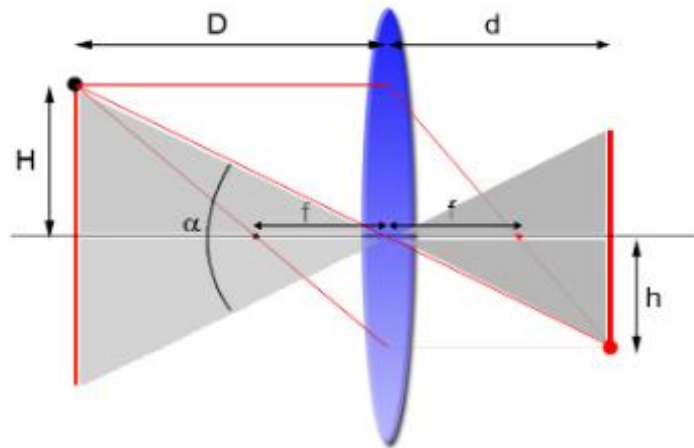


Figura 10: Representación del ángulo de visión.

De forma que, para obtener la longitud vertical de la imagen ($2H$) a partir de la distancia (D), podemos aplicar la siguiente relación:

$$\tan \frac{\alpha}{2} = \frac{2h}{2f} = \frac{H}{D} \quad (ec. 5)$$

Como se puede observar, a medida que disminuye la distancia focal (f), el ángulo de visión aumenta, casi, exponencialmente.

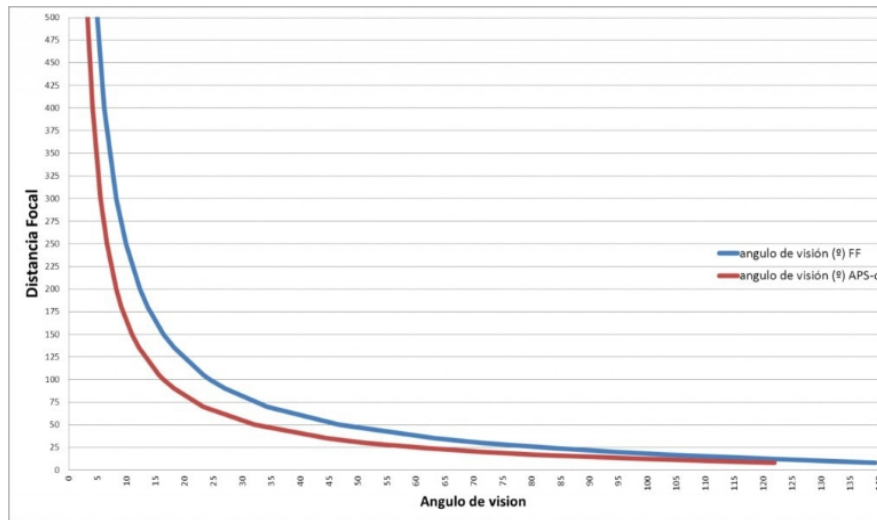
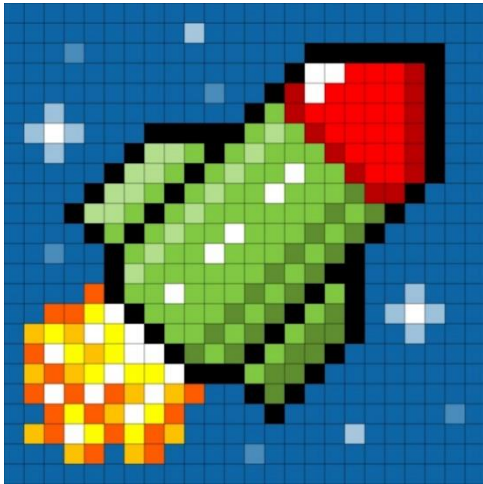


Figura 11: Relación de la distancia focal con el ángulo de visión.

Gracias a esta relación obtenida, se va a poder medir el tamaño de la imagen reflejada en el sensor. Este sensor está compuesto por un gran número de captadores, denominados píxeles, y están distribuidos homogéneamente por el sensor. Estos nos van a permitir fragmentar el sensor en trozos lo suficientemente pequeños para poder tener precisiones inferiores al milímetro. De todos modos, en el siguiente capítulo hablaremos más de ellos.

Píxel

Las imágenes visualizadas a partir de fotografías digitalizadas, está compuestas de una cantidad de puntos que contienen la información del color en un área determinada. La localización de un cierto número de ellas en una matriz, generan la imagen. A medida que aumenta el número de píxeles, se aprecia menos su presencia observando las variaciones del color de una forma más suave y progresiva.



a



b

Figura 12: Imagen representada en un plano de 576 píxeles (a). Imagen representada en un plano con 1000000 de píxeles (b).

Las cámaras fotográficas digitales, disponen de una entrada de luz que, después de atravesar un conjunto de lentes, la concentran en un sensor de un área determinada. Esa área está compuesta por un número determinado de sensores fotosensibles que captan la luz y la almacenan asignándoles un valor según el nivel de intensidad incidente en el mismo. Como resultado final, el conjunto de sensores genera una matriz bidimensional de niveles de gris.

Además de recoger la imagen captada, nos van a permitir realizar mediciones en la imagen a partir de la distancia a la que se encuentra el objeto. Pero esto será desarrollado más adelante.

Color

El color es la impresión producida por un tono de luz en los órganos visuales. [6]

Depende, en parte, de la longitud, o longitudes, de onda de las ondas luminosas incidentes. A partir de ciertos límites, inferiores y superiores, deja de producir esa impresión visual.

En la naturaleza se producen dos tipos de combinaciones de colores: pigmento y luz. El primer tipo se realiza con colores de síntesis sustractiva. Como su nombre dice, los pigmentos, al mezclarse, sustraen o absorben más luz formando tonos más oscuros. De forma que reflejan las ondas de unas longitudes de onda determinadas y absorben el resto. La combinación de todos los colores pigmento básicos (cian, magenta, amarillo) absorbe todas las ondas de espectro visible, obteniendo el color negro. En el segundo tipo de combinación, se realiza con colores de síntesis aditiva. De forma que la combinación de todos los colores básicos (verde, rojo y azul) suman luz y se obtiene el color blanco.

Como se ha mencionado anteriormente, el espectro de color se puede obtener a partir de tres componentes base. Este está constituido por tres planos bidimensionales que almacenan el nivel de intensidad de cada color. El color de cada píxel es el resultado de la combinación lineal de los valores almacenados en cada uno de los canales.

Dependiendo de la base empleada para obtener la gama de colores mediante combinaciones lineales, se disponen de diferentes representaciones del color, denominados espacios de color. Estos espacios, son siempre tridimensionales y se pueden enumerar los siguientes:

- RGB (Red/Green/Blue): Empleado en imágenes digitales.
- CYM (Cyan/Magenta/Yellow): Síntesis sustractiva.
- YUV y YIQ (Intensity/Cromacity): Transmisión de señales de televisión.
- HSV (Hue/Saturation/Value): Imágenes artísticas.

El espacio de color más empleado es el RGB. Este está basado en los tres tipos de receptores que tiene el ojo humano (verde, rojo y azul). En cada plano de color, se representa uno de estos colores primarios. Se puede modelar este espacio a partir de un cubo cuyos ejes son los colores primarios y cada esquina es un color obtenido a partir de la combinación de los colores primarios.

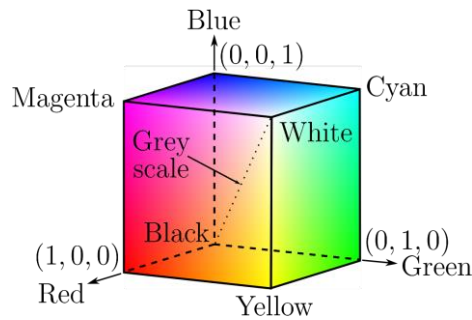


Figura 13: Cubo modelo del espacio de color RGB

En el origen del cubo, se representa el color obtenido por la ausencia de color (negro) y, en la esquina opuesta, se obtiene el color representado por la máxima intensidad de cada color (blanco).

Otro espacio de color definido es el espacio HSV. Este se considera ser el más cercano a la percepción humana y los tres parámetros básicos son el matiz (Hue), la saturación (Saturation) y el brillo (Value). El matiz representa la longitud de onda dominante, la saturación hace referencia a la pureza (ausencia de blanco) y el brillo representa la intensidad del color.

Comportamiento del color en espacios vegetales

El análisis ideal sería tratar de identificar un objeto localizado en un entorno con tonalidades de color estable y homogéneo. Existen numerosos trabajos de identificación de fruta rodeado de un entorno de color blanco.

En este caso, al tratarse de identificar objetos en un entorno variable y muy diverso, se va a tener que realizar un proceso de filtrado muy selectivo.

El plano por analizar está compuesto por vegetación, suelo arenoso, posibles fondos de las paredes del invernadero y el fruto a detectar.

En primer lugar, se procede a analizar los distintos planos de color.

En los planos RGB, se observa cómo los frutos disminuyen su nivel de verde a medida que va madurando. De esta forma, cuando llega al nivel de maduración máximo, se localiza con un nivel mínimo (muy cerca del 0). Por otra parte, se puede observar cómo el plano azul muestra niveles inferiores en los frutos y superiores en el resto.



Plano RGB completo



Plano ampliado



Plano Rojo

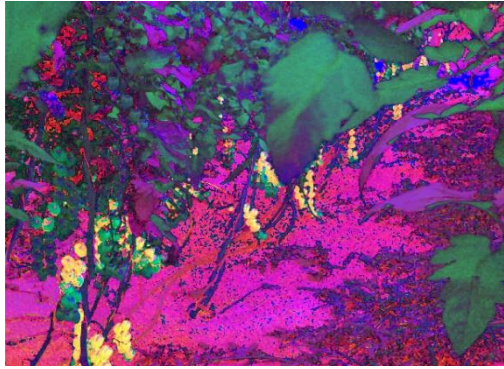


Plano Verde

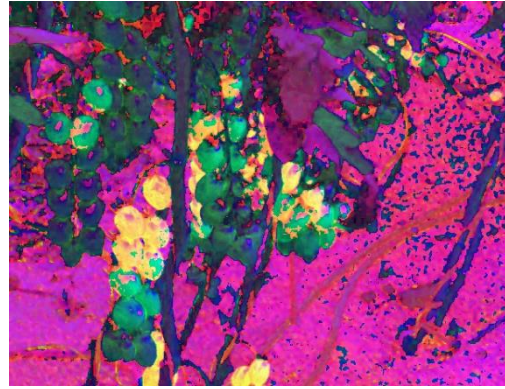


Plano Azul

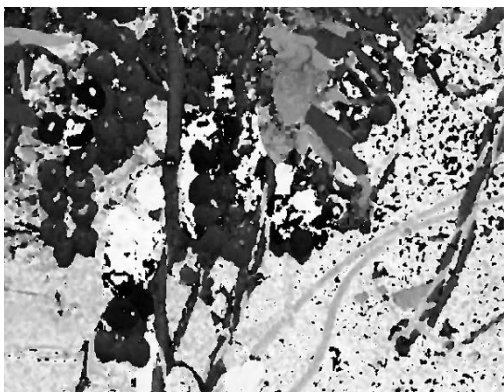
Analizando los planos HSV, se puede observar cómo los frutos maduros llegan a niveles de Matiz y saturación situados muy en los extremos (superior en el matiz e inferior en la saturación). También se observa cómo, prácticamente, todos los frutos disponen de un área reducida y distanciada del resto de frutos, en el que tiene un valor de brillo elevado.



Plano HSV completo



Plano HSV ampliado



Plano Matiz



Plano Saturación



Plano Brillo

Después de analizar estas características, se puede determinar que el plano que permite distinguir entre los frutos y el resto de los elementos del entorno se encuentra en el espacio HSV.

Técnicas de segmentación

Segmentar una imagen consiste en dividirla en regiones de interés que permitan localizar partes u objetos que la componen. Para ello, se trata de buscar características homogéneas dentro de ellas que contrasten con las de alrededor. A todas estas regiones se las denominarán, objetos de la imagen.

Para poder realizar esta función, se debe proceder mediante tres fases: Preprocesado, Extracción de características y Forma de segmentar.

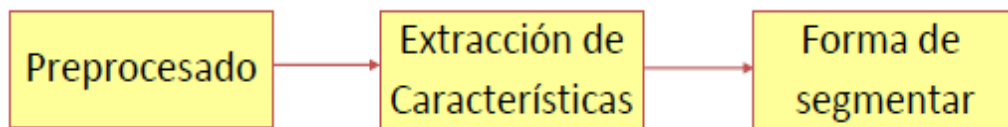


Figura 14: Representación de las fases para el proceso de segmentación de una imagen.

En la fase de preprocesado se manipula la imagen con el fin de eliminar todas las partes que no interesan. Para ello se aplican recortes, ampliación-disminución y rotaciones de esta.

En la fase de extracción de características, se destacan todas aquellas que definen con mayor claridad los objetos que se trata de detectar en la misma.

Por último, en la fase de segmentación se aplican las técnicas necesarias para extraer todos esos objetos de interés.

Dentro de las innumerables técnicas posibles para segmentar una imagen, las más empleadas consisten en:

Segmentación basada en el Histograma

Existen ocasiones en las que los distintos elementos de interés de la imagen son identificables en el histograma. La complejidad del algoritmo dependerá del grado de dispersión de los histogramas de los distintos objetos.

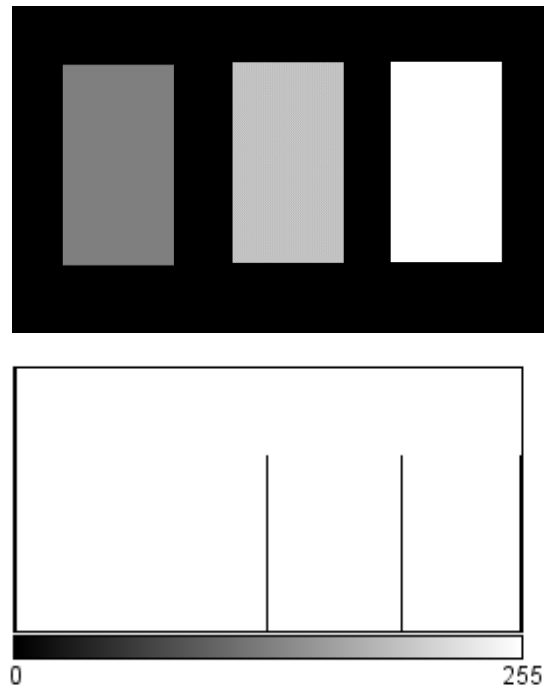


Figura 15: Imagen original en escala de grises y centrada en 4 tonalidades (arriba). Histograma de la imagen donde se aprecia la gran dominancia del negro con valor mínimo, los valores intermedios y el blanco con valor máximo (abajo).

En el histograma mostrado, la gran separación de los lóbulos que representan cada objeto nos permitirá hacer fácilmente una segmentación de estos.

En esta técnica, cada lóbulo corresponde a la distribución en probabilidad de cada clase de objeto en la imagen.

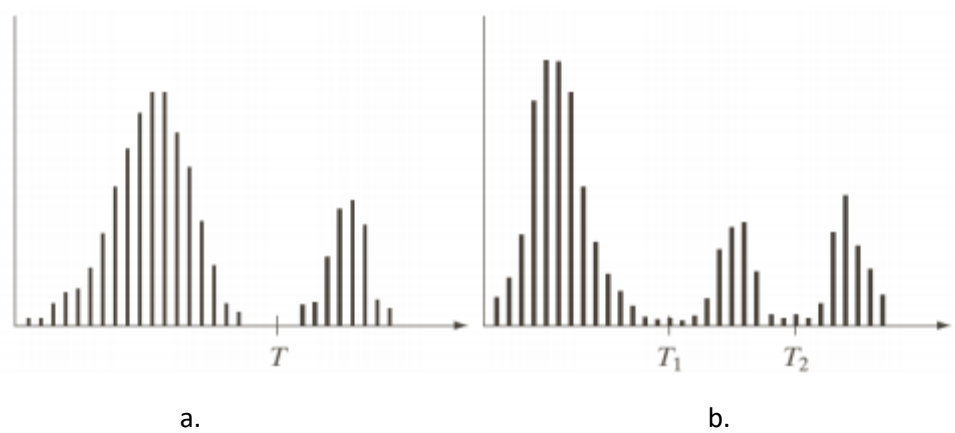


Figura 16: Representación de histograma con umbral simple (a) y umbral múltiple (b).

En la figura podemos observar un caso posible de una imagen con dos objetos. Existen dos lóbulos en diferentes niveles de gris de la imagen. Esto indica la existencia de dos regiones diferentes que pueden pertenecer a dos objetos.

Para separar los lóbulos visualizados en dos regiones distintas, se fija un valor como umbral que separe un nivel de otro.

$$g(u, v) = \begin{cases} 255 & \text{si } f(u, v) > T \\ 0 & \text{si } f(u, v) \leq T \end{cases} \quad (\text{ec. 6})$$

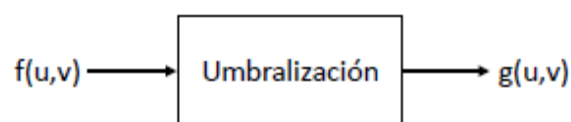


Figura 17: Caracterización de la salida del sistema $g(u, v)$ después de filtrar los píxeles de una imagen $f(u, v)$ por un selector umbralizado.

Esta técnica se puede aplicar de distintas formas:

- Umbralización global

En este método se aplica un umbral para toda la imagen. Permite destacar todos los objetos situados dentro del área umbralizada.



Figura 18: Representación de funciones de umbralización.

Esto se aplica en imágenes con características muy definidas. En casos en los que no sea así, se aplican técnicas como las que posteriormente se van a citar.

- Umbralización Local

Existen ocasiones en que los valores de intensidad de la imagen varían en función de la iluminación de la escena. Por este motivo, es muy difícil definir un umbral que sea global para toda la imagen.



Imagen 4: Vista de imagen con distintos niveles de brillo en el plano.

En este caso, se puede emplear una técnica de umbralización adaptativo y aplicarlo de forma local y diferente para cada entorno. Esa técnica consiste en la segmentación del total de la imagen en elementos reducidos en los que la intensidad varíe lo menos posible. La forma de umbralizar cada elemento sería aplicando métodos estadísticos

(media, moda, mediana, ...) que permitan distinguir los elementos de interés en cada uno de ellos.



Imagen 5: Original (a), con umbralización global (b) y con umbralización local (c).

- Umbralización multispectral

Esta técnica se aplica para situaciones en las que las imágenes dispongan de más de un plano y necesite de todos ellos para segmentar satisfactoriamente los objetos localizados en ella.

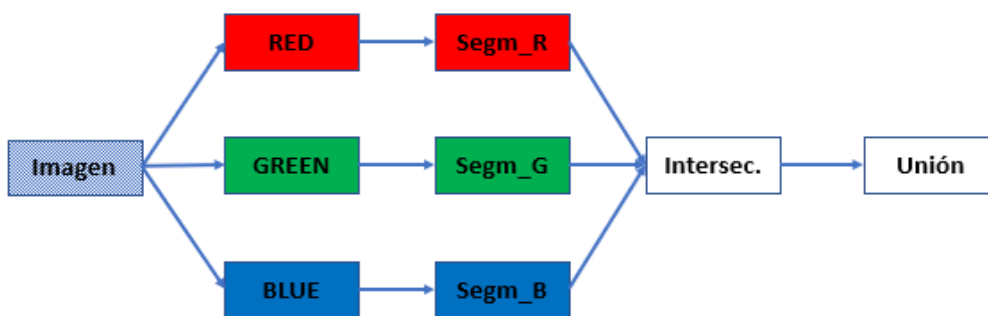


Figura 19: Esquema de proceso para filtrado de la imagen con filtros RGB umbralizados.

En este caso, se debe localizar un umbral en cada plano y, mediante unión o intersección, poder extraer la segmentación en color.

En la imagen, se muestra una segmentación mediante los planos RGB, pero también se puede emplear mediante los planos de otras componentes (como HSV). Se puede aplicar una, otra o todas.

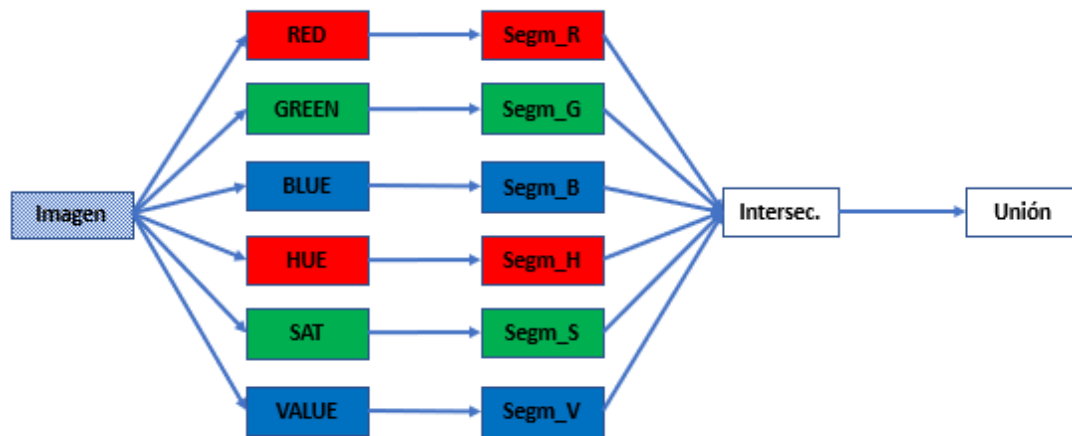


Figura 20: Esquema de proceso para filtrado de la imagen con filtros RGB & HSV umbralizados.

Segmentación basada en Bordes

Los bordes en la imagen son los cambios de intensidad significativos que hay en ella. Esto cambios pueden ser producidos por discontinuidades en la superficie, en la profundidad, en el color o textura, en la iluminación, sombras, ...

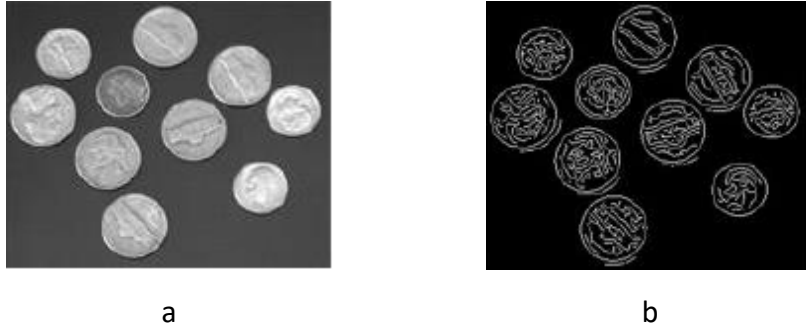


Imagen 6: Imagen original (a) a imagen después de detector de bordes (b).

En las ocasiones que los objetos estén situados en un entorno con textura homogénea, la posibilidad de detectarlos con esta técnica es ideal. De esta forma, se reconoce la región de los objetos de forma muy precisa y con un tiempo de computación muy bajo.

Por otra parte, al tratarse de objetos en un entorno menos homogéneo, requiere la aplicación de muchas otras técnicas antes y después del filtrado. Incluso, en ocasiones, deja de ser la mejor técnica para aplicar en el modelo.

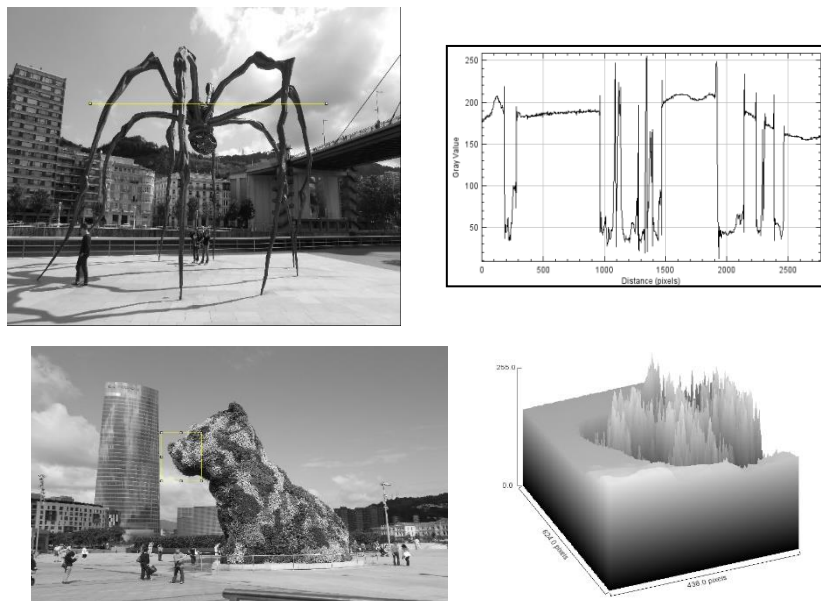


Imagen 7: Vista de niveles 2D (arriba) y 3D (abajo) en imágenes.

En estas imágenes se puede observar la gran cantidad de discontinuidades que se pueden encontrar y la dificultad existente a la hora de definir un umbral que identifique

el borde. También, en las imágenes, se suele encontrar niveles de ruido muy elevado imposibilitando su detección.

Para poder detectar estos cambios de intensidad en el plano analizado de la imagen, se emplean máscaras predeterminadas (según el método aplicado) aplicando un proceso de convolución sobre la imagen.

El proceso de convolución consiste en la suma al valor de intensidad del píxel analizado el resto de los píxeles vecinos, aplicando cierto coeficiente definido por la máscara. La razón que define este proceso sería el siguiente [7]:

$$g(x, y) = h * f(x, y) = \sum_{dx=-a}^a \sum_{dy=-b}^b h(dx, dy) f(x + dx, y + dy) \quad (ec.7)$$

Donde $g(x, y)$ se trata de la imagen filtrada, $f(x, y)$ la imagen original y h el filtro aplicado.

Un filtro identidad (no realiza ninguna modificación en la imagen) sería del tipo:

$$h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

De esta forma todos los vecinos tienen una influencia con factor 0 sobre el píxel analizado.

Las técnicas de detección de bordes conocidas son las basadas en la primera derivada o en la segunda.

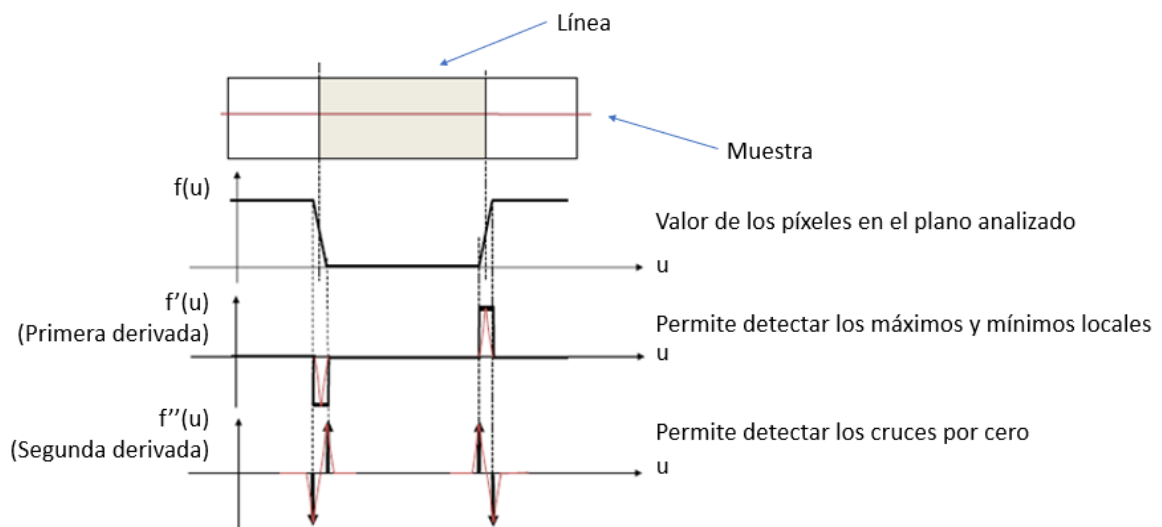


Figura 21: Representación gráfica durante el proceso de detección de un borde.

Para poder aplicar estas técnicas, antes se debe realizar un proceso de preparación con el fin de disponer de una imagen óptima para la detección de los bordes. Por ello se debe eliminar, en la mayor medida posible, el ruido localizado en la imagen.

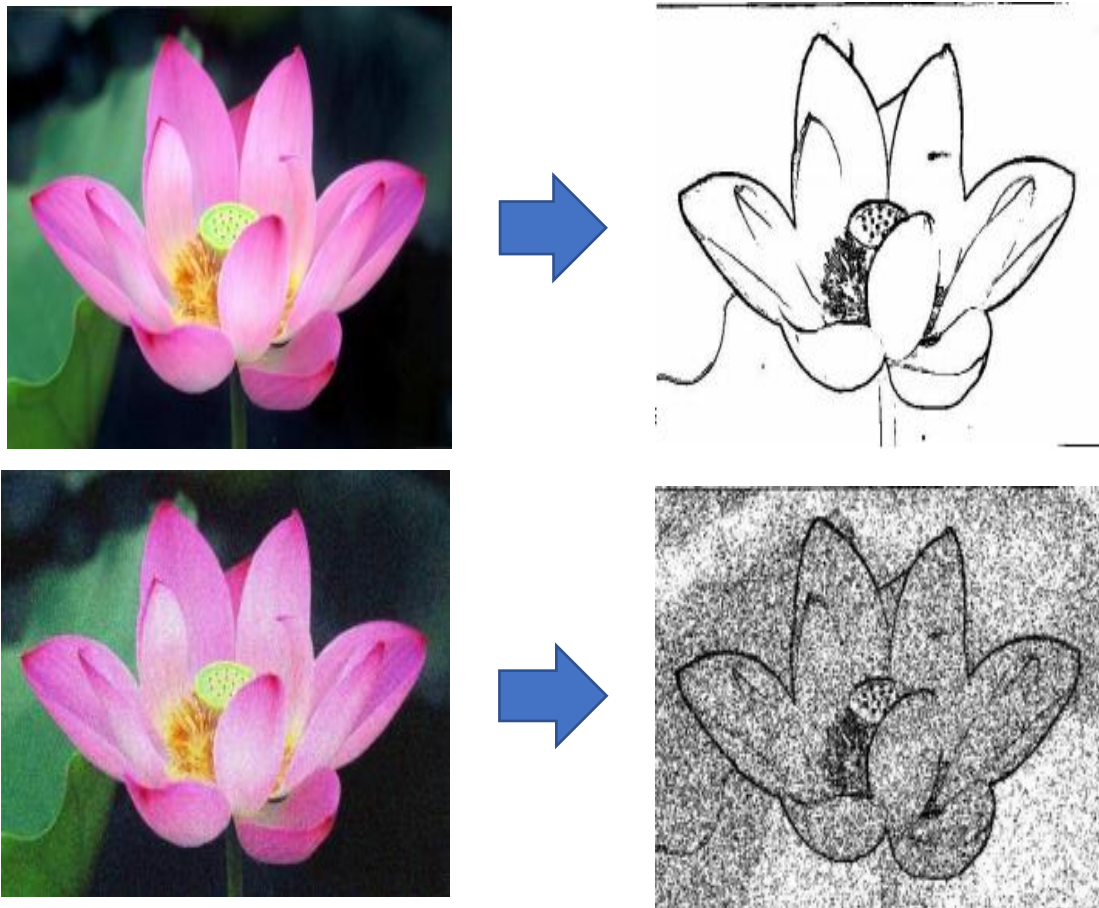


Figura 22: Visualización de resultado obtenido por detector de bordes en imagen limpia de ruido (arriba) y con ruido (abajo).

A medida que disminuya el ruido en el objeto y el contorno, será óptima la detección del borde que los separa.

Finalmente, una vez que se distingue con claridad los bordes posibles, se deben realzar y analizar la posibilidad de eliminar todos aquellos que representen ruido y mantener el resto. Para ello será necesaria la definición de un umbral.

Entre las técnicas de detección mencionadas (primera y segunda derivada) podemos mencionar los siguientes métodos que las aplican:

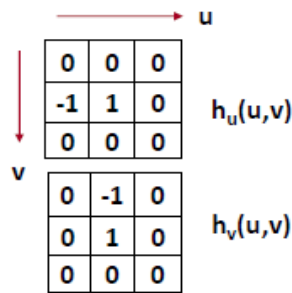
- Primera derivada

- Gradiente

Este método detecta los cambios que se producen en la dirección del gradiente seleccionado.

$$G[f(u, v)] = \begin{bmatrix} \frac{\partial}{\partial u} f(u, v) \\ \frac{\partial}{\partial v} f(u, v) \end{bmatrix} \begin{cases} |G| = \sqrt{G_u^2 + G_v^2}, & |G| = |G_u| + |G_v| \\ \theta(u, v) = \tan^{-1} \frac{G_v}{G_u} \end{cases} \quad (ec. 8)$$

$$g(u, v) = \begin{cases} 255 & \text{si } |G[f(u, v)]| > T \\ 0 & \text{si } |G[f(u, v)]| < T \end{cases} \quad (ec. 9)$$



Las máscaras empleadas, se aplican en toda la imagen y, a partir del umbral T definido, pone al nivel máximo todos aquellos píxeles que tienen una variación por encima y al mínimo los que está por debajo.

Se pueden definir tantos gradientes como se consideren. Finalmente, se procede a unir todos los resultados obteniendo algo como lo que se muestra a continuación:

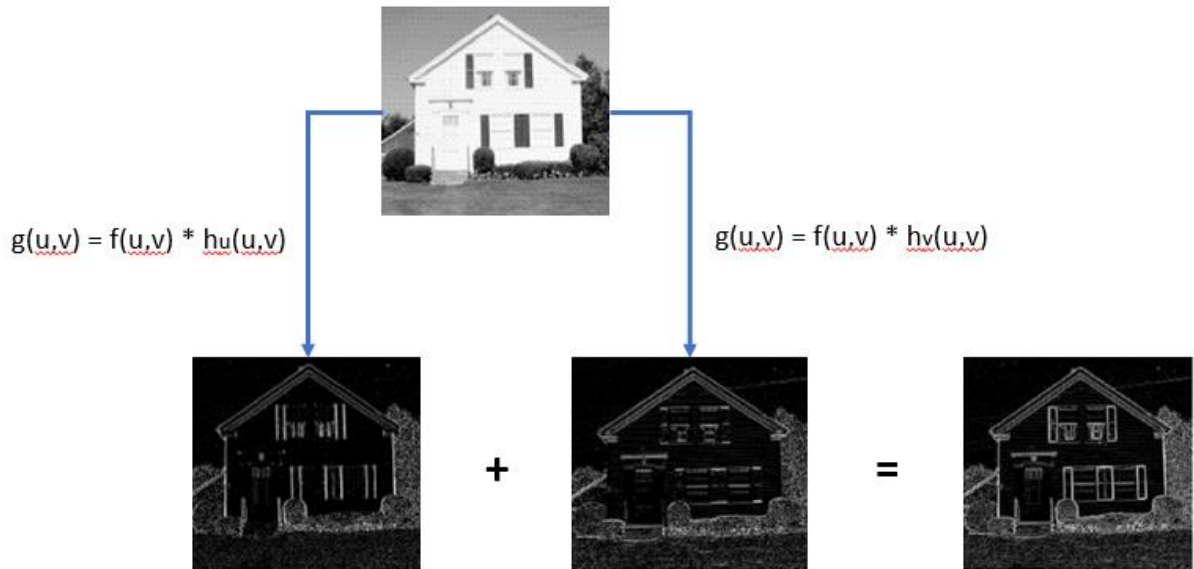


Figura 23: Representación de la unión de imágenes sometidas a dos máscaras diferentes: Gradiente vertical (h_v) y gradiente horizontal (h_u).

- Orientación

En este método, además de detectar el borde, también aporta información del ángulo de orientación que tiene el gradiente.

Un ejemplo de filtros empleados sería los siguientes:

$$\begin{array}{cccc}
 k_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & k_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & k_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & k_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
 [0^\circ] & [45^\circ] & [90^\circ] & [135^\circ] \\
 k_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & k_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & k_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & k_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\
 [180^\circ] & [225^\circ] & [270^\circ] & [315^\circ]
 \end{array}$$

$$\begin{aligned}
 |G[f(u, v)]| &= \max\{|k_0 * f(u, v)|, |k_1 * f(u, v)|, \dots, |k_7 * f(u, v)|\} \\
 &= \max\{|k_i * f(u, v)|\} \text{ (ec. 10)}
 \end{aligned}$$

$$\theta = \text{ángulo de la dirección correspondiente a } \max\{|k_i * f(u, v)|\}$$

El método, detecta qué filtro es el que presenta mayor valor en el resultado. Esto define el grado de existencia de un borde y el ángulo que tiene.

Un ejemplo del resultado que se puede obtener sería el siguiente:

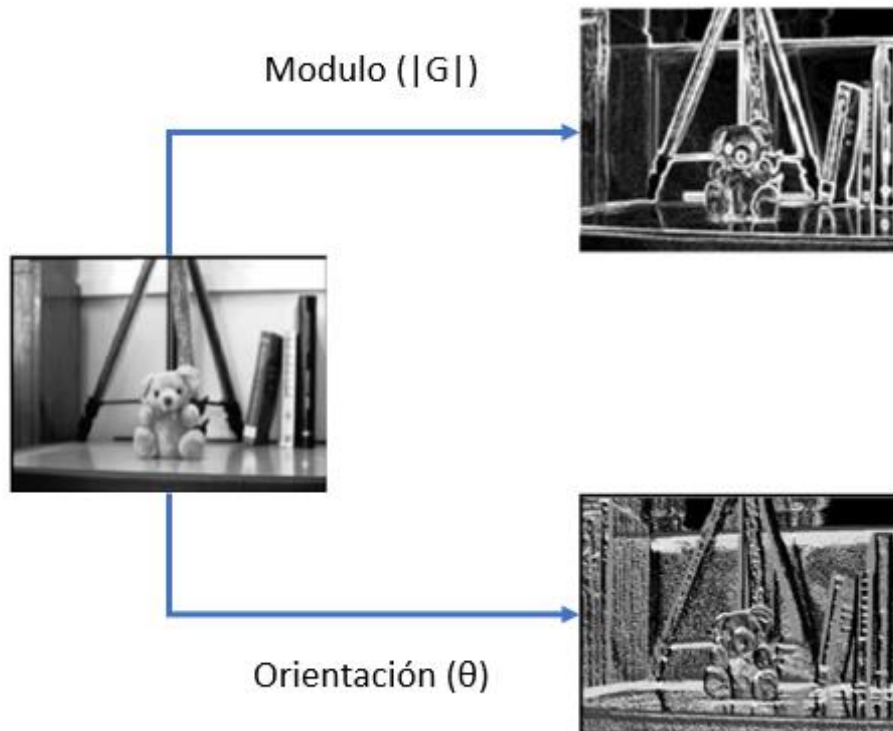


Figura 24: Representación del módulo y la orientación de los bordes de una imagen.

- Canny

Este método es el más interesante de los que aplican esta técnica ya que, además de extraer los bordes, cierra los contornos.

Tiene su fundamento en la técnica de la derivada, pero aplica a otras técnicas para mejorar la detección de estos.

El proceso de detección se fundamentó en tres fases:

- Suavizado: A partir de un filtro gaussiano.
- Obtención del gradiente (G y θ) en cada píxel.
- Adelgazamiento de los bordes detectados hasta reducirlo a un píxel.

Los métodos que aplican esta técnica aplican un umbral a partir del que se considera un píxel como borde o no. Eso acarrea ciertos problemas. Si se define un umbral demasiado alto, se pierden parte de los bordes. En el caso contrario, se introduce demasiado ruido.

El método Canny introduce una nueva forma de definir umbrales a partir de uno superior y otro inferior. Se consideran bordes válidos en los casos que se supere el

umbral superior. Pero también se consideran válidos los píxeles vecinos que se encuentren dentro de los umbrales superior e inferior.

$$I_n(u, v) > T_2(\text{pixels con variación fuerte})$$

$$T_1 < I_n(u, v) < T_2(\text{pixels con variación débil})$$

De esta forma obtenemos un conjunto de bordes conectados de contornos mucho más limpio en la que desaparecen muchos de los píxeles interiores a los bordes.

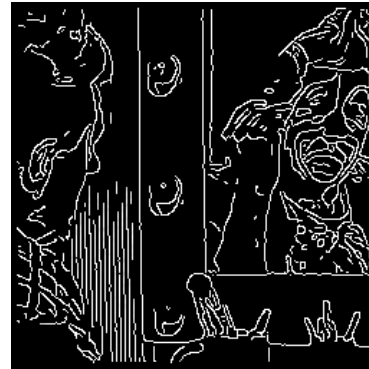
Es posible que haya que añadir un último paso de cierre de contornos que pudiesen haber quedado abiertos por problemas de ruido.



a



b



c



Figura 25: Visualización de la imagen de entrada (a) después de pasar por el filtro Canny: (b) Filtro gaussiano con desviación (σ) 1, umbral superior (T_1) fijado en valor 255 y umbral inferior (T_2) situado en 1; (c) $\sigma = 1$, $T_1 = 255$ y $T_2 = 250$; (d) $\sigma = 1$, $T_1 = 128$ y $T_2 = 1$; (e) $\sigma = 2$, $T_1 = 128$ y $T_2 = 1$.

- Segunda derivada
 - Laplaciana

La segunda derivada tiene una respuesta más fuerte ante pequeños detalles en variaciones de la imagen. De forma que se emplea esta para detectar pequeños puntos con variaciones que existentes en la imagen.

$$\nabla^2 f(u, v) = \frac{\partial^2 f(u, v)}{\partial u^2} + \frac{\partial^2 f(u, v)}{\partial v^2} \quad (ec. 11)$$

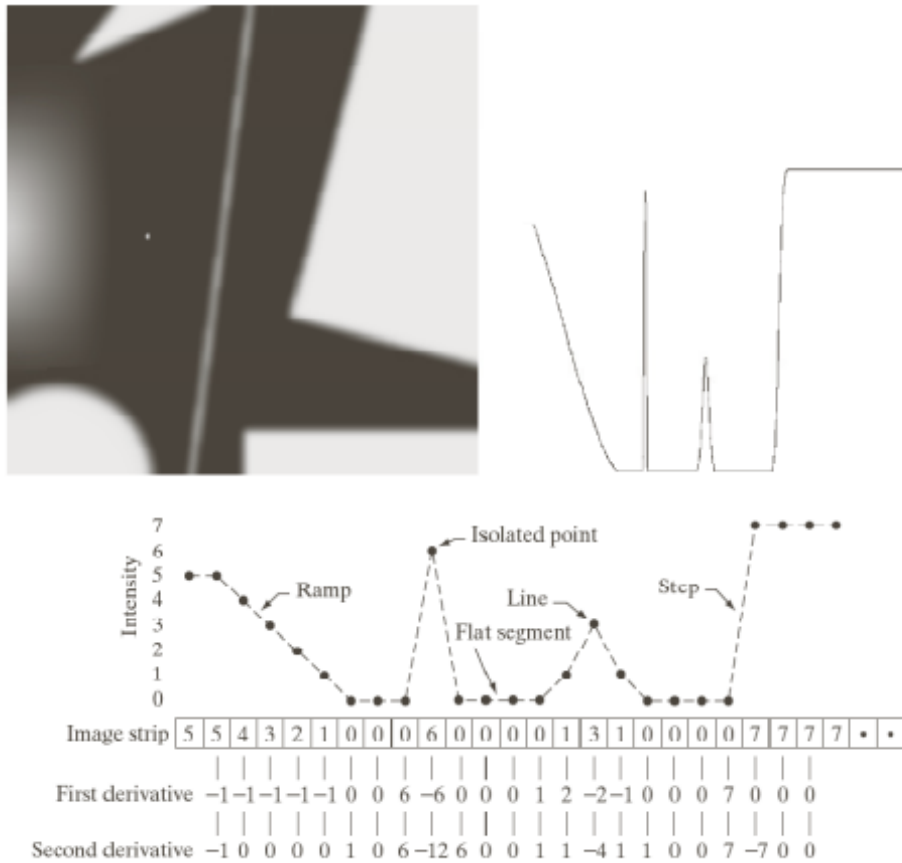


Imagen 9: Vista de la imagen analizada, los niveles observados en una línea y los valores obtenidos a partir de la aplicación de la primera y segunda derivada.

Para poder ajustar ese nivel de sensibilidad en la detección de variaciones, se aplica el método a partir de filtros. Cuanto mayor es el tamaño de estos, menor será la definición en la detección.

$$h = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$h = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Figura 26: Representación de los filtros matriciales aplicados al filtro Laplaciano.



Máscara 5x5

Máscara 9x9

Imagen 10: Resultado de imágenes filtrada con filtro Laplaciano.

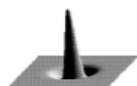
- Laplaciana de la Gaussiana

Dispone de un filtro Gaussiano que suaviza la imagen (igual que se hace con el método Canny) antes de pasar por el detector.

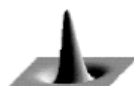
$$G(u, v) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(u^2+v^2)}{2\sigma^2}\right] \quad (\text{ec. 12})$$

$$\nabla^2 G(u, v) = \frac{1}{\pi\sigma^4} \left[1 - \frac{(u^2 + v^2)}{2\sigma^2}\right] \exp\left[-\frac{(u^2 + v^2)}{2\sigma^2}\right] \quad (\text{ec. 13})$$

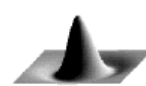
Aplica la técnica del operador sombrero mexicano.



$$\sigma^2 = 0.5$$



$$\sigma^2 = 1.0$$



$$\sigma^2 = 2.0$$

Máscara 13x13

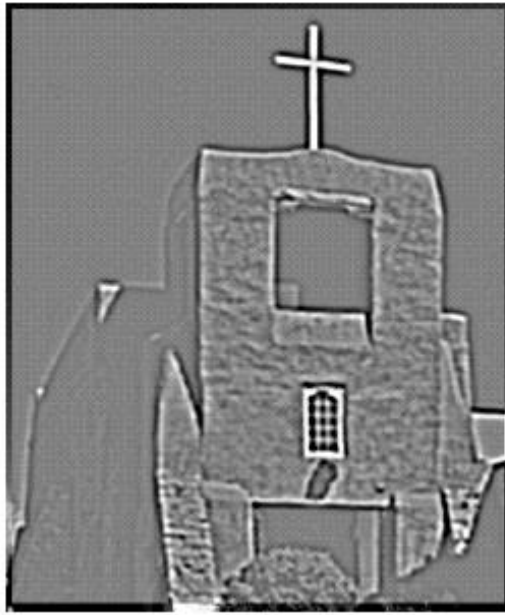


Imagen 11: Resultado de una imagen filtrada a partir del método Laplaciano de la Gaussiana.

Una vez conocidos todos los métodos de detección, se puede enumerar las formas de detectar segmentos de línea recta y de circunferencias.

El método mayormente empleado para esto es la **transformada de Hough**.

En la detección de líneas rectas, cada elemento de borde detectado vota por una línea recta a la que pueda pertenecer.

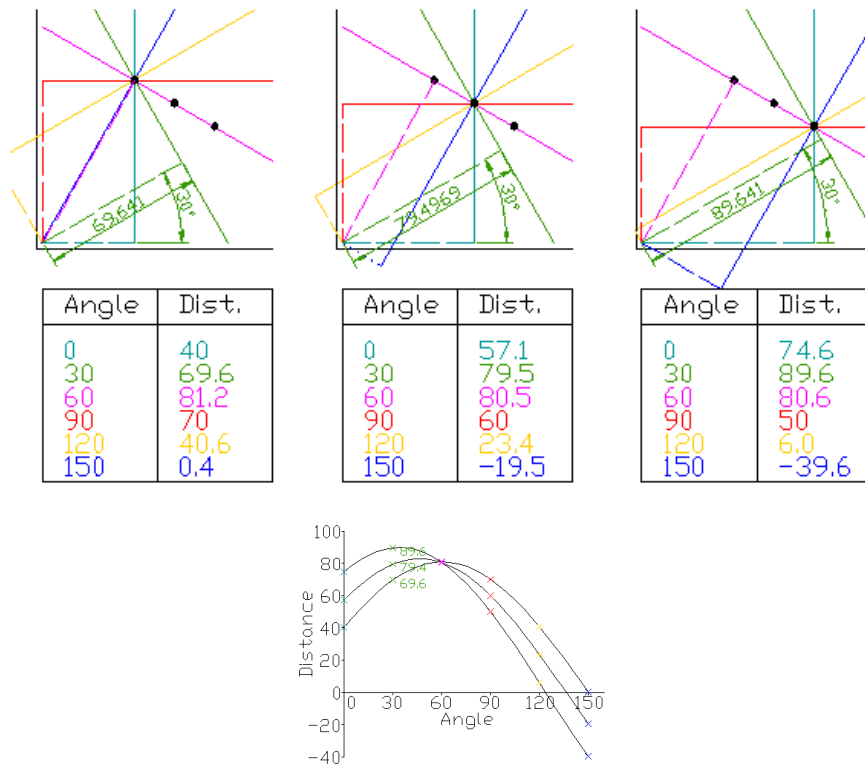
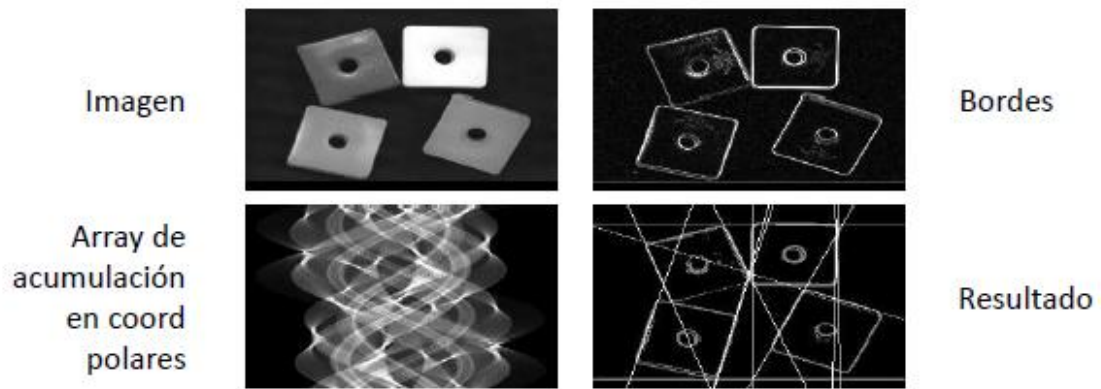


Figura 27: Representación de la transformada de Hough a partir de tres puntos situados en línea. En la parte superior se muestran los ángulos y la distancia de los distintos planos respecto del origen. En la gráfica inferior se muestra una visualización de los distintos puntos detectados y una clara convergencia en el ángulo 60°.

Aquellos puntos en el espacio, que reciben un gran número de votos, se convierten en líneas en la imagen.



También es empleado para obtener formas geométricas como circunferencias y elipses.

Detección de líneas. Parámetros: (r, θ)

$$r = x \cos\theta + y \operatorname{sen}\theta \quad (\text{ec. 14})$$

Detección de círculos. Parámetros: (r, x_0, y_0)

$$r^2 = (x - x_0)^2 + (y - y_0)^2 \quad (\text{ec. 15})$$

Detección de elipses. Parámetros: $(r_x, r_y, \theta, x_0, y_0)$

$$1 = \frac{(x - x_0)^2}{r_x^2 \cos^2\theta} + \frac{(y - y_0)^2}{r_y^2 \operatorname{sen}^2\theta} \quad (\text{ec. 16})$$

Segmentación basada en Regiones

Para comenzar la segmentación de una imagen aplicando esta técnica, se escoge un conjunto de puntos semilla y se van haciendo crecer las regiones a su alrededor.

La unión de todas las regiones de la imagen $\longrightarrow R = \bigcup_{i=1}^K R_i$

Ningún píxel pertenece a más de una región $\longrightarrow R_i \cap R_j = \emptyset \forall i, j; i \neq j$

Coherencia espacial $\longrightarrow R_i$ es una región conectada

Coherencia de características \longrightarrow Para algún predicado P:

$$P(R_i) \text{ es cierto para } i = 1, 2, \dots, K$$

$$P(R_i \cup R_j) \text{ es FALSO para } R_i, R_j \text{ adyacentes e } i \neq j$$

Entre las condiciones que se deben tomar para considerar un píxel perteneciente a una misma región, existen las siguientes:

- El píxel debe ser adyacente a la región.
- El píxel debe ser similar a los píxeles de esa región.

El criterio de similitud empleado consiste en la aceptación de todos aquellos que la diferencia de características con respecto a los píxeles de la región es inferior a un determinado valor umbral (T).

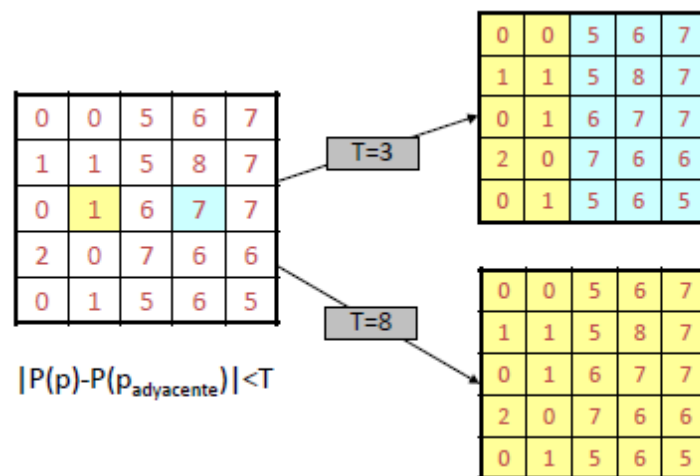


Figura 28: Representación de los resultados de un array de píxeles sometidos a un umbral T.

Aplicando esta técnica se pueden obtener resultados como los que se pueden observar en las imágenes a continuación.

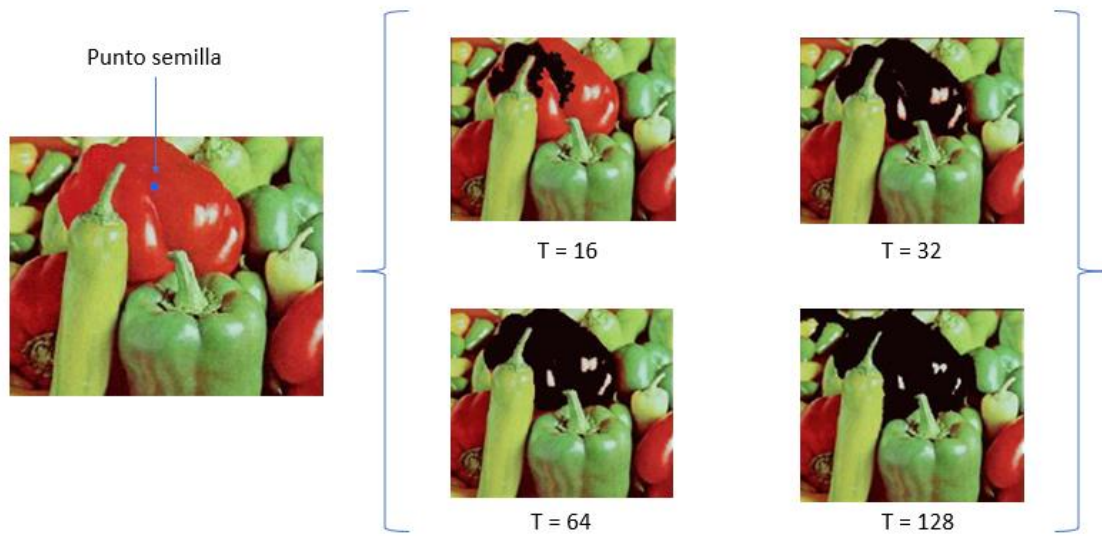


Imagen 12: Visualización de los resultados de la segmentación por crecimiento de regiones según el umbral seleccionado.

HARDWARE

Existen innumerables formas de extraer imágenes de un lugar y enviarlas a un servidor remoto que las vaya almacenando. En este caso, al tener datos de un sensor, se requiere de un sistema que esté conectado a un repositorio de imágenes y una base de datos donde ir almacenando toda la información que se va recogiendo.

Para esta labor existían dos opciones. Se podría realizar un diseño del sistema o trabajar con uno existente en el mercado. El diseño de uno nuevo supone la realización de un análisis de la cámara que se va a integrar y el sensor más apropiado. Por otra parte, se debería diseñar un sistema donde conectar los dispositivos y estar dotado de un modem que conectara el mismo con alguna plataforma. Y, finalmente, diseñar el flujo que van a seguir los datos por la red y la plataforma donde se van a almacenar, visualizar y prestar servicios para su extracción desde otras plataformas (conexión SSH, ...). Todo esto sin pararse en mencionar los aspectos de seguridad de este. Por otra parte, existe mucha industria que ya tiene desarrolladas todas estas funciones en entornos como los que se presentan en este proyecto y a precios muy competitivos.

Dado que la finalidad del proyecto está centrada en la creación de un modelo para aplicar a sistemas de visión artificial, se ha decidido emplear un sistema desarrollado que facilite la captura de imágenes sin tener que desarrollarlo. Para ello se ha decidido contar con los dispositivos de PESSL al tratarse de una empresa, con mucha presencia en este país, con mucha trayectoria en el sector agrícola. Esta dispone, también, de una plataforma donde los dispositivos van enviando los datos y esta permite visualizarlos y extraerlos. Concretamente, hablamos de la plataforma FIELDCLIMATE (<https://ng.fieldclimate.com/>).

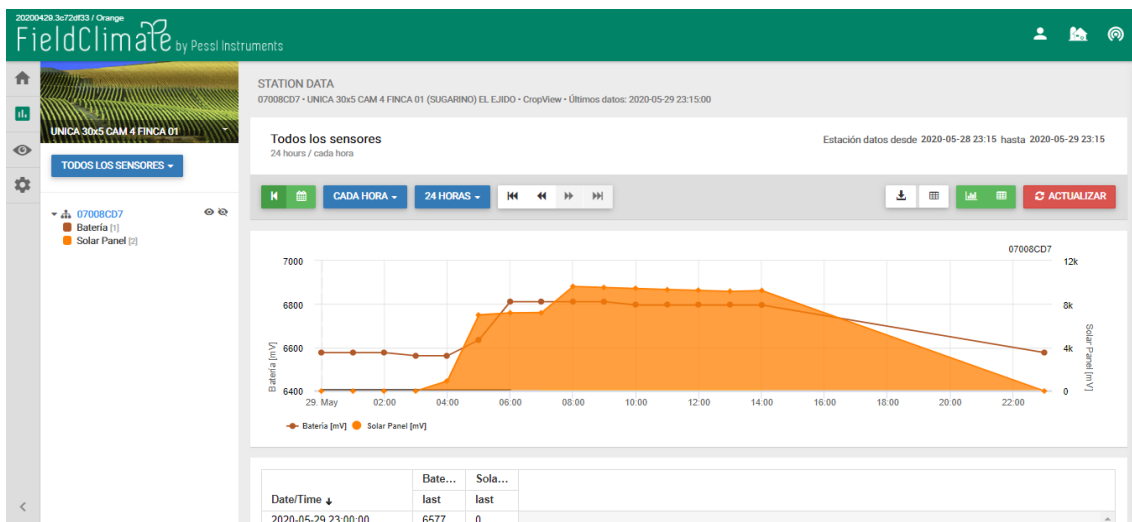


Imagen 13: Vista del portal FIELDCLIMATE

Ubicación de las cámaras

En cada localización a estudio, dos en este caso, se va a realizar la instalación de una estación dotada de cámaras (una estándar y otra panorámica) y un sensor de radiación solar.

Las cámaras estarán enfocadas a una planta situada a una distancia mínima de dos y máxima de cuatro metros.

El sensor de radiación solar permite medir el índice de radiación solar recibido por la planta. Esto nos ayudará a estudiar la influencia de este factor sobre el espectro reflejado por las mismas, lo cual pudiera influir en los resultados obtenidos.



Imagen 14: Cámara instalada en el invernadero.

Una vez revisado este fundamento, se puede observar la posibilidad de poder realizar mediciones en objetos situados en el plano analizado.

En este proyecto, se dispone de dos cámaras con objetivos diferentes. Una de ellas es de longitud focal corta y otro de longitud focal larga.

SOFTWARE

Matlab



MATLAB es un sistema de cómputo numérico con un entorno de desarrollo integrado (IDE) y lenguaje de programación propios. El sistema está desarrollado por MathWorks y dispone de licencia de uso.

Entre sus principales funciones básicas se encuentra la de manipulación de matrices, representación de datos y funciones, creación de interfaces de usuario y la comunicación con programas en otros lenguajes, además de otros dispositivos hardware.

El lenguaje que emplea es interpretado y puede ejecutarse en entorno interactivo y a través de un archivo de script.

En este proyecto, se van a emplear las funciones de procesamiento de imágenes disponibles y las herramientas propias recogidas en la toolbox de Procesamiento de Imágenes y Visión Computacional.

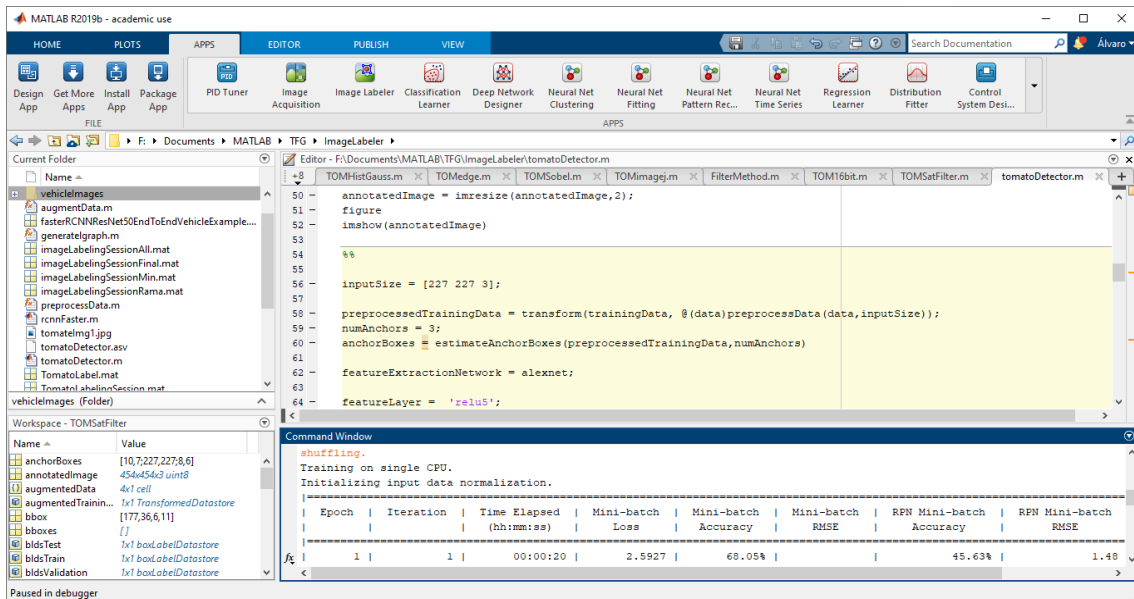


Imagen 15: Vista del menú principal del programa MATLAB.

El empleo de esta herramienta en el proyecto facilita el desarrollo del modelo de clasificación definitivo y el algoritmo que va a proveer la información extraída de cada imagen. También permite el desarrollo de un panel para la interacción del usuario con el sistema y proceder al análisis de los resultados obtenidos.

ImageJ

ImageJ
Image Processing and Analysis in Java



Imagej es un programa para el procesamiento de imágenes bajo licencia de código abierto. Está desarrollado en lenguaje JAVA, permitiendo su uso en los principales sistemas operativos (Windows, Mac OS y Linux).

Permite la visualización, edición, análisis, procesado, almacenamiento e impresión de imágenes en 8, 16 y 32 bits.

Entre sus numerosas herramientas, permite el cálculo de áreas y visualización de valores estadísticos en píxeles seleccionados, medición de distancia y ángulos, visualización de histogramas de densidad y gráficas con valores en línea y procesamiento de funciones para manipulación del contraste, afinado, alisado, detección de bordes y filtrado.

También tiene la capacidad para realizar transformaciones geométricas (escalado, rotación, giros, ...).

También permite la instalación de más características a través de plugins adaptados al programa.

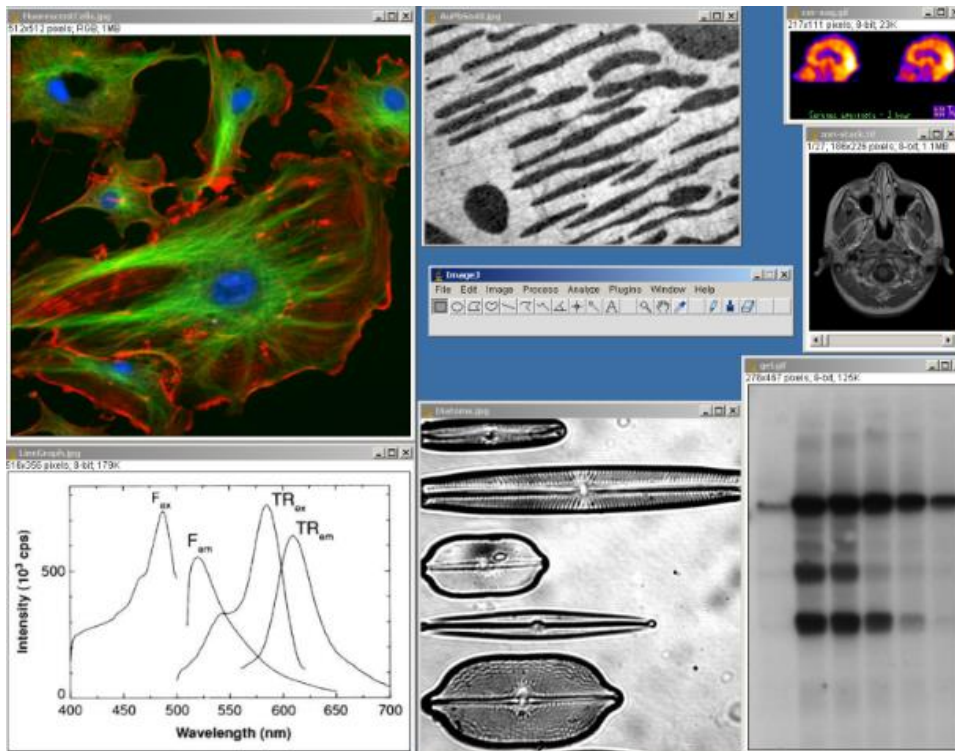


Imagen 16: Vista del menú de la aplicación ImageJ con algunas pantallas de visualización de resultados.

Esta aplicación va a permitir al desarrollador realizar análisis y pruebas a mayor velocidad que directamente con el programa empleado para el desarrollo del algoritmo definitivo. Eso es posible gracias al panel de herramientas que dispone con accesos a filtros y máscaras, además de poseer numerosos métodos de análisis y procesamiento predefinidos. Por otra parte, al estar desarrollado en lenguaje JAVA, permite incluir algoritmos desarrollados por uno mismo o descargados desde terceras personas.

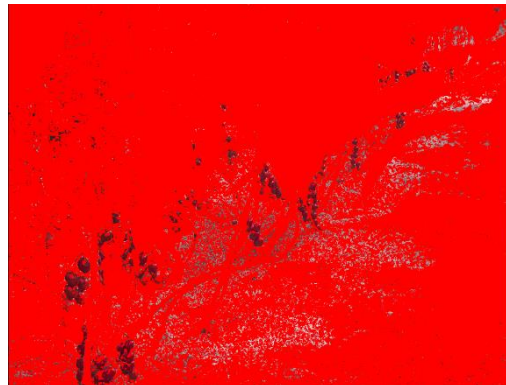
DESARROLLO DEL TRABAJO

Detección y conteo

Anteriormente se determinó que el modelo de color HSV es el que permite distinguir, en mayor grado, los frutos del resto de entorno. Se observó cómo los frutos reúnen unos índices en los planos matiz y saturación notablemente diferenciados.

Analizando el plano matiz se observa, después de realizar un filtrado de todos los píxeles con un umbral definido en el 20%, que permanecen en el plano todos aquellos relacionados, mayoritariamente, con frutos NO MADUROS. Aunque también permanecen elementos relacionados con el entorno que reúnen características similares.

Por otra parte, después de realizar un filtrado de todos los píxeles con un umbral definido en el 95%, se observa la permanencia en el plano de los píxeles relacionados con frutos MADUROS. En este caso, la proporción de estos con los relacionados con el entorno es notablemente superior. Aunque no es total.



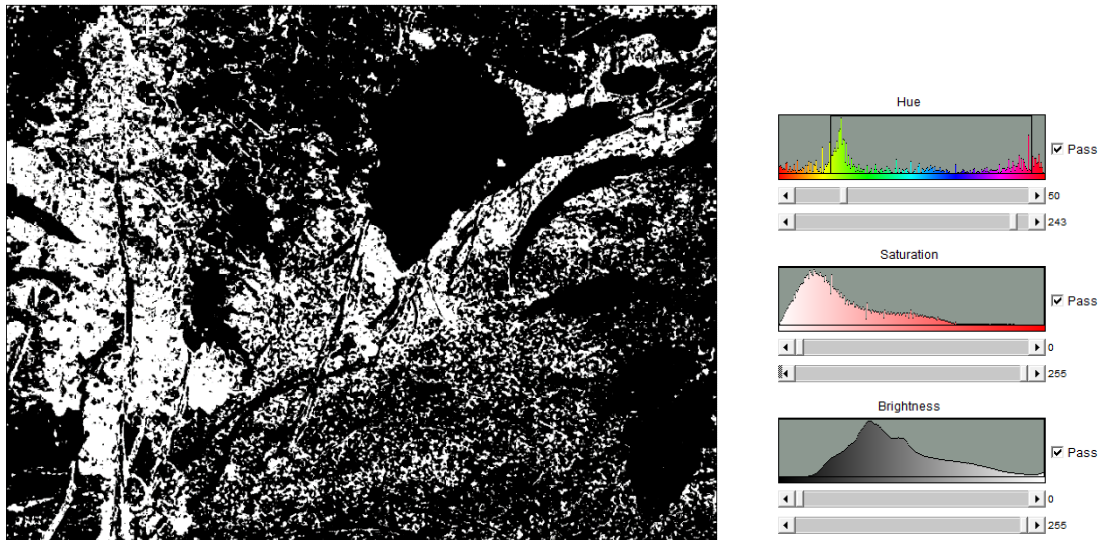


Imagen 17: Filtro de matiz situado entre umbral inferior al 20% y superior al 95%.

En el plano de saturación se observa que los frutos reúnen unos valores por encima de cierto umbral. El resto del entorno, sin embargo, dispone de numerosos componentes con valores muy bajos debido a, como se menciona en la definición del componente, la gran influencia del color blanco. Esto se aprecia en zonas con alto reflejo de luz (en las hojas y el suelo), y el tono claro de las paredes del invernadero. El área de los frutos que está sometida a este efecto es muy reducida dada su geometría (pasando prácticamente desapercibido en el caso de los frutos maduros).

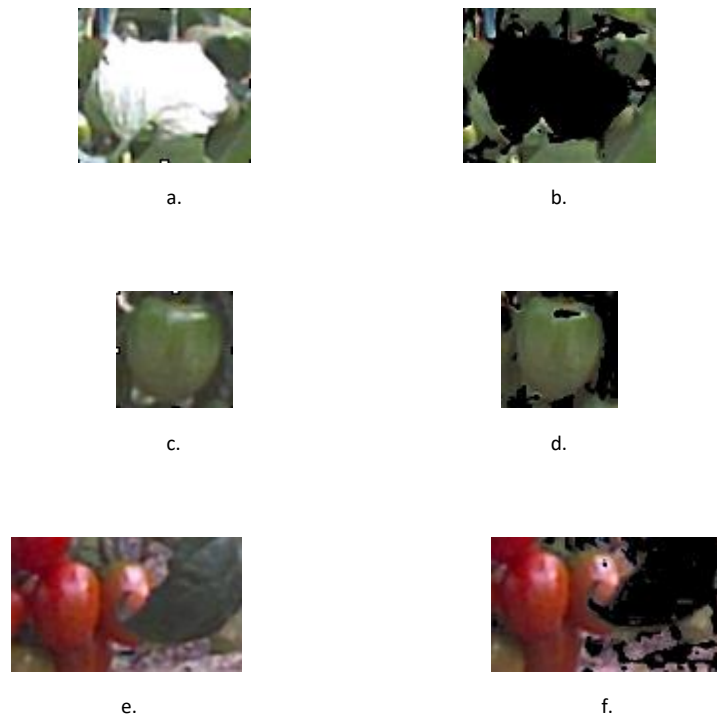


Figura 29: Visualización del efecto del filtro antirreflejo luminoso.

Imágenes originales (a, c, e). Imágenes después del filtro antirreflejo luminoso (b, d, e).

Una vez realizado este análisis, se procede a realizar un filtrado de todos los píxeles con valores inferiores al umbral situado en el 15% obteniendo el resultado mostrado en la imagen.

A partir de este filtro, se posibilita la eliminación de todos aquellos elementos con bajos niveles de saturación como las paredes del invernadero, gran parte del suelo y vegetación con luz reflejada.

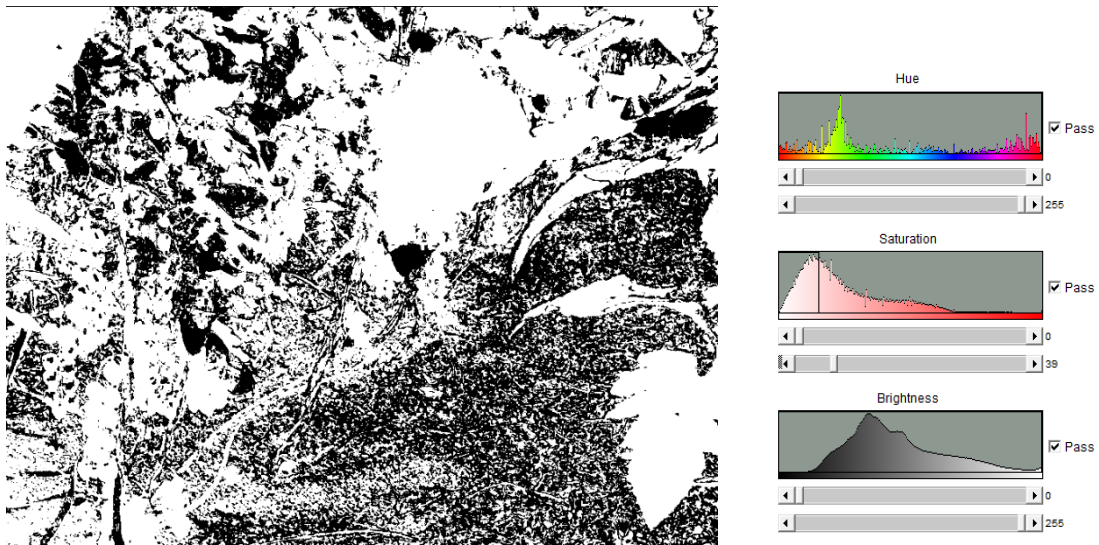


Figura 30: Filtro de saturación situado con un umbral inferior al 15%.

Existen elementos del entorno que cumplen las condiciones marcadas por los filtros de matiz y saturación. Por ejemplo, vegetación que no recibe luz directa o partes del suelo que pueden tener un elemento de matiz elevado, pero, al no reflejar luz, puede disponer de un nivel de saturación elevado y encontrarse en el espectro de los frutos. Sin embargo, al encontrarse en esas condiciones, se aprecia un patrón común de brillo con niveles por debajo de los que disponen los frutos.

Situando un filtro con los niveles de brillo bajo el umbral del 20%, se consigue distinguir las zonas oscuras del entorno, que pueden disponer de un nivel de saturación elevado, del resto del entorno. Sin embargo, todos los frutos se encuentran en niveles superiores, por lo que no son afectados por el citado filtro.

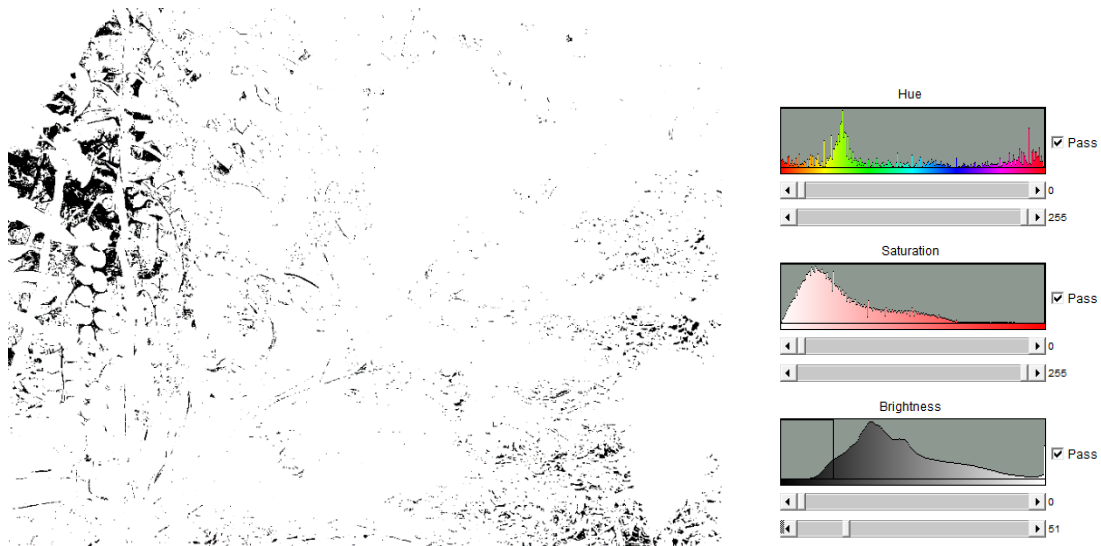


Figura 31: Filtro de brillo situado con un umbral inferior al 20%.

Después de todos los análisis previos realizados, tras superponer todos los filtros generados, se obtiene el resultado mostrado en la figura.

Como se puede observar en la figura, la imagen de los frutos apenas queda oculto por el filtro. Pero existen ciertos elementos del entorno, aunque de forma muy difusa, que tampoco les afecta.

Para solucionar esos elementos dispersos del entorno, se procede a aplicar la técnica de erosión al filtro. De esta forma, eliminamos los elementos menores del diámetro elegido.

Como se puede observar en la imagen, únicamente permanecen los elementos que, mayoritariamente, pertenecen a los frutos.

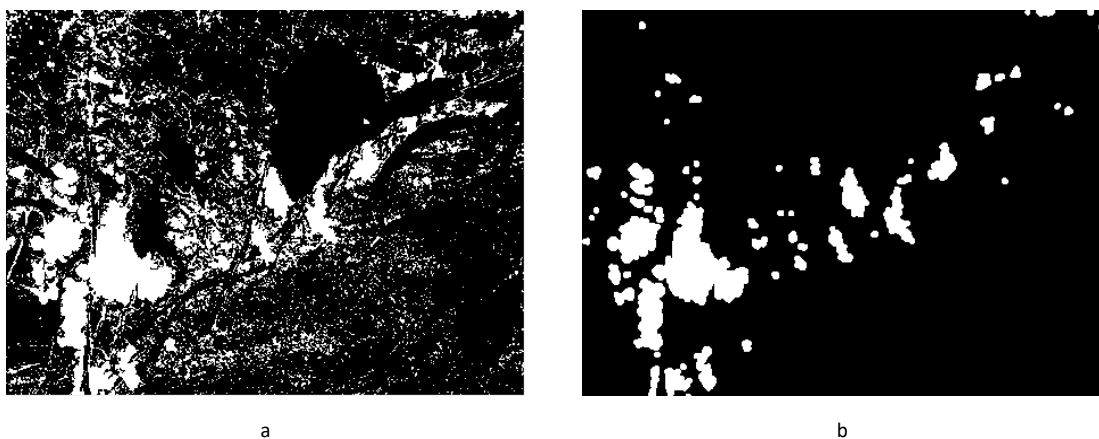


Figura 32: imagen después del filtro total. En la figura (a) se trata de la imagen obtenida directamente en la salida. En la imagen (b) se muestra después de someterla a un proceso de erosión.

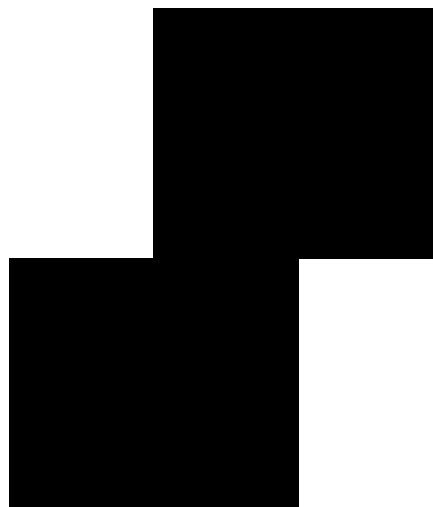
A continuación, se procede a realizar un análisis de cada elemento que permanece en el plan, después del proceso de filtrado previo.

Una vez que se han delimitado las regiones en las que se encuentran los frutos, se procede a realizar un análisis particular a cada una de ellas. En este análisis se va a permitir realizar un segundo análisis aplicando filtros que anteriormente no eran válidos ya que había elementos del entorno que lo cumplían.

Para la realización de este nuevo análisis, se va a proceder a la realización de un barrido por regiones empleando la herramienta “`regionprops`” que nos permite, además, extraer características de cada una.



a



b

c

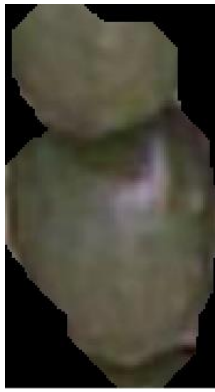


d



e

f



g



h



i



J



k



l

Imagen 18: Visualización de la región seleccionada (a, d, g, j), máscara resultante después de binarización con un umbral superior de matiz del 20% (b, e, h, k) y máscara resultante después de binarización con un umbral inferior de matiz del 95% (c, f, i, l).

Resultado

El resultado obtenido de este análisis es el conteo de frutos a partir del conteo de regiones independientes.

```
[IcropImage, IcropCounter] = bwlabel(IcropFilterErode);
```

Todos los conteos detectados, se procede a insertarlos en una matriz.

```
nArr = [nArr IcropCounter];
```

Y, finalmente, se realiza un sumatorio de todos ellos para obtener un resultado.

```
nArrSum = sum(nArr);
```

```
Número de frutos detectados: 101
```


Reconocimiento de estado de madurez

Una referencia internacionalmente aceptada y aplicada para definir el estado de maduración del tomate se encuentra en la tabla de clasificación del color publicada por el Departamento de Agricultura de los Estados Unidos. En ella se clasifican 6 tonalidades diferentes.

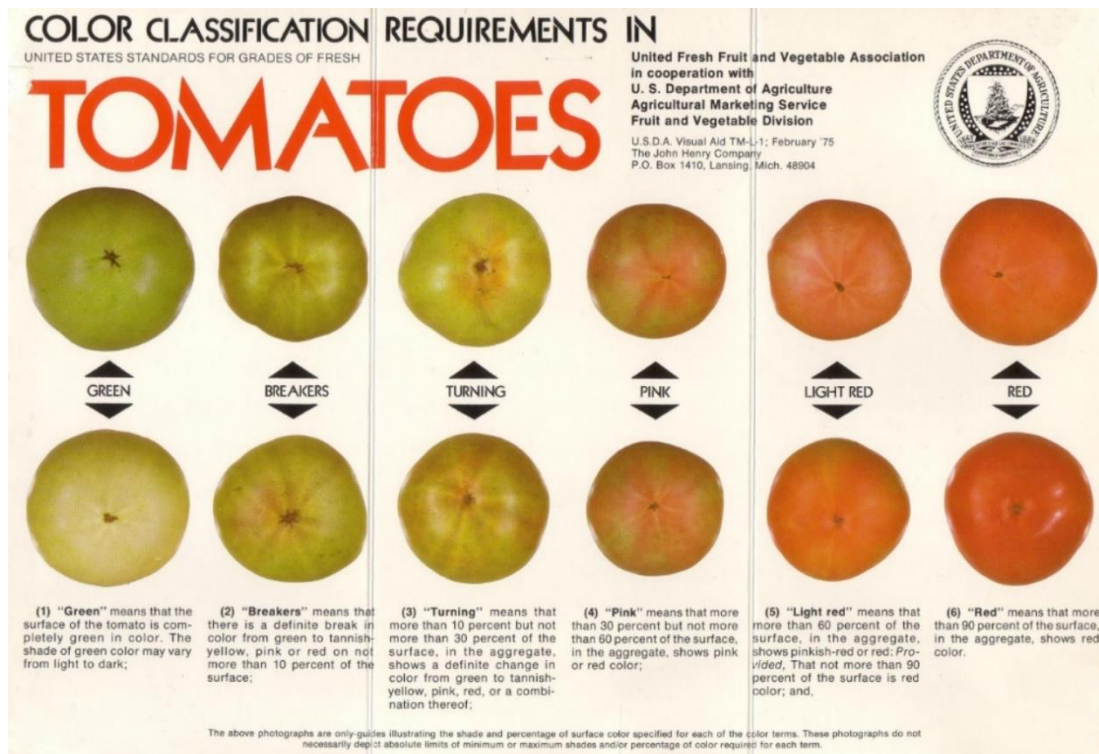


Imagen 19: Tabla de clasificación del color publicada por el Departamento de Agricultura de los Estados Unidos.

Esas tonalidades, como se ha visto antes, se distinguen claramente en el plano matiz de la componente HSV. Se puede visualizar en las siguientes imágenes.

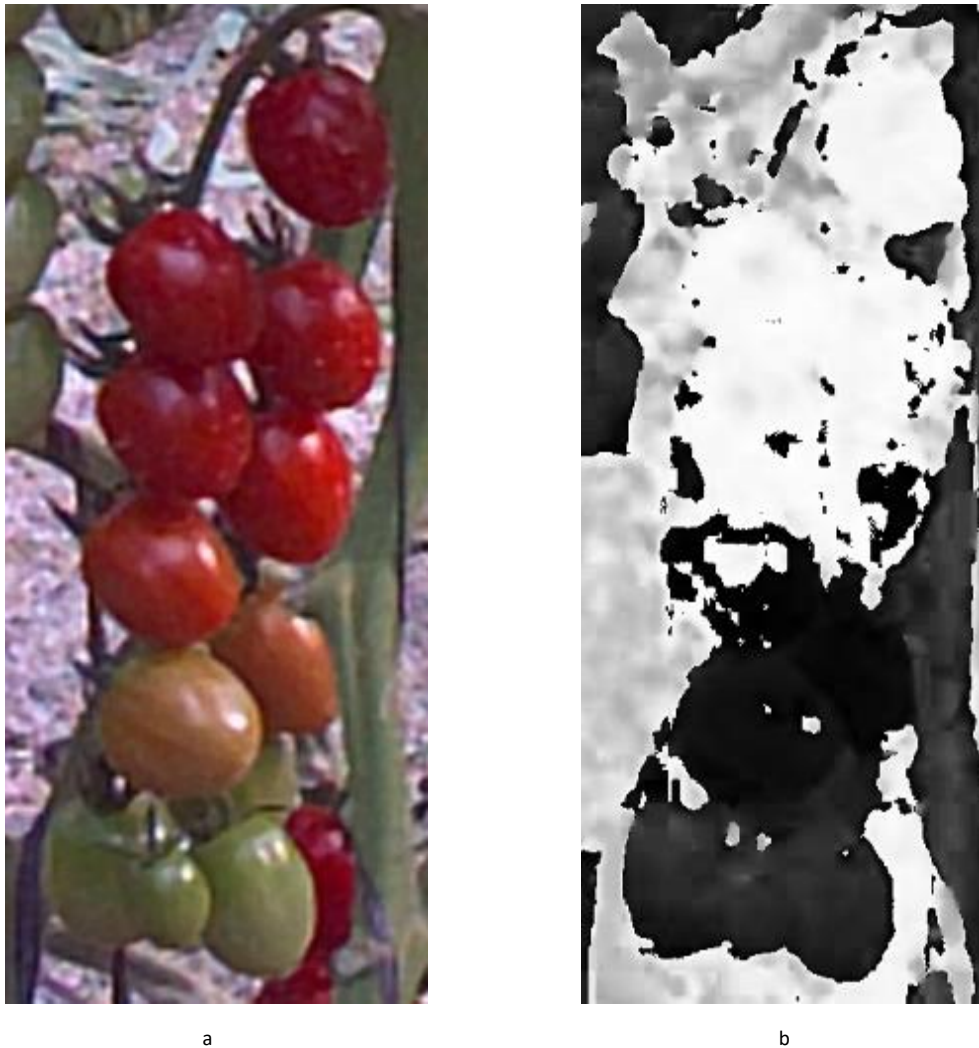


Imagen 20: Original (a). Capa matiz (b).

Con el fin de simplificar los estados, se puede tener como referencia tres estados, en vez de seis, siendo estos los estados maduros (“red” y “light red”), semi-maduros (“Pink” y “Turning”) y no maduros (“Green2” y “Breakers”).

En las imágenes, se puede observar cómo los frutos maduros se encuentran con valores cercanos al máximo (blanco), los que disponen de tonos anaranjados con valores cercanos al nulo (negro) y los no maduros con valores inferiores al 20% y superiores al 10% (25-50 de 255 aprox.).

Resultado

El resultado obtenido de este análisis es la obtención de la ratio de ocupación de frutos en un estado de madurez u otro.

Para ello, en primer lugar, realizamos la detección del área en la que se encuentran posibles frutos.

```
zoneArea = nnz(
    ((imcomplement(imbinarize(IcropHsv(:,:,1),0.2))
    |imbinarize(IcropHsv(:,:,1),0.95))
    & IcropHsv(:,:,3)>0)==1
    );
```

Para filtrar esa área de la región, únicamente nos basaremos en realizar un filtrado de todos aquellos píxeles que estén dentro de los valores, en la capa matiz, identificados para los frutos.

A continuación, se extraerá el área de los píxeles que pertenecen a cada estado.

```
maduro = nnz(
    (imbinarize(IcropHsv(:,:,1),0.95)
    &IcropHsv(:,:,3)>0)==1)/zoneArea;
semi = nnz(
    (imcomplement(imbinarize(IcropHsv(:,:,1),0.1))
    &IcropHsv(:,:,3)>0)==1)/zoneArea;
noMaduro = nnz(
    ((imcomplement(imbinarize(IcropHsv(:,:,1),0.2))
    &imbinarize(IcropHsv(:,:,1),0.1))
    &IcropHsv(:,:,3)>0)==1)/zoneArea;
```

Para la extracción se realiza un conteo de los píxeles no nulos localizados dentro del filtro de cada estado. Finalmente, relativizamos el valor respecto de los píxeles totales.

Todos estos valores, relativizados, se deben ir añadiendo a un array de valores que, al final, emplearemos para obtener la esperanza de cada estado.

```

maduroArr = [maduroArr maduro];
semiArr = [semiArr semi];
noMaduroArr = [noMaduroArr noMaduro];

```

Como resultado definitivo de esta sección del modelo, se extrae la fracción ocupada por los píxeles de cada estado respecto del total de la imagen. Para realizar ese cálculo, se aplica la técnica de extracción del índice de esperanza estadística.

$$E[X] = x_1p(X = x_1) + \dots + x_np(X = x_n) = \sum_{i=1}^n x_ip(x_i) \quad (\text{ec. 17})$$

Explicación de esperanza matemática [11]

Para poder realizar esta operativa, además de los datos recogidos anteriormente, vamos a necesitar el peso, sobre el total de la planta, de cada valor relativo a cada región. Para ello se debe disponer de un vector con el número de frutos detectados en cada región (`nArray`) y multiplicar cada uno por cada valor del vector con los valores relativos en cada estado.

```

maduroArrn = nArr.*maduroArr;
semiArrn = nArr.*semiArr;
noMaduroArrn = nArr.*noMaduroArr;

nArrSum = sum(nArr);

maduroAvg = sum(maduroArrn)/nArrSum;
semiAvg = sum(semiArrn)/nArrSum;
noMaduroAvg = sum(noMaduroArrn)/nArrSum;

```

Finalmente, dividiendo el total entre el valor de frutos contabilizado en la planta, obtenemos el valor relativo total de la misma.

Estado de madurez en los frutos:		
MADUROS	SEMI-MADUROS	NO MADUROS
-----	-----	-----
26.88%	32.30%	40.82%

Calibrado del fruto

Para el correcto calibrado de los frutos se necesita desarrollar una herramienta a partir de los fundamentos mostrados en el capítulo “Principios básicos de óptica”. Concretamente en el apartado “Ángulo de visión”.

Medición de objetos

La teoría, anteriormente mostrada, muestra la posibilidad de obtener el ángulo de visión a partir de la distancia focal de la lente y la longitud del sensor (vertical, horizontal o diagonal).

Llevado a la práctica, se puede visualizar esa teoría en un escenario como el mostrado en la figura.

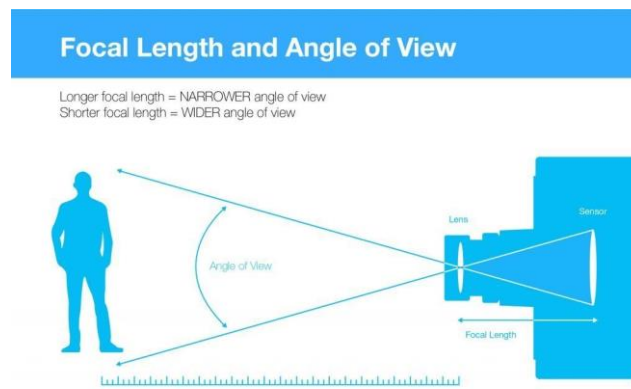


Figura 33: Visualización de la relación del ángulo de visión con el tamaño del sensor.

En uno de los lados se contempla la imagen que se encuentra frente a la cámara y en el otro la propia cámara con su lente y el sensor interno compuesto de elementos fotosensibles que recogen la intensidad lumínica recibida y la digitalizan.

Con los datos técnicos, aportados por el fabricante de las cámaras, se pueden extraer los datos necesarios para obtener el ángulo de visión y, por consiguiente, el tamaño de la imagen.

ESPECIFICACIONES TÉCNICAS

Caja	Fuente de alimentación y caja de soporte del sensor. 41 cm L x 13 cm W x 7 cm H
Peso sin sensores	2.2 kg
Módulo de cámara	Soporte de acero inoxidable con caja IP65 de 27 cm L x 17 cm W x 9 cm H, peso: 1.5 kg
Fuente de Poder	Batería de plomo de 6V y 4.5 AH con panel solar
Modelo/Tipo	Módulo de procesador Cortex M4 con modelo de comunicación integrado para operación UMTS/LTE
Cámara y óptica	Sensores CMOS MT9J003 de 10 Mega Pixel 2/3"- Óptica DSL377A- Lentes 650-F2.8 2/3" con 2.5 mm de longitud focal y lentes DSL901]- 650-F3.0 2/3" con 12 mm de longitud focal.
Sensores meteorológicos opcionales compatibles (disponible pronto)	Pluviómetro de 0.2 mm (0.01 pulg.), sensor de humedad de la hoja, Temperature del suelo o aire, sensor de temperatura y humedad relativa

Tabla 1: Especificaciones técnicas de las cámaras instaladas.

Se dispone de sensores CMOS MT9J003 de 10 Megapíxel y un tamaño del sensor de 2/3" (8,8 x 6,6 mm [8]). En el caso de la cámara con objetivo tipo panorama, dispone de una lente con una longitud focal de (f) 2,5mm. En la cámara con objetivo tipo zoom, dispone de una lente con una longitud focal (f) de 12mm.

Con toda esa información, se puede obtener los siguientes datos de la imagen:

$$\alpha = 2 \arctan \frac{2h}{2f} \quad (\text{ec. 18})$$

(siendo "2h" la longitud del sensor vertical u horizontal)

	Zoom	Panorama
α_{vertical}	30,75°	105,7°
$\alpha_{\text{horizontal}}$	40,27°	120,8°

Tabla 2: Ángulos de visión horizontal y vertical de las cámaras (zoom y panorama).

Una vez obtenido el ángulo de visión de la cámara, se puede obtener el tamaño de la imagen "2H" a partir del ángulo "α" (previamente calculado) y la distancia del objeto "D":

$$2H = 2D \times \tan \frac{\alpha}{2} \quad (\text{ec. 19})$$

En las instalaciones realizada, disponemos de cámara ubicada a 2,38 y 3,6 metros de la planta. De forma que se puede construir la siguiente tabla de resultados:

	Zoom	Panorama
2H Vertical (D=2m)	1,1 m	5,28 m
2H Vertical (D=2,38m)	1,31 m	6,28 m
2H Vertical (D=3,6m)	1,98 m	9,5 m
2H Horizontal (D=2m)	1,47 m	7,04 m
2H Horizontal (D=2,38m)	1,75 m	8,38 m
2H Horizontal (D=3,6m)	2,64 m	12,67 m

Tabla 3: Longitud vertical y horizontal de la imagen captada por los sensores instalados en las cámaras instaladas.

En la cámara tipo “zoom”, ubicada a 2,38 metros, se dispone de un plano de 1,31x1,75 metros y, en la ubicada a 3,6 metros, se dispone de un plano de 1,98x2,64 metros.

Algo que va a resultar muy útil, y necesario, para el cálculo del calibre de un fruto es conocer la escala real del entorno. Conociendo las dimensiones del plano real, se necesita descomponer en partes mínimas en las que se pueda delimitar el objeto que se necesita medir dentro la misma. Para ello se van a emplear los píxeles contenidos en el plano.

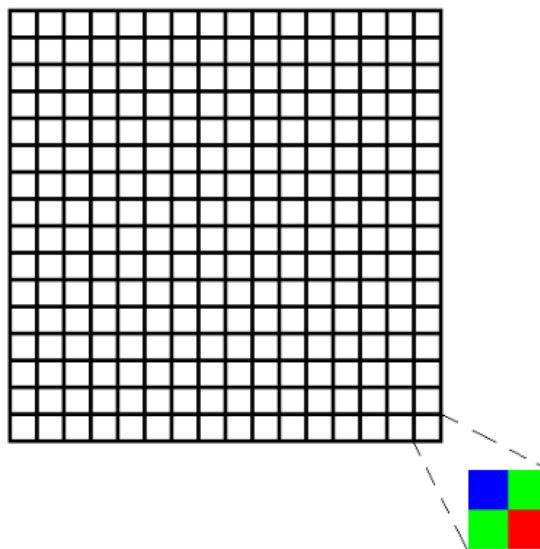


Figura 34: Representación de un píxel RGB en la matriz de la imagen

Conociendo la concentración de estos localizados en el sensor (10M píxeles) y las medidas de este (8,8x6,6 [8]), se puede deducir los que se encuentran en cada eje:

$$n_{pixel}(x) = \sqrt{nPixeles} * 2 * \frac{x}{x+y} \text{ (ec. 20)}$$

$$n_{pixel}(y) = \sqrt{nPixeles} * 2 * \frac{y}{x+y} \text{ (ec. 21)}$$

En la ecuación, primero se extrae el número de píxeles que corresponderían a un lado de un cuadrado ($\sqrt{nPixeles}$). Finalmente, se aplica el factor incremental del lado respecto al mismo ($\frac{x}{x+y}$).

De esta forma, se obtienen los siguientes datos de píxeles localizados en los ejes:

$$n_{pixel}(x) = 3614$$

$$n_{pixel}(y) = 2710$$

Este resultado obtenido es estimativo, siendo válido siempre que no existan imágenes extraídas, previamente, del dispositivo analizado. En este caso, al disponer de ellas, se pueden extraer los datos a partir de las mismas:

$$n_{pixel}(x) = 3664$$

$$n_{pixel}(y) = 2748$$

Como se puede observar, no se distancian, en exceso, de los datos estimados. Pero conociéndolo directamente del dispositivo, siempre aportará mayor precisión en la medición que se realice.

Una vez conocidas las dimensiones de la imagen y la cantidad de píxeles localizados en cada eje, se puede calcular la distancia entre ellos:

$$d = \frac{2H}{n_{pixel}} \text{ (ec. 22)}$$

	Zoom	Panorama
Alto (2 m)	0,4 mm	1,92 mm
Ancho (2 m)	0,4 mm	1,92 mm
Alto (2,38 m)	0,477 mm	2,29 mm
Ancho (2,38 m)	0,477 mm	2,29 mm
Alto (3,6 m)	0,72 mm	3,46 mm
Ancho (3,6 m)	0,72 mm	3,46 mm

Tabla 4: Escala de la imagen respecto a píxeles.

Como se puede observar la distancia entre píxeles en el eje 'x' e 'y' es la misma. Esto demuestra que los resultados obtenidos tienen una alta probabilidad de ser correctos.

Para terminar de demostrar estos resultados, se escogen 2 imágenes sacadas con distintas cámaras:



a



b

Imagen 21: Vista del calibrador en cámara zoom (a). Vista del calibrador en la cámara panorama (b).

En la realidad, el cuadrado mide 127 mm de lado. En el plano, el lado del cuadrado mide 384 píxeles en la imagen realizada con la cámara de objetivo tipo zoom. Esto quiere decir que, para extraer su medida, debe emplearse la siguiente ecuación:

$$l = 384 \times 0,477 = 183 \text{ mm}$$

Según el valor obtenido, la cámara dispone, además, de un **coeficiente zoom de 1,45**.

$$l = \frac{384 \times 0,477}{1,45} = 127 \text{ mm}$$

En otra imagen extraída, desde otra cámara, se ha obtenido un **coeficiente zoom de 1,55**. De forma que se puede aplicar un **coeficiente de 1,5** por defecto (produciendo un error cercano al 1% en la medida).

Para demostrar la teoría anteriormente mencionada, se prueba con la cámara de objetivo tipo panorama.

Para extraer su medida, se debe emplear la siguiente ecuación:

$$l = 50 \times 2,29 = 115 \text{ mm}$$

Al ser una cámara con objetivo tipo panorama, y dispone de un ángulo de visión de 105º, el elemento calibrador recoge un número muy reducido de píxeles, afectando ello a la pérdida de precisión (ya que un pequeño valor en el número de píxeles afecta notablemente al resultado).

Como se puede observar en la imagen, los bordes quedan muy difusos, lo que no permite definir claramente sus límites.

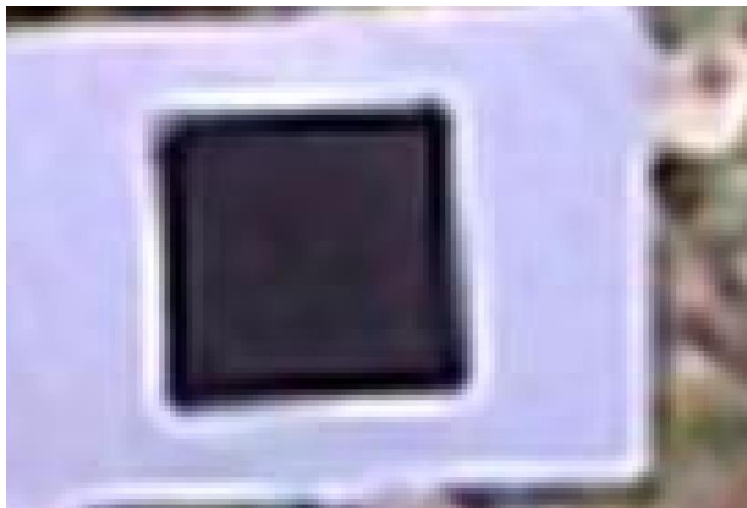


Imagen 22: Vista ampliada del calibrador en cámara panorama. Se aprecia la dificultad de definición del borde del cuadro.

En este proyecto, interesa disponer de la máxima definición del fruto y una visión lo más centrada en la planta a analizar. Esto evitará que la imagen disponga, en exceso, de

interferencias producidas por otros frutos situados en plantas que se encuentran fuera del plano analizado.

Por otra parte, también es importante tener la máxima definición de los frutos para poder diferenciar entre ellos y obtener un calibre lo más preciso posible.

Resultado

En cada región detectada, en el apartado anterior, se extrae el área en la que se localizan los frutos. En las regiones donde solamente se encuentra un fruto, el área detectada se trata del total del fruto. Pero existen regiones en las que se localizan una agrupación de frutos con bordes difícilmente detectables (sin contar los frutos con vista parcial).

La información que tenemos de la región es el área total de frutos detectado. Por otra parte, los frutos se pueden considerar que disponen de una forma geométrica cercana a la circular.



Imagen 23: Vista de la cercanía en la geometría del fruto respecto de un círculo.

De esta forma, al disponer, también, la relación métrica respecto a los píxeles, se aplica la relación del área circular con el radio y, así, obtener un calibre aproximado.

$$r^2 = \frac{A}{\pi} \text{ (ec. 23)}$$

Finalmente, en el modelo aplicamos la relación a cada región y se divide entre el número de frutos detectados para obtener el calibre medio.

```
meinAreapx = zoneArea/IcropCounter;  
meinGagemm = 2*sqrt(meinAreapx/pi)*scalepxmm/zoom;
```

Posteriormente, se va incluyendo cada valor de área a la matriz.

```
meinGagemmArr = [meinGagemmArr meinGagemm];
```

Definitivamente, se procede a obtener la media aritmética de todas las áreas detectadas.

```
meinGagemmAvg = sum(meinGagemmArr) / length(meinGagemmArr);
```

Y se obtiene un resultado.

```
Calibre Medio: 32.65 mm
```

Visualización de resultados

Como salida del modelo se consideran los siguientes resultados:

- Total de frutos detectados.

Cuenta total de los frutos detectados.

- Calibre medio.

Medición de calibre medio entre todos los frutos detectados.

- Cantidad relativa del estado de madurez.

Cuantificación de los frutos en cada estado analizado que se encuentra entre el total de ellos.

A la salida del modelo, se visualiza una tabla de resultados con los datos previamente calculados.

```
Visualización de resultados del modelo.  
Número de frutos detectados: 163  
Calibre Medio: 30.07 mm  
Estado de madurez en los frutos:  
MADUROS      SEMI-MADUROS  NO MADUROS  
-----  
36.71%      34.87%      28.41%
```

Imagen 24: Visualización de los resultados obtenidos a la salida del algoritmo.

EXPERIMENTOS Y RESULTADOS

Ejecución del modelo definitivo

A continuación, se procede a ejecutar el modelo y ver los resultados que se obtienen.

En primer lugar, se selecciona una imagen a introducir.

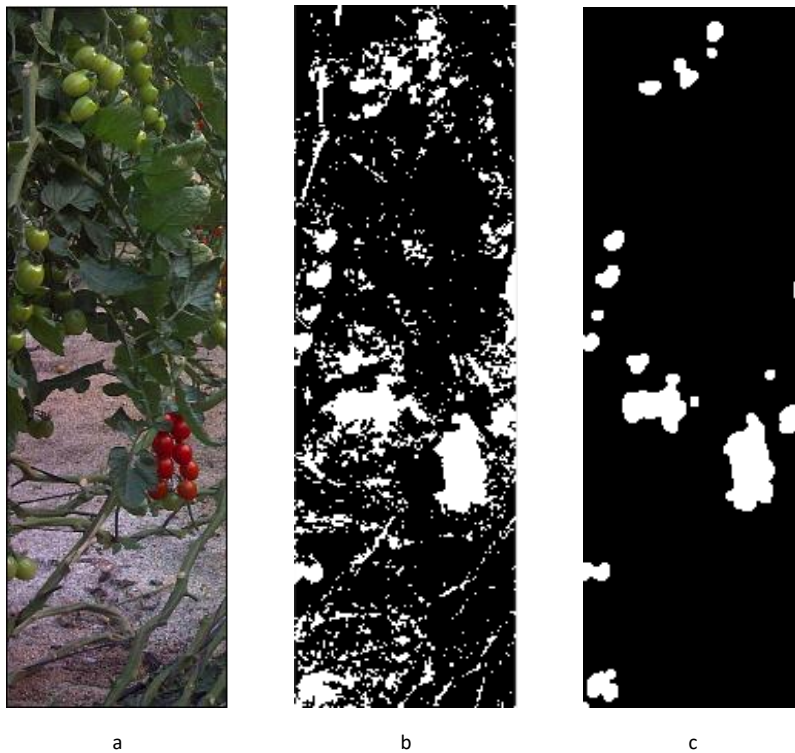


Figura 35: Representación del proceso de segmentación de la imagen.

En una primera vista, se observa la capacidad de reconocimiento, de una forma clara, en los frutos maduros. Los no maduros, algunos de ellos, los desestima al no pasar el filtro de vegetación.

A continuación, se procede al análisis por regiones.



Cuenta	Calibre (mm)	Ratio Maduro	Ratio Semi-Maduro	Ratio No Maduro
1	44,11	0,24%	3,93%	95,83%



Cuenta	Calibre (mm)	Ratio Maduro	Ratio Semi-Maduro	Ratio No Maduro
1	40,65	0%	0%	100%



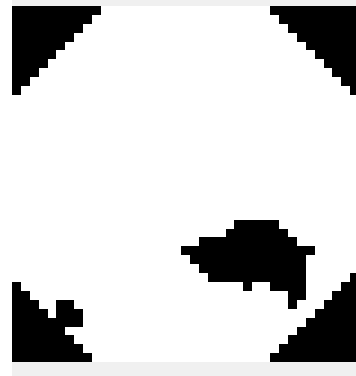
Cuenta	Calibre (mm)	Ratio Maduro	Ratio Semi-Maduro	Ratio No Maduro
10	37,53	49,96%	39,45%	10,60%

Detecciones erróneas

El filtro del modelo también recoge algunos datos erróneos.



Cuenta	Calibre (mm)	Ratio Maduro	Ratio Semi-Maduro	Ratio No Maduro
3	31,16	38,86%	58,48%	2,66%



Cuenta	Calibre (mm)	Ratio Maduro	Ratio Semi-Maduro	Ratio No Maduro
1	19,38	14,62%	80,22%	5,16%

Finalmente, el sistema nos muestra el resultado definitivo del análisis.

Visualización de resultados del modelo.

Número de frutos detectados: 32

Calibre Medio: 30.57 mm

Estado de madurez en los frutos:

MADUROS	SEMI-MADUROS	NO MADUROS
-----	-----	-----
38.32%	43.88%	17.80%

Vista global de resultados

A medida que se realizan ensayos con imágenes diferentes, se observan variaciones en el error de la detección.

Como se ha podido ver en el apartado anterior, en ocasiones no se distingue bien entre una región perteneciente al suelo respecto a un fruto. El sistema comete el error al disponer de índices matiz y brillos muy similares.

Por ello, se ha procedido a la realización de un análisis de las regiones detectadas para definir la calidad de los resultados obtenidos.

Se muestra el resultado de la sensibilidad y de la especificidad del modelo. La sensibilidad nos indica la capacidad de acertar en la detección de los casos positivos. La especificidad, sin embargo, nos indica la capacidad de detectar correctamente los casos negativos de la muestra [12].



SENSIBILIDAD		ESPECIFICIDAD	
CORRECTA	INCORRECTA	CORRECTA	INCORRECTA
99,9%	0,1%	100%	0%

En esta imagen, la calidad de los resultados es óptima. Por lo que sería perfectamente válida.



SENSIBILIDAD		ESPECIFICIDAD	
CORRECTA	INCORRECTA	CORRECTA	INCORRECTA
45,1%	54,9%	20%	80%

Este caso sería el otro extremo de la calidad de resultado.

Una observación detectada es que, la mayoría de los errores debidos a la sensibilidad del sistema se concentran en falsos positivos en regiones del suelo.



En este caso, realiza una detección de 8 frutos (posiblemente maduros). De forma que ayuda a estos errores de precisión detectados.

Mejora con recorte de la imagen

Durante el desarrollo del proyecto, el análisis y las pruebas se han ido realizando sobre la misma planta en el mismo invernadero.

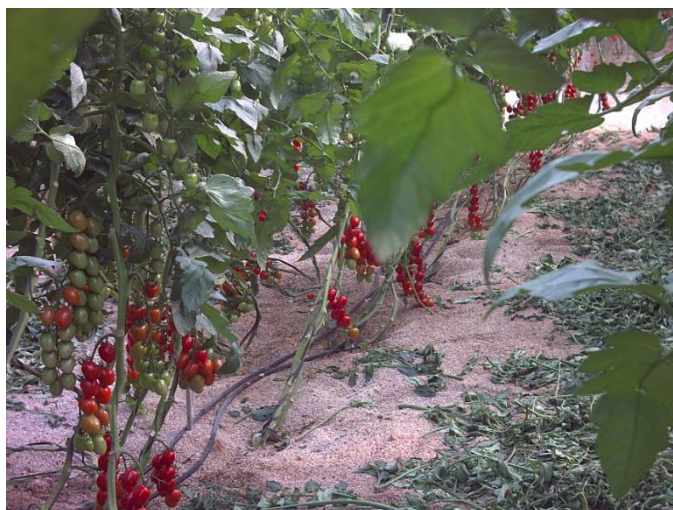


Imagen 25: Vista total de la cámara.

Como se puede apreciar, además de los frutos localizados en la planta más cercana (izquierda de la imagen), también se muestran los de las plantas del fondo. De forma que se obtienen los siguientes resultados:

```
Visualización de resultados del modelo.  
Número de frutos detectados: 163  
Calibre Medio: 30.07 mm  
Estado de madurez en los frutos:  
MADUROS      SEMI-MADUROS  NO MADUROS  
-----  
36.71%      34.87%       28.41%
```

Tabla 5: Resultados del modelo con la imagen total.

Si recortamos la imagen, se puede observar la variación de la medida:



Imagen 26: Vista recortada centrada en la planta a analizar.

Visualización de resultados del modelo.
Número de frutos detectados: 101
Calibre Medio: 32.65 mm
Estado de madurez en los frutos:

MADUROS	SEMI-MADUROS	NO MADUROS
-----	-----	-----
26.88%	32.30%	40.82%

Tabla 6: Resultados obtenidos del modelo con la vista recortada.

Ensayos no llevados al proyecto

En este apartado, se muestran ensayos que, por los resultados obtenidos, no se han empleado en el proyecto. Se muestran con el fin de conocer las herramientas y abrir la posibilidad de, en futuras mejoras, poder aplicarlos.

En ocasiones, no obtener los resultados esperados es cuestión de dedicarle más tiempo y conocimiento.

Detector de bordes

Aplicando esta herramienta sobre la imagen se puede apreciar la gran cantidad de bordes que detecta. Esto se debe a la falta de homogeneidad en la textura del contorno que rodea a los frutos.



Imagen 27: Detector de bordes con el método Canny.



Imagen 28: Detector de bordes con el método Sobel.



Imagen 29: Detector de bordes con el método Roberts.



Imagen 30: Detector de bordes con el método Prewitt.

Se consigue una gran cantidad de regiones que habría que analizar una a una y ello sobrecargaría el tiempo de computación del modelo. De forma que se ha preferido buscar otras vías que realicen un filtrado más intenso antes de proceder a la aplicación de este tipo de técnicas.

También se ha empleado el método de detección Canny con los siguientes resultados:



Imagen 31: Aplicación del método Canny en imagen centrada en rama.

La gran cantidad de ruido imposibilita su detección.

A continuación, se realiza la prueba en una rama delimitada obteniendo el siguiente resultado.

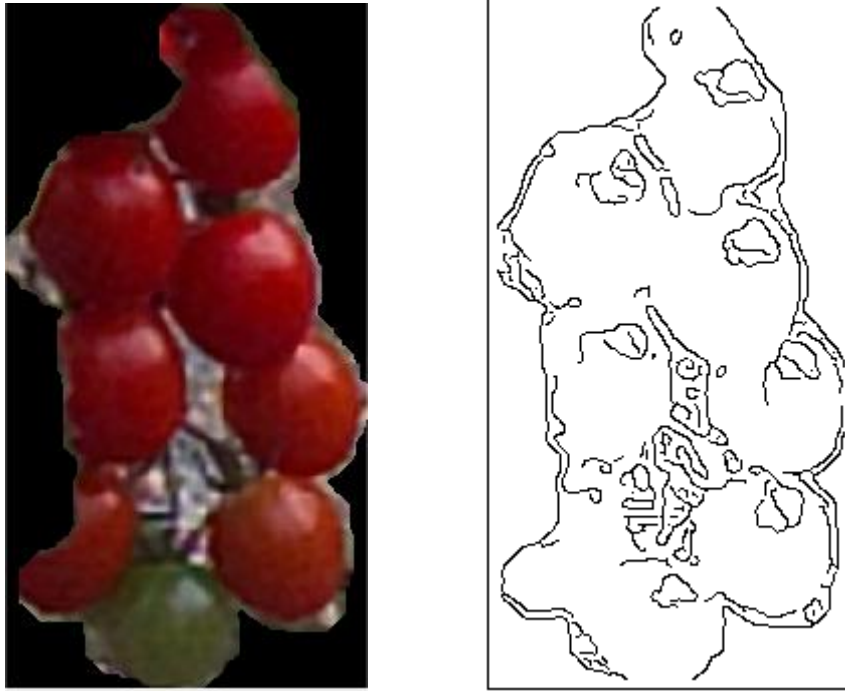


Imagen 32: Aplicación del método Canny en imagen centrada en rama (con máscara de región).

Se ha eliminado gran parte del ruido, pero aún permanece parte en la imagen. Por otra parte, también existe ruido interno del fruto, como el reflejo de luz, de forma que esta técnica no resulta del todo fiable para trabajar con ella en este entorno.

RCNN

Una técnica muy aplicada para la detección de objetos en imágenes es la de regiones con convolución de redes neuronales (R-CNN).

Una red convolucional (CNN) es uno de los algoritmos más populares para Deep Learning en el que un modelo aprende a realizar tareas de clasificación directamente a partir de imágenes, videos, textos o sonidos.

Son muy útiles para el reconocimiento de objetos en imágenes a partir de patrones localizados en ellas. Además, tiene la capacidad de autoaprendizaje a partir de patrones facilitados, eliminando la necesidad de extraer manualmente sus características.

Esta técnica es ampliamente utilizada en aplicaciones que involucran reconocimiento de objetos y visión artificial. Algunos de los ejemplos en los que se aplica serían en los vehículos de conducción autónoma y aplicaciones de reconocimiento facial [9].

Un valor añadido de este tipo de herramientas se trata de la capacidad de aprender características de la imagen de forma automática, eliminando su extracción manual, a partir de una arquitectura de capas. Esta arquitectura dota al sistema de capacidad de auto extraer esas características útiles para el reconocimiento de objetos. Aplican filtros de entrenamiento con distintas resoluciones, características simples como el brillo y el contraste y otras más complejas que definen inequívocamente el objeto.

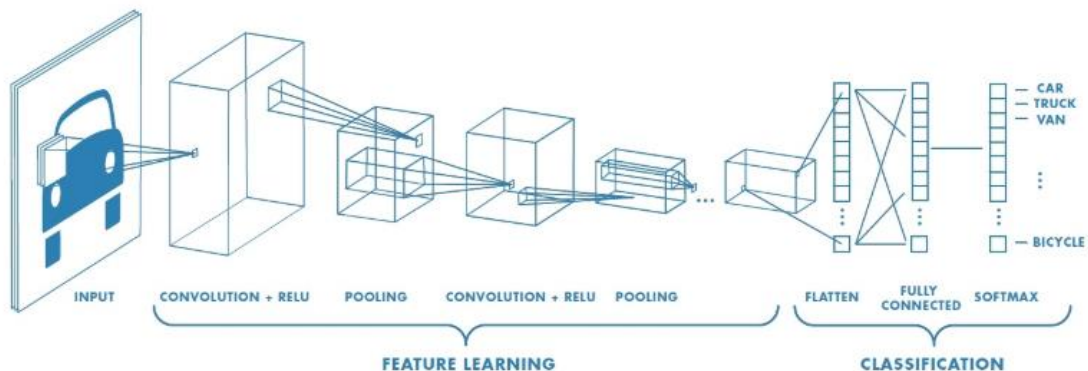


Figura 36: Representación de arquitectura de capas para la clasificación de un modelo RCNN.

Originalmente, esta herramienta requería de un arquitecto que definiera las capas y los filtros de la herramienta y una gran cantidad de muestras (millones) para proceder a entrenar el modelo. Eso hace que en proyectos sin tanta información sea muy complicado su desarrollo. Una vez generado, se le va alimentando con más muestras con el fin de ir perfeccionándolo.

En la actualidad, se disponen de numerosos modelos ya generados y alimentados con millones de muestras. Estas permiten generar otros con un número muy inferior

(centenares) de imágenes con el fin de recoger aquellas características que diferencian estas del resto ya entrenadas. Algunos de los modelos más conocidos son GoogLeNet, AlexNet, vgg16 y vgg19 así como otros procedentes de Caffe y TensorFlow-Keras [9].

Para el entrenamiento empleando la técnica R-CNN, se han recogido algunas imágenes disponibles y se ha etiquetado todos aquellos frutos que queremos que el sistema detecte.



Imagen 33: Visualización de la selección manual de regiones para someterlas a entrenamiento del modelo RCNN.

Posteriormente, se ha procedido a su entrenamiento a partir del modelo AlexNet. Este modelo dispone de un entrenamiento con 1.2 millones de imágenes extraídas de ImageNet [10] y una estructura de 23 capas con capacidad de clasificar imágenes en 1000 categorías de objetos.

Las imágenes que se introducen en el entrenamiento del modelo deben disponer de un tamaño de 227x227 (ya que el entrenamiento original es en ese tamaño). De forma que se reduce el tamaño original (3664x2748) a este.



Imagen 34: Vista de la imagen, redimensionada, que entra en el proceso de clasificación de un modelo RCNN.

Y, finalmente, se procede a su entrenamiento.

```

Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Mini-batch | Mini-batch | RPN Mini-batch | RPN Mini-batch | Base Learning |
|       |          | (hh:mm:ss)  | Loss       | Accuracy   | RMSE       | Accuracy       | RMSE           | Rate          |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1     | 1       | 00:00:20   | 2.5927    | 68.05%    |             | 45.63%        | 1.48           | 0.0010       |
| 10    | 30      | 00:08:46   | 0.4768    | 100.00%   |             | 93.12%        | 0.44           | 0.0010       |
=====
Detector training complete.

```

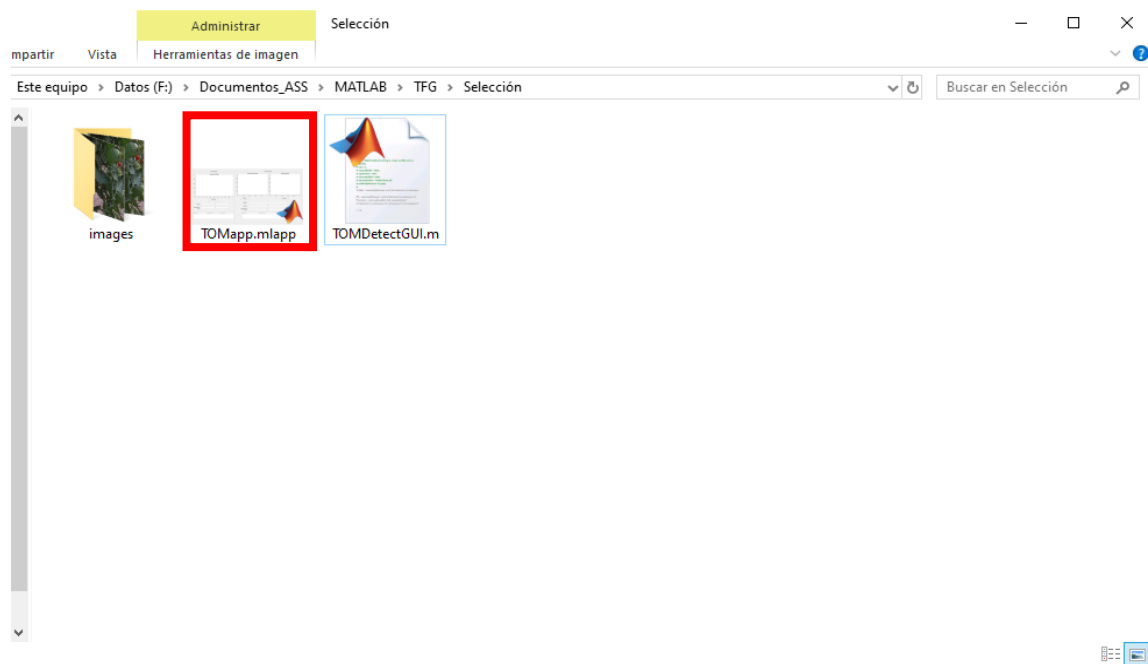
Después de todo este proceso, el sistema no es capaz de detectar ningún fruto, de forma que se desestima aplicar esta técnica para este proyecto.

INTERFAZ DE USUARIO

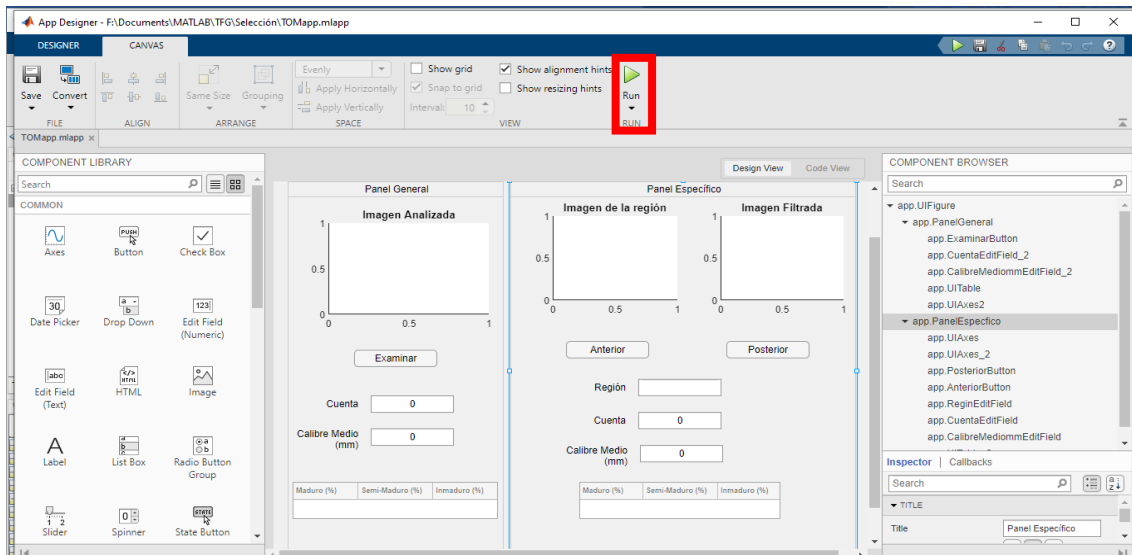
Con el fin de facilitar la interacción y realizar pruebas para el análisis del modelo, se ha desarrollado una aplicación que permite cargar imágenes y, después de su correspondiente tratamiento, visualizar los resultados obtenidos. También se ha incluido un panel secundario en el que poder analizar cada región detectada con la correspondiente salida del filtro que realiza la segmentación definitiva para determinar los frutos localizados en dicha región.

En el directorio donde se ha descargado el programa, se aloja el código con el algoritmo (TOMDetectGUI.m), un archivo con la interfaz de usuario (TOMapp.mlapp) y una carpeta con algunas imágenes de prueba (images).

A continuación, se procede a ejecutar el programa (TOMapp.mlapp) en MATLAB.



Al ejecutarse MATLAB, presentará una vista como la que se muestra a continuación.



Se procederá a ejecutarlo, directamente, pulsando el botón “RUN”.

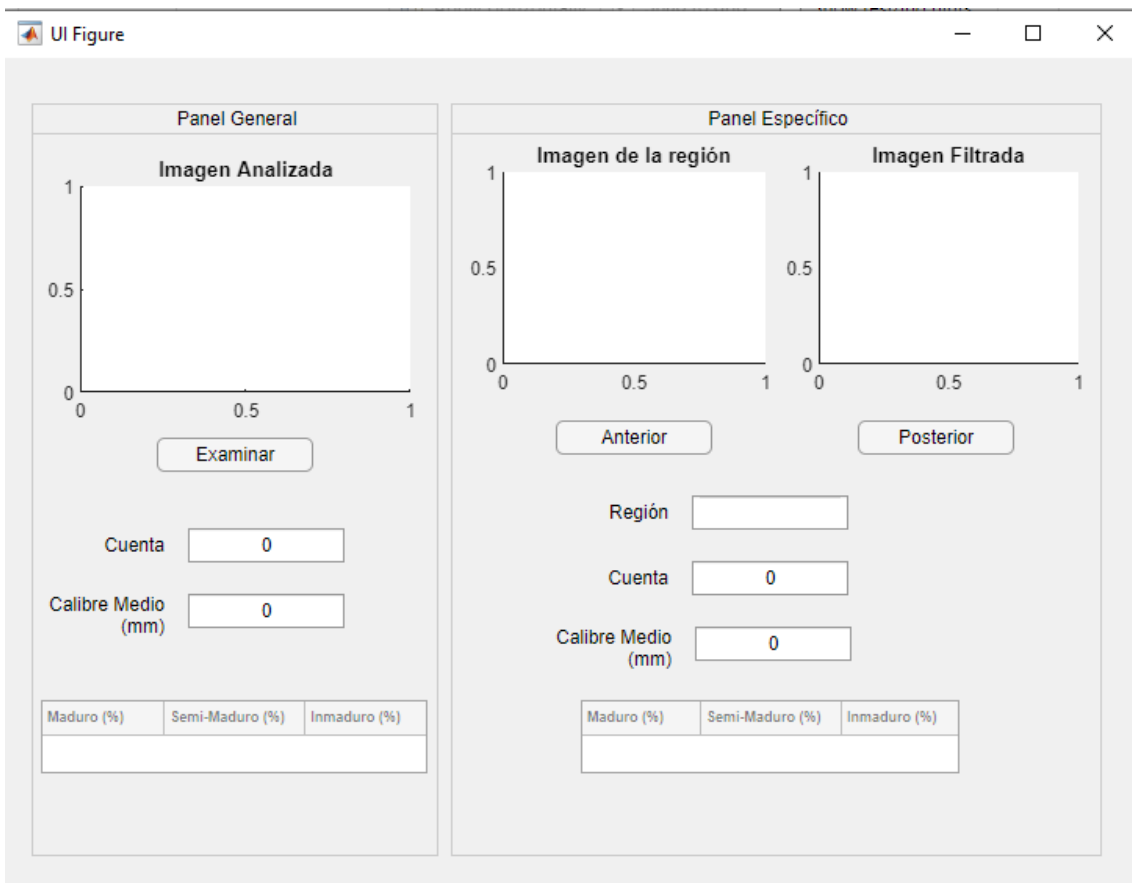


Figura 37: Visualización de la aplicación inicializada.

Esta aplicación está compuesta por dos paneles: Panel general y panel específico.

El panel general dispone de un botón (Examinar) que acciona otra ventana para la selección del fichero a analizar por el modelo. Una vez seleccionado el fichero, la aplicación ejecuta el correspondiente algoritmo, que incluye el modelo, y visualiza los resultados.

La cuenta total de los frutos detectados se visualiza en el campo llamado "Cuenta". También se visualiza el calibre medio de la plantación (Calibre Medio) y la proporción de frutos Maduros, Semi-Maduros y No Maduros detectados.

En el panel específico se visualiza la región en la que se han detectado frutos y la salida del filtro que determina los puntos en la región que han determinado esa decisión tomada por el modelo. A través de los botones "Anterior" y "Posterior", se puede visualizar cada región detectada con su correspondiente filtro.

En el campo denominado "Región" se visualiza el índice de región visualizada respecto del total detectado. En el campo "Cuenta" se indica el número de frutos detectados en la región. En el campo "Calibre Medio" se indica el calibre medio de todos los frutos detectados en la región. Finalmente, en los campos "Maduro", "Semi-Maduro" y "No Maduro", se indica la proporción de frutos detectados que se encuentran en el correspondiente estado de maduración.

A medida que se avanza en la selección de regiones, en la imagen situada en el panel general se dibuja un recuadro que especifica la localización de dicha región visualizada.



Figura 38: Visualización de la aplicación con imagen cargada.

DISCUSIÓN Y CONCLUSIONES

El entrenamiento del modelo se ha realizado empleando las imágenes capturadas durante el primer mes de cultivo (octubre). En este periodo, el grado de acierto es bastante elevado. Al realizar la prueba de detección en meses posteriores, se ha detectado un incremento elevado (cercano al 50%) de errores en el modelo.

Cabe mencionar que, en la agricultura, la imagen de las plantas depende, en gran escala, de la intención del agricultor por mantener las plantas en un estado concreto. Esto quiere decir, si durante un periodo se ha mantenido con muchas hojas y, posteriormente, se realiza una limpieza rigurosa, es de esperar que la imagen de la planta varíe considerablemente (influyendo en el modelo).

No se han realizado pruebas en otros invernaderos, de forma que no se puede certificar la inalterabilidad del modelo.

En una primera instancia, es posible que se necesite un modelo por invernadero hasta que se pueda encontrar un patrón común de todos ellos. Habría que tener en cuenta que la planta se siembre en el mismo lugar.

También se ha detectado, cuando se procedió a deshojar las plantas, que las plantas que se encuentran en otras líneas también se visualizan en la imagen.

Cuando el dispositivo se queda bajo de batería, no envía imágenes.

El periodo de muestreo es diario. Esto restringe la disponibilidad de datos de evolución del fruto a un dato diario (disminuyendo su resolución).

Conclusiones

- Es necesario un modelo por cultivo.
- Se podría contemplar hacer un modelo por cada estado de la planta (de no poder hacerse un deshojado desde el principio). Si no, habría que empezar a ejecutar el modelo solamente durante el periodo que se pudiera deshojar. Es importante que la planta mantenga una estética similar.
- Sería conveniente emplear una planta que no tenga ninguna otra detrás que pudiera aparecer en el campo de visión (una planta de una línea situada en el perímetro).
- El dispositivo empleado para la captura y envío de imágenes debería garantizar energía de captura con baja energía y almacenamiento hasta de recursos suficientes (energía o cobertura).
- Se debería poder disponer de datos en periodos horarios (no solo diarios).

Con estos aspectos tenidos en cuenta, se podría dar por validado el modelo.

LINEAS FUTURAS

Detección por localización

- Convertir el sistema fijo en móvil.

En este proyecto, se toma una planta como modelo del resto. Idealmente, debería poderse rastrear todas las plantas del cultivo, identificarlas, geolocalizarlas y analizarlas.

Mejora la detección de hojas.

- Es necesario hacer un modelo de detección de hojas.

Como se ha podido observar, es difícil distinguir entre ciertos frutos y las hojas (al compartir numerosos patrones de color). Sería recomendable desarrollar un modelo de detección de hojas que se pudiera aplicar en la mayoría de los cultivos ornamentales. Se podría identificar un patrón común entre las hojas de los diferentes cultivos. Dada la gran variabilidad geométrica (y cromática) entre frutos de diferentes naturalezas, se estima que sea más recomendable analizar las hojas al no tener tanta como los frutos.

PLANIFICACIÓN TEMPORAL Y PRESUPUESTO

Planificación temporal

09/09/2019 – 20/12/2019: Formación

01/12/2020 – 01/03/2020: Análisis inicial

01/03/2020 – 19/04/2020: Primer Modelado

20/04/2020 – 30/04/2020: Prueba RCNN

01/05/2020 – 01/06/2020: Modelado definitivo

04/05/2020 – 17/06/2020: Memoria

Presupuesto

Los datos de rangos salariales se han obtenido en un estudio realizado por la empresa Page Personal [13], teniendo en cuenta el perfil desarrollador como ingeniero de proyectos sin experiencia. Con estos datos, se obtiene una banda salarial de 25000€ anuales brutos. Contando un trabajo de 8 horas en cada día laborable, habiendo 253 días laborables y 23 de vacaciones en el 2020, el trabajo del ingeniero tiene un coste de **13,6€/hora**.

Recursos Humanos			
Concepto	€/ud	Ud	Total
Formación	13,6€	60	816€
Análisis inicial	13,6€	16	217,6€
Modelo inicial	13,6€	56	761.6€
Ensayo RCNN	13,6€	18	244,8€
Modelo definitivo	13,6€	20	272€
Memoria	13,6€	50	680€
TOTAL	81,6€	220	2992€

Tabla 7: Costes de recursos humanos en el proyecto.

Materiales					
Medio material	Valor total	Vida estimada	Valor diario	Días	Total
Estación de cámaras	1769€	5 años	0,484€	282	136,49€
Licencia Matlab	800€	1 año	0,22€	282	62,04€
TOTAL	2569€		0,704	564	198,53€

Tabla 8: Costes materiales en el proyecto.

Valoración total	
Concepto	Subtotal
Recursos Humanos	2992€
Recursos Materiales	198.53€
Base Imponible	3190,53€
IVA (21%)	670,01€
TOTAL	3860,54€

Tabla 9: Costes totales del proyecto.

ANEXOS

Código de aplicación para interfaz de usuario MATLAB

```
classdef TOMapp < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)

        UIFigure                matlab.ui.Figure
        PanelGeneral            matlab.ui.container.Panel
        ExaminarButton          matlab.ui.control.Button
        CuentaEditField_2Label  matlab.ui.control.Label
        CuentaEditField_2       matlab.ui.control.NumericEditField
        CalibreMediomLabel     matlab.ui.control.Label
        CalibreMediomEditField_2 matlab.ui.control.NumericEditField
        UITable                 matlab.ui.control.Table
        UIAxes2                 matlab.ui.control.UIAxes
        PanelEspecifico         matlab.ui.container.Panel
        UIAxes                  matlab.ui.control.UIAxes
        UIAxes_2               matlab.ui.control.UIAxes
        PosteriorButton         matlab.ui.control.Button
        AnteriorButton          matlab.ui.control.Button
        ReginEditFieldLabel     matlab.ui.control.Label
        ReginEditField          matlab.ui.control.EditField
        CuentaEditFieldLabel    matlab.ui.control.Label
        CuentaEditField         matlab.ui.control.NumericEditField
        CalibreMediomLabel_2    matlab.ui.control.Label
        CalibreMediomEditField  matlab.ui.control.NumericEditField
        UITable_2              matlab.ui.control.Table
    end
end
```

```
properties (Access = private)
```

```
nArrSum = 0;  
meinGagemmAvg = 0;  
maduroAvg = 0;  
semiAvg = 0;  
noMaduroAvg = 0;  
nArr = 0;  
meinGagemmArr = 0;  
maduroArr = 0;  
semiArr = 0;  
noMaduroArr = 0;  
IcropCell = 0;  
IcropFilterCell = 0;  
boundinBoxCell = 0;  
  
index = 1;  
I = 0;
```

```
end
```

```
methods (Access = private)
```

```
function img = TOMDetect(app, imageFile)  
  
    img=imread(imageFile);  
  
    imgHsv = rgb2hsv(img);  
  
    imgHSVHUE = imgHsv(:,:,1);  
  
    imgHSVSAT = imgHsv(:,:,2);  
  
    imgHSVVAL = imgHsv(:,:,3);
```

```
end
```

```
end
```



```

% Callbacks that handle component events

methods (Access = private)

% Button pushed function: ExaminarButton
function ExaminarButtonPushed(app, event)
    [FileName,FilePath]=uigetfile('*.png');
    %imageFile = [FilePath FileName];
    imageFile = fullfile(FilePath, FileName);

%
    TOMDetect (app,imageFile);

TOMDetect = TOMDetectGUI(imageFile);

app.nArrSum = TOMDetect{1};
app.meinGagemmAvg = TOMDetect{2};
app.maduroAvg = TOMDetect{3};
app.semiAvg = TOMDetect{4};
app.noMaduroAvg = TOMDetect{5};
app.nArr = TOMDetect{6};
app.meinGagemmArr = TOMDetect{7};
app.maduroArr = TOMDetect{8};
app.semiArr = TOMDetect{9};
app.noMaduroArr = TOMDetect{10};
app.IcropCell = TOMDetect{11};
app.IcropFilterCell = TOMDetect{12};
app.boundinBoxCell = TOMDetect{13};
app.index = 1;

app.I = imread(imageFile);
img
insertShape (app.I, 'Rectangle', app.boundinBoxCell{app.index}, 'LineWidth
',10);
imshow(img, 'parent', app.UIAxes2);

```

```

        app.CuentaEditField_2.Value = app.nArrSum;

        app.CalibreMediommmEditField_2.Value = app.meinGagemmAvg;

        app.UITable.Data =
[app.maduroAvg, app.semiAvg, app.noMaduroAvg];

        app.ReginEditField.Value =
strcat(int2str(app.index), '/', int2str(length(app.nArr)));

        imshow(app.IcropCell{app.index}, 'parent', app.UIAxes);

imshow(app.IcropFilterCell{app.index}, 'parent', app.UIAxes_2);

        app.CuentaEditField.Value = app.nArr(app.index);

        app.CalibreMediommmEditField.Value =
app.meinGagemmArr(app.index);

        app.UITable_2.Data =
[app.maduroArr(app.index), app.semiArr(app.index), app.noMaduroArr(app.i
ndex)];

end

% Button pushed function: AnteriorButton
function AnteriorButtonPushed(app, event)

    if app.index <= 1

        app.index = 1;

    else

        app.index = app.index - 1;

    end

    img =
insertShape(app.I, 'Rectangle', app.boundinBoxCell{app.index}, 'LineWidth
', 10);

    imshow(img, 'parent', app.UIAxes2);

    app.ReginEditField.Value =
strcat(int2str(app.index), '/', int2str(length(app.nArr)));

    imshow(app.IcropCell{app.index}, 'parent', app.UIAxes);

imshow(app.IcropFilterCell{app.index}, 'parent', app.UIAxes_2);

        app.CuentaEditField.Value = app.nArr(app.index);

        app.CalibreMediommmEditField.Value =
app.meinGagemmArr(app.index);

```

```

        app.UITable_2.Data =
[app.maduroArr(app.index), app.semiArr(app.index), app.noMaduroArr(app.i
ndex)];

    end

% Button pushed function: PosteriorButton
function PosteriorButtonPushed(app, event)
    if app.index >= length(app.nArr);
        app.index = length(app.nArr);
    else
        app.index = app.index + 1;
    end

    img =
insertShape(app.I, 'Rectangle', app.boundinBoxCell{app.index}, 'LineWidth
',10);

    imshow(img, 'parent', app.UIAxes2);

    app.ReginEditField.Value =
strcat(int2str(app.index), '/', int2str(length(app.nArr)));

    imshow(app.IcropCell{app.index}, 'parent', app.UIAxes);

imshow(app.IcropFilterCell{app.index}, 'parent', app.UIAxes_2);

    app.CuentaEditField.Value = app.nArr(app.index);

    app.CalibreMediommmEditField.Value =
app.meinGagemmArr(app.index);

    app.UITable_2.Data =
[app.maduroArr(app.index), app.semiArr(app.index), app.noMaduroArr(app.i
ndex)];

    end

% Callback function
function Button_4Pushed(app, event)
    [nArrSum, meinGagemmAvg, maduroAvg, semiAvg, noMaduroAvg,
...
nArr, meinGagemmArr, maduroArr, semiArr, noMaduroArr, ...
IcropArr, IcropFilterErodeArr, boundinBoxArr] =
TOMdetectGUI(imageFile)

    end

```

```

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 726 532];
        app.UIFigure.Name = 'UI Figure';

        % Create PanelGeneral
        app.PanelGeneral = uipanel(app.UIFigure);
        app.PanelGeneral.TitlePosition = 'centertop';
        app.PanelGeneral.Title = 'Panel General';
        app.PanelGeneral.Position = [18 22 260 482];

        % Create ExaminarButton
        app.ExaminarButton = uibutton(app.PanelGeneral, 'push');
        app.ExaminarButton.ButtonPushedFcn = createCallbackFcn(app,
@ExaminarButtonPushed, true);
        app.ExaminarButton.Position = [80 246 100 22];
        app.ExaminarButton.Text = 'Examinar';

        % Create CuentaEditField_2Label
        app.CuentaEditField_2Label = uilabel(app.PanelGeneral);
        app.CuentaEditField_2Label.HorizontalAlignment = 'right';
        app.CuentaEditField_2Label.Position = [41 188 44 22];
        app.CuentaEditField_2Label.Text = 'Cuenta';

        % Create CuentaEditField_2

```

```

app.CuentaEditField_2 = uieditfield(app.PanelGeneral,
'numeric');
app.CuentaEditField_2.HorizontalAlignment = 'center';
app.CuentaEditField_2.Position = [100 188 100 22];

% Create CalibreMediommLabel
app.CalibreMediommLabel = uilabel(app.PanelGeneral);
app.CalibreMediommLabel.HorizontalAlignment = 'right';
app.CalibreMediommLabel.Position = [5 140 80 28];
app.CalibreMediommLabel.Text = {'Calibre Medio'; '(mm)'};

% Create CalibreMediommEditField_2
app.CalibreMediommEditField_2 =
uieditfield(app.PanelGeneral, 'numeric');
app.CalibreMediommEditField_2.HorizontalAlignment =
'center';
app.CalibreMediommEditField_2.Position = [100 146 100 22];

% Create UITable
app.UITable = uitable(app.PanelGeneral);
app.UITable.ColumnName = {'Maduro (%)'; 'Semi-Maduro (%)';
'Inmaduro (%)'};
app.UITable.RowName = {};
app.UITable.Position = [6 53 247 47];

% Create UIAxes2
app.UIAxes2 = uiaxes(app.PanelGeneral);
title(app.UIAxes2, 'Imagen Analizada')
xlabel(app.UIAxes2, '')
ylabel(app.UIAxes2, '')
app.UIAxes2.Position = [10 278 240 171];

% Create PanelEspecfico
app.PanelEspecfico = uipanel(app.UIFigure);
app.PanelEspecfico.TitlePosition = 'centertop';

```

```

app.PanelEspecfico.Title = 'Panel Específico';
app.PanelEspecfico.Position = [286 22 416 482];

% Create UIAxes
app.UIAxes = uiaxes(app.PanelEspecfico);
title(app.UIAxes, 'Imagen de la región')
xlabel(app.UIAxes, '')
ylabel(app.UIAxes, '')
app.UIAxes.Position = [12 296 194 162];

% Create UIAxes_2
app.UIAxes_2 = uiaxes(app.PanelEspecfico);
title(app.UIAxes_2, 'Imagen Filtrada')
xlabel(app.UIAxes_2, '')
ylabel(app.UIAxes_2, '')
app.UIAxes_2.Position = [214 296 192 162];

% Create PosteriorButton
app.PosteriorButton = uibutton(app.PanelEspecfico, 'push');
app.PosteriorButton.ButtonPushedFcn =
createCallbackFcn(app, @PosteriorButtonPushed, true);
app.PosteriorButton.Position = [260 257 100 22];
app.PosteriorButton.Text = 'Posterior';

% Create AnteriorButton
app.AnteriorButton = uibutton(app.PanelEspecfico, 'push');
app.AnteriorButton.ButtonPushedFcn = createCallbackFcn(app,
@AnteriorButtonPushed, true);
app.AnteriorButton.Position = [67 257 100 22];
app.AnteriorButton.Text = 'Anterior';

% Create RegInEditFieldLabel
app.RegInEditFieldLabel = uilabel(app.PanelEspecfico);
app.RegInEditFieldLabel.HorizontalAlignment = 'right';

```

```

app.ReginEditFieldLabel.Position = [95 209 44 22];
app.ReginEditFieldLabel.Text = 'Región';

% Create ReginEditField
app.ReginEditField = uieditfield(app.PanelEspecfico,
'text');

app.ReginEditField.HorizontalAlignment = 'center';
app.ReginEditField.Position = [154 209 100 22];

% Create CuentaEditFieldLabel
app.CuentaEditFieldLabel = uilabel(app.PanelEspecfico);
app.CuentaEditFieldLabel.HorizontalAlignment = 'right';
app.CuentaEditFieldLabel.Position = [95 167 44 22];
app.CuentaEditFieldLabel.Text = 'Cuenta';

% Create CuentaEditField
app.CuentaEditField = uieditfield(app.PanelEspecfico,
'numeric');

app.CuentaEditField.HorizontalAlignment = 'center';
app.CuentaEditField.Position = [154 167 100 22];

% Create CalibreMediommLabel_2
app.CalibreMediommLabel_2 = uilabel(app.PanelEspecfico);
app.CalibreMediommLabel_2.HorizontalAlignment = 'right';
app.CalibreMediommLabel_2.Position = [61 119 80 28];
app.CalibreMediommLabel_2.Text = {'Calibre Medio'; '(mm)'};

% Create CalibreMediommEditField
app.CalibreMediommEditField =
uieditfield(app.PanelEspecfico, 'numeric');

app.CalibreMediommEditField.HorizontalAlignment = 'center';
app.CalibreMediommEditField.Position = [156 125 100 22];

% Create UITable_2

```

```

        app.UITable_2 = uitable(app.PanelEspecfico);
        app.UITable_2.ColumnName = {'Maduro (%)'; 'Semi-Maduro (%)';
        'Inmaduro (%)'};
        app.UITable_2.RowName = {};
        app.UITable_2.Position = [83 53 242 47];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = TOMapp

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 0
            clear app
        end
    end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end

```


end
end

Algoritmo MATLAB

```
function res = TOMDetectGUI(imageFile)
% clear all;
debug=false;

img=imread(imageFile);
imgHsv = rgb2hsv(img);
imgHSVHUE = imgHsv(:,:,1);
imgHSVSAT = imgHsv(:,:,2);
imgHSVVAL = imgHsv(:,:,3);

if debug
    figure(1)
    imshow(img)
end

%%
%Filtro de tomates maduros
imgTOMMad = imbinarize(imgHSVHUE,0.95) | (imcomplement(imbinarize(imgHSVHUE,0.20)));
imgTOMMad = imgTOMMad & imbinarize(imgHSVSAT,0.15);
imgTOMMad = imgTOMMad & imbinarize(imgHSVVAL,0.2);
%Limpiamos los pixeles dispersos
imgTOMMadFill = imfill(imgTOMMad,'holes');
mask = 'disk';
se = strel(mask, 20);
imgTOMMadClean = imerode(imgTOMMadFill,se);
imgTOMMadClean = imdilate(imgTOMMadClean,se);

if debug
    figure(2)
    subplot(1,2,1)
    imshow(imgTOMMadFill)
    subplot(1,2,2)
    imshow(imgTOMMadClean)
end
```

```

end

%%
%Analizamos cada region
s = regionprops(imgTOMMadClean, 'BoundingBox', 'FilledImage', 'Image',
'centroid', 'area');

%%
originalWidthpx = 3664;
originalHeightpx = 2748;
distancem = 3.6;
Hsvmm = 6.6; %H sensor vertical en mm
Hshmm = 8.8; %H sensor horizontal en mm
fmm = 12;
alphahrad = 2*atan(Hshmm/2/fmm);
Hohmm = distancem*tan(alphahrad/2)*1e3; %H objeto horizontal en mm
scalepxmm = 2*Hohmm/originalWidthpx; %escala en imagen mm/px
zoom = 1.5;

IcropCell = {};
IcropFilterCell = {};
boundinBoxCell = {};
maduroArr = [];
semiArr = [];
noMaduroArr = [];
meinGagemmArr = [];
nArr = [];

output = [];
outputId = [];
for i = 1:length(s)
%       ss = s(96);
    ss = s(i);
    Icrop = imresize(iscrop(img, ss.BoundingBox), size(ss.FilledImage))
.* uint8(ss.FilledImage);

```

```

%Detector de frutos

IcropHsv = rgb2hsv(Icrop);

IcropFilter =
imbinarize(IcropHsv(:,:,1),0.95)&imbinarize(IcropHsv(:,:,2),0.3)&imbin
arize(IcropHsv(:,:,3),0.45);

IcropFilter =
(imcomplement(imbinarize(IcropHsv(:,:,1),0.2))&imcomplement(imbinarize
(IcropHsv(:,:,2),0.35))&imbinarize(IcropHsv(:,:,3),0.3));

if debug
    figure(3)
    subplot(1,2,1)
    imshow(Icrop)
    subplot(1,2,2)
    imshow(IcropFilter)
end

mask = 'disk';
se = strel(mask, 4);
IcropFilterDilate = imdilate(IcropFilter,se);

IcropFilterFill = imfill(IcropFilterDilate,'holes');

%Aplicamos erosion para eliminar elementos indeseados
se = strel(mask, 10);
IcropFilterErode = imerode(IcropFilterFill,se);

[IcropImage, IcropCounter] = bwlabel(IcropFilterErode);

if debug
    figure(4)
    subplot(2,2,1)
    imshow(IcropFilter)
    subplot(2,2,2)
    imshow(IcropFilterDilate)
    subplot(2,2,3)

```

```

    imshow(IcropFilterFill)
    subplot(2,2,4)
    imshow(IcropFilterErode)
end

if IcropCounter > 0
    IcropCell{end+1} = Icrop;
    IcropFilterCell{end+1} = IcropFilterErode;
    %Estado de madurez
    zoneArea =
    nnz(((imcomplement(imbinarize(IcropHsv(:,:,1),0.2))|imbinarize(IcropHsv(:,:,1),0.95))&IcropHsv(:,:,3)>0)==1);

    maduro =
    nnz((imbinarize(IcropHsv(:,:,1),0.95)&IcropHsv(:,:,3)>0)==1)/zoneArea;

    semi =
    nnz((imcomplement(imbinarize(IcropHsv(:,:,1),0.1))&IcropHsv(:,:,3)>0)==1)/zoneArea;

    noMaduro =
    nnz(((imcomplement(imbinarize(IcropHsv(:,:,1),0.2))&imbinarize(IcropHsv(:,:,1),0.1))&IcropHsv(:,:,3)>0)==1)/zoneArea;

    boundinBoxCell{end+1} = ss.BoundingBox;

    maduroArr = [maduroArr maduro*100];
    semiArr = [semiArr semi*100];
    noMaduroArr = [noMaduroArr noMaduro*100];

    % A continuación, se procede a deducir el calibre de los frutos.

    meinAreapx = zoneArea/IcropCounter;
    meinGagemm = 2*sqrt(meinAreapx/pi)*scalepxmm/zoom;

    nArr = [nArr IcropCounter];
    meinGagemmArr = [meinGagemmArr meinGagemm];
end
end

```

```

nArrSum=0;
meinGagemmAvg=0;
maduroAvg=0;
semiAvg=0;
noMaduroAvg=0;
if length(maduroArr) > 0
    maduroArrn = nArr.*maduroArr;
    semiArrn = nArr.*semiArr;
    noMaduroArrn = nArr.*noMaduroArr;

    nArrSum = sum(nArr);

    maduroAvg = (sum(maduroArrn)/nArrSum);
    semiAvg = (sum(semiArrn)/nArrSum);
    noMaduroAvg = (sum(noMaduroArrn)/nArrSum);

    meinGagemmAvg = sum(meinGagemmArr)/length(meinGagemmArr);
end

res = {nArrSum, meinGagemmAvg, maduroAvg, semiAvg, noMaduroAvg, ...
    nArr, meinGagemmArr, maduroArr, semiArr, noMaduroArr, ...
    IcropCell, IcropFilterCell, boundinBoxCell};

%%
fprintf Resutls

%   fprintf('Visualización de resultados del modelo.\n')
%   fprintf('Número de frutos detectados: %d\n',nArrSum)
%   fprintf('Calibre Medio: %.2f mm\n',meinGagemmAvg)
%   fprintf('Estado de madurez en los frutos:\n')
%   fprintf('MADUROS\t\tSEMI-MADUROS\tNO MADUROS\n')
%   fprintf('-----\t\t-----\t-----\n')
%
fprintf('%.2f%%\t\t%.2f%%\t\t%.2f%%\n',maduroAvg*100,semiAvg*100,noM
aduroAvg*100)

```

Tabla de bandas salariales

Salario anual en euros													
NACIONAL													
CIFRA DE NEGOCIO <20M€				CIFRA DE NEGOCIO 20 - 50M€				CIFRA DE NEGOCIO >50M€					
CALIDAD													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
Técnico de Calidad	18 - 24K	26 - 36K	28 - 38K	-	23 - 28K	26 - 36K	35 - 40K	-	23 - 33K	35 - 40K	40 - 45K	-	↔
Director / Responsable de Calidad	30 - 38K	35 - 42K	40 - 45K	-	30 - 40K	40 - 45K	45 - 55K	10%	38 - 48K	50 - 60K	60 - 70K	10%	↗
MANTENIMIENTO													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
Electromecánico	20 - 25K	25 - 30K	30 - 33K	-	20 - 25K	28 - 35K	33 - 38K	-	20 - 25K	28 - 35K	30 - 40K	-	↗
Director / Resp. de Mantenimiento	35 - 40K	32 - 40K	35 - 42K	-	35 - 40K	40 - 45K	50 - 60K	10%	45 - 50K	45 - 55K	50 - 60K	10%	↔
SAT (Servicio Atención Técnico)	25 - 30K	30 - 38K	35 - 43K	-	25 - 30K	33 - 42K	38 - 48K	10%	25 - 35K	35 - 40K	40 - 50K	10%	↔
INGENIERÍA / OFICINA TÉCNICA													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
Director Técnico / Ingeniería	35 - 42K	45 - 55K	50 - 60K	10%	42 - 52K	55 - 65K	60 - 70K	10%	50 - 60K	70 - 80K	80 - 90K	15%	↔
Project Manager	25 - 30K	28 - 38K	35 - 45K	-	25 - 30K	35 - 42K	40 - 50K	10%	25 - 30K	40 - 50K	50 - 60K	15%	↗
Proyectista	25 - 30K	28 - 35K	34 - 38K	-	25 - 30K	32 - 40K	35 - 42K	-	25 - 30K	32 - 40K	35 - 42K	-	↗
I + D													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
Resp. de I+D	25 - 32K	30 - 38K	40 - 50K	-	30 - 40K	35 - 45K	55 - 65K	10%	40 - 50K	50 - 60K	65 - 75K	15%	↔
PRODUCCIÓN													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
Jefe / Resp. de Producción	27 - 35K	30 - 38K	30 - 40K	-	30 - 40K	38 - 45K	40 - 50K	10%	40 - 50K	45 - 50K	55 - 65K	15%	↔
Ing. de Procesos / Mejora Continua	25 - 30K	30 - 35K	35 - 45K	-	28 - 38K	35 - 40K	37 - 47K	-	35 - 40K	38 - 48K	45 - 55K	10%	↗
Plant Manager	40 - 50K	55 - 65K	60 - 70K	10%	50 - 60K	70 - 80K	75 - 85K	10%	55 - 65K	90 - 100K	100 - 110K	15%	↗
SEGURIDAD Y MEDIO AMBIENTE (HSE)													
POSICIONES	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	<5 años	6 - 10 años	+11 años	Bonus	Tendencia
HSE Manager	20 - 30K	25 - 33K	30 - 37K	-	25 - 35K	35 - 40K	37 - 47K	-	30 - 40K	40 - 50K	60 - 70K	10%	↔

BIBLIOGRAFÍA

[1]: Recomendaciones de estado para recolección y postcosecha (apartados 10 y 11) del tomate, definidos según el portal Infoagro:

<http://www.infoagro.com/hortalizas/tomate3.htm>

[2]: La visión artificial en la agricultura de precisión:

<https://www.lanner-america.com/es/blog-es/la-ia-y-la-vision-artificial-redefinen-la-agricultura-de-precision/>

[3]: Caso de éxito en sistema de reconocimiento facial para CAMPOSOL:

<https://www.hispatec.es/reconocimiento-facial-en-agricultura/>

[4]: Proyecto de sistema de reconocimiento de frutas:

<http://opera.eii.us.es/pid/public/uploads/pid/entregables/2016-2017/G2016-2017-2/Sistemadereconocimientodefrutas.pdf>

[5]: Relación del ángulo de visión con el tamaño del sensor:

<http://fotoigual.com/8-calculo-del-angulo-vision-funcion-la-distancia-focal-tamano-del-sensor/>

[6]: Definición de color:

<https://es.wikipedia.org/wiki/Color>

[7]: Proceso de convolución que permite aplicar una máscara a una imagen:

https://en.wikipedia.org/wiki/Kernel_image_processing

[8]: Tabla de los tamaños de los sensores en las cámaras:

https://es.wikipedia.org/wiki/Formato_del_sensor_de_imagen

[9]: Explicación de Redes Neuronales Convolucionales ofrecida por MATHWORKS:

<https://es.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>

[10]: Principal repositorio de imágenes origen de los modelos mayoritariamente aplicados:

<http://image-net.org/index>

[11]: Explicación de esperanza matemática:

https://es.wikipedia.org/wiki/Esperanza_matem%C3%A1tica

[12]: Definición de sensibilidad y especificidad:

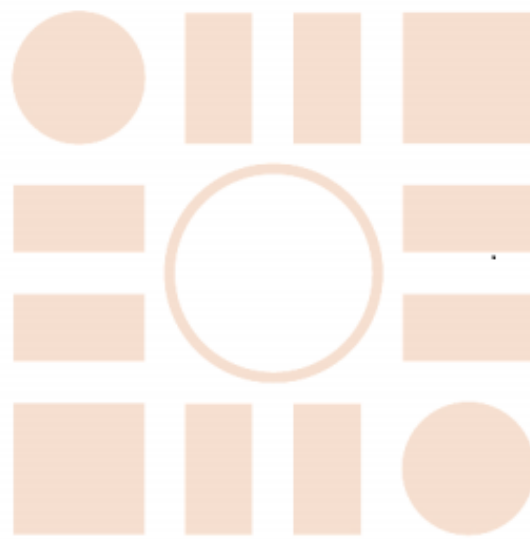
[https://es.wikipedia.org/wiki/Sensibilidad_y_especificidad_\(estad%C3%ADstica\)](https://es.wikipedia.org/wiki/Sensibilidad_y_especificidad_(estad%C3%ADstica))

}

[13]: Consulta de estudios de remuneración ofrecida por Page Personnel:

<https://www.pagepersonnel.es/prensa-estudios/estudios/estudios-de-remuneracion>

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR

