

DOCUMENTACIÓN PROYECTO

G R U P O V C - 6

2024-2025

MARTA GORRAIZ
IRENE GONZÁLEZ
IKER MARCELO

ÍNDICE

1.	INTRODUCCIÓN	2
2.	OBJETIVOS DEL PROYECTO	3
3.	DISEÑO DEL PROYECTO	3
4.	DATASET UTILIZADO	4
5.	ETAPAS DEL PROCESO	4
5.1	CAPTURA DE IMÁGENES	4
5.2	PROCESAMIENTO DE IMÁGENES	5
5.3	EXTRACCIÓN DE CARACTERÍSTICA	6
6.	ALMACENAMIENTO	8
6.1	EXTRUCTURA DEL DICCIONARIO	9
6.2	PROCESO DE ALMACENAMIENTO	10
7.	IMPLEMENTACIÓN	11
8.	BIBLIOGRAFIA	13

1. INTRODUCCIÓN

Este proyecto tiene como objetivo el desarrollo de un sistema para la detección en tiempo real de piezas de frutas, con el propósito de identificar tanto el tipo de fruta como su estado (bueno o malo).

Para ello, se emplean diversas técnicas avanzadas de procesamiento de imágenes, como el filtrado de colores, la detección de contornos y el análisis de características geométricas y de texturas. Estas técnicas permiten obtener una representación precisa de las frutas en tiempo real, generando la información necesaria para que un modelo de aprendizaje automático pueda clasificar correctamente la fruta y evaluar su estado. Posteriormente, en función de esta clasificación, un brazo robótico se encargará de colocar la fruta en la categoría que le corresponda.

En esta memoria se tratará específicamente del proceso de visión por computador, que constituye la primera fase del proyecto y desempeña un papel fundamental en su desarrollo. Este sistema es responsable de capturar imágenes de las frutas y realizar un análisis detallado para extraer una representación precisa que sirva como entrada al modelo de clasificación. La implementación de estas técnicas de visión por computador asegura la calidad de los datos utilizados en las etapas posteriores, siendo la base de este proyecto.

2. OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto es desarrollar un sistema eficiente de visión por computadora para la detección en tiempo real de frutas, que permita identificar tanto el tipo de fruta como su estado (bueno o malo). Los objetivos específicos para la parte de visión por computadora son los siguientes:

1. **Captura de imágenes en tiempo real:** Implementar un sistema que capture imágenes en tiempo real de las frutas mediante una cámara.
2. **Procesamiento de imágenes:** Aplicar técnicas avanzadas de procesamiento de imágenes, como el ajuste de brillo y contraste, filtrado de colores, data augmentation, suavizado, detección de bordes y contornos, para mejorar la calidad de las imágenes y facilitar la identificación de las frutas.
3. **Extracción de características geométricas:** Extraer características geométricas clave de las frutas que permiten describir la forma de las frutas de manera precisa.
4. **Análisis de características de color y textura:** Analizar las características cromáticas para proporcionar una descripción detallada de las frutas y su estado.
5. **Almacenamiento de los datos:** Almacenar de manera estructurada todas las características en un archivo de formato CSV.

3. DISEÑO DEL PROYECTO

En este apartado se describe el diseño del proyecto y cómo se ha estructurado para llevarlo a cabo. A continuación, se presenta una imagen que muestra el diseño general del sistema.

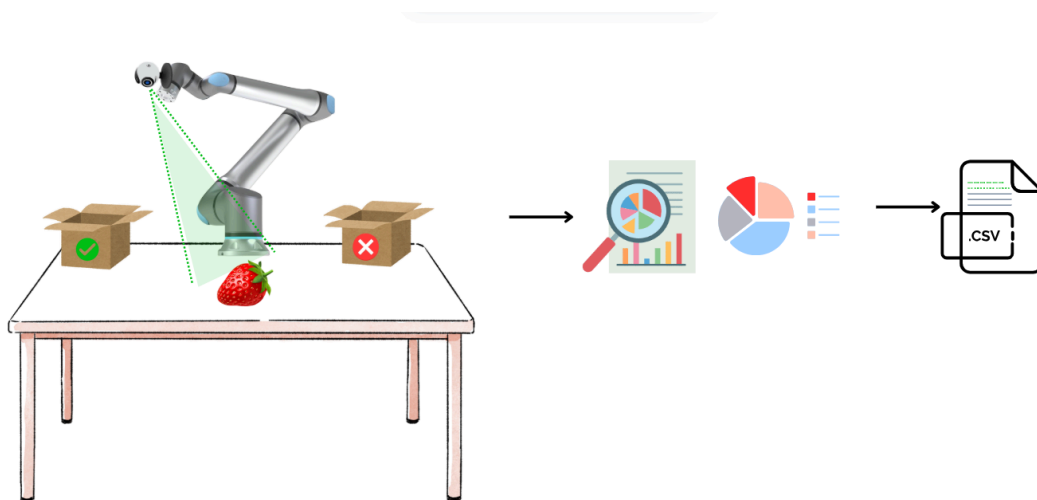


Ilustración 1: Diseño del proyecto

Se coloca la fruta sobre una superficie plana. A continuación, se utiliza una cámara para detectar la fruta, y, al presionar una tecla, se captura una imagen. Una vez tomada la fotografía, la imagen se procesa utilizando técnicas de visión por computadora para extraer las características relevantes de la fruta. Estas características incluyen datos geométricos, de color y de textura, que se almacenan en un archivo CSV para su posterior análisis y clasificación.

4. DATASET UTILIZADO

El dataset utilizado en este proyecto ha sido creado manualmente, realizando fotos de diferentes frutas. Las imágenes fueron tomadas a mano y organizadas en carpetas correspondientes a las diferentes categorías de frutas, con el fin de facilitar su procesamiento y clasificación. Con el objetivo de aumentar el tamaño del dataset, se ha aplicado data augmentation para obtener más imágenes con el brillo reducido y aumentado.

El dataset está organizado de la siguiente manera:

- **plátano**
 - *plátano_bien*
 - *plátano_defecto* estado.
- **limón**
 - *limón_bien*
 - *limón_defecto*
- **fresa**
 - *fresa_bien*
 - *fresa_defecto*
- **uva**
 - *uva_bien*
 - *uva_defecto*
- **pera**
 - *pera_bien*
 - *pera_defecto*

5. ETAPAS DEL PROCESO

En este apartado se va a tratar las 3 etapas principales: captura de imágenes, procesamiento de imágenes y extracción de características. A continuación, se describe cada una de ellas:

5.1 CAPTURA DE IMÁGENES

La primera etapa del proyecto consiste en capturar las imágenes de las frutas en tiempo real. Para ello, se emplea una cámara web conectada al sistema, que permite obtener imágenes de manera continua.

La captura de imágenes se gestiona a través de dos bibliotecas: OpenCV y NumPy. Ambas herramientas permiten un manejo eficiente de las imágenes. OpenCV es la biblioteca principal utilizada para acceder a la cámara y obtener las imágenes. Mediante la función `cv2.VideoCapture()`, se logra la captura de imágenes en tiempo real, permitiendo que el sistema capture los frames necesarios para el análisis. Esta función asegura que la comunicación entre el sistema y la cámara sea fluida, optimizando la adquisición de imágenes para el siguiente paso en el flujo de trabajo.

Una vez capturadas las imágenes, se almacenan temporalmente y se preparan para ser procesadas en las siguientes etapas, en las cuales se realizarán el data augmentation y las transformaciones necesarias para la detección precisa de las frutas en diferentes estados.

5.2 PROCESAMIENTO DE IMÁGENES

El procesamiento de imágenes es una de las primeras etapas clave del proyecto, ya que transforma las imágenes capturadas en un formato adecuado para la extracción de características. Durante este proceso, se aplican varias técnicas con el objetivo de mejorar la calidad de la imagen y facilitar la detección del elemento principal (la fruta). Las técnicas principales utilizadas son las siguientes:

- **Brillo y Contraste**

Para mejorar la visibilidad de los detalles en la imagen, se ha implementado una función que ajusta el brillo y el contraste. El ajuste de brillo permite corregir imágenes que están demasiado oscuras o claras, mientras que el ajuste de contraste mejora la diferenciación entre la fruta y el fondo. Esto se logra mediante el uso de la función `cv2.convertScaleAbs`, que ajusta la intensidad de los píxeles en la imagen sin perder información importante. Este paso ayuda a resaltar las frutas y sus contornos, haciendo que sean más fáciles de detectar en los siguientes pasos.

- **Suavizado y filtrado**

Las imágenes originales suelen contener ruido, lo que puede afectar la calidad del análisis. Para reducir este ruido, se aplica un filtro Gaussiano con un kernel de 5x5, lo que suaviza la imagen. Este proceso ayuda a minimizar las variaciones aleatorias de brillo que no aportan información útil, suavizando los contornos de las frutas y facilitando la identificación de los bordes. Este paso es fundamental para mejorar la precisión de los métodos de detección de bordes.

- **Detección de contornos y bordes**

La detección de contornos se usa para identificar las frutas dentro de la imagen. En este proyecto, se utiliza el filtro Canny, un detector de bordes que destaca las transiciones de intensidad en la imagen, es decir, los cambios bruscos en los valores de los píxeles. Este proceso convierte la imagen en una representación binaria (0 o 1), donde los bordes detectados se muestran en blanco y el resto de la imagen en negro.

Una vez que los bordes han sido identificados, se aplica la función `cv2.findContours()`, que toma la imagen binaria generada por el filtro Canny para detectar las formas cerradas que corresponden a los objetos en la imagen. Los contornos permiten identificar de manera precisa los límites de las frutas.

Al finalizar esta etapa, las imágenes ya han sido procesadas para eliminar ruido y resaltar las características clave, lo que las hace que estén listas para la extracción de características en la siguiente etapa.

5.3 EXTRACCIÓN DE CARACTERÍSTICAS

En esta etapa, se realiza un análisis dividiendo el proceso en tres partes clave: geométricas, de color y de textura. A continuación, se detallan los procesos específicos para cada una de estas características:

○ Características Geométricas:

El primer paso consiste en calcular el área y el perímetro del contorno más grande identificado en la imagen, utilizando las funciones `cv2.contourArea` y `cv2.arcLength`. Estas métricas permiten cuantificar el tamaño y delinear la forma general del objeto.

Por otra parte, se obtienen características como la relación de aspecto, que es el cociente entre la anchura y la altura del objeto. Para calcular estas dimensiones, se utiliza el método `cv2.boundingRect`, que genera un rectángulo que encierra completamente el contorno. Este análisis también incluye la altura y anchura del rectángulo delimitador, proporcionando datos relevantes para diferenciar frutas de formas alargadas de las más redondeadas.

Otra característica importante es la circularidad, que mide qué tan cercana es la forma de la fruta a un círculo perfecto. Esta se calcula con la fórmula $(4 * \pi * \text{area}) / (\text{perimeter}^2)$. Un valor cercano a 1 indica una forma altamente circular, mientras que valores más bajos indican formas más irregulares. Este parámetro es particularmente útil para distinguir entre frutas de geometría esférica y aquellas con bordes más angulosos.

Por último, se extraen los Momentos Hu, un conjunto de siete características invariantes ante transformaciones geométricas como rotaciones, escalas y traslaciones. Estos momentos, calculados mediante la función `cv2.HuMoments`, son fundamentales para identificar frutas independientemente de su orientación o posición en la imagen. Además, los momentos Hu ofrecen una representación robusta de la forma, lo que resulta esencial en el análisis de patrones complejos.

○ **Características de color:**

Durante el análisis de los colores predominantes en las imágenes de frutas, se utiliza un enfoque basado en el espacio de color HSV (Hue, Saturation, Value), que facilita la separación de colores al ser menos sensible a las variaciones de brillo y saturación. El proceso comienza con la detección de contornos en la imagen, los cuales se perfeccionan mediante el cálculo de sus envolventes convexas utilizando la función `cv2.convexHull(cnt)`. Este paso asegura que los contornos sean cerrados y definidos, eliminando detalles irrelevantes o ruidos.

Con las envolventes convexas obtenidas, se construye una máscara que delimita claramente las regiones relevantes de la imagen, es decir, aquellas que corresponden a las frutas. La máscara se dibuja con `cv2.drawContours` y luego se aplica a la imagen original para aislar únicamente estas áreas de interés. Este aislamiento permite convertir las regiones seleccionadas al espacio de color HSV, optimizando la extracción de información cromática al centrarse únicamente en las áreas de interés.

En la siguiente imagen se muestra el proceso completo: desde la creación de la máscara hasta su aplicación a la imagen original. Además, se presentan los rangos de los cinco colores predominantes detectados en la fruta analizada, definidos a partir del histograma de tonos (canal H) del espacio de color HSV.



Ilustración 2: Aplicación de máscara (1)



Ilustración 3: Aplicación de máscara (2)

- **Características de textura:**

Gracias a estas características de texturas, es posible clasificar con mayor precisión el estado de una fruta. Este proceso se lleva a cabo mediante el cálculo de características texturales derivadas de la Matriz de Co-ocurrencia de Niveles de Gris (GLCM) y el análisis de la rugosidad superficial.

Para calcular las características de textura, la imagen se convierte a escala de grises, lo que simplifica el análisis de las intensidades de los píxeles. A través de la biblioteca mahotas, se genera la GLCM, que mide cómo las intensidades de los píxeles se relacionan espacialmente en la imagen. A partir de esta matriz, se extraen características texturales clave que ayudan a describir la superficie de la fruta. Estas métricas incluyen el contraste, que mide las diferencias de intensidad y refleja la prominencia de los bordes en la textura, y la correlación, que identifica patrones regulares en la distribución de los píxeles. Otras métricas, como la energía y la homogeneidad, evalúan la repetitividad y la uniformidad local de los patrones texturales, respectivamente.

Además, la rugosidad de la superficie de las frutas se estima calculando la varianza de la imagen en escala de grises. Este valor describe la dispersión de las intensidades, donde un valor alto sugiere superficies con mayor irregularidad o complejidad.

6. ALMACENAMIENTO

Una vez que las características se han extraído de cada imagen, el siguiente paso es almacenarlas de forma estructurada. Las características extraídas se organizan en un diccionario para cada imagen, el cual contiene la información sobre la geometría, color, textura y rugosidad de la fruta. A continuación, se muestra un listado de todas las características que se almacenan en el diccionario.

Características geométricas:

- Área
- Perímetro
- Relación de aspecto
- Circularidad
- Momentos Hu (Hu 1, Hu 2, Hu 3, Hu 4, Hu 5, Hu 6, Hu 7)

Características de color:

- Histograma de tonos (HSV)

- Rango de colores predominantes

Características de textura:

- Contraste
- Correlación
- Energía
- Homogeneidad
- Rugosidad

6.1 ESTRUCTURA DEL DICCIONARIO

Este diccionario se crea para cada imagen procesada, conteniendo todas las características necesarias para su posterior análisis. En las siguientes imágenes, se muestra el código utilizado para extraer estas características y cómo se organiza dicha estructura.

```
features = {
    'fruit_type': fruit_type,
    'category': category,
    'image_name': image_name,
    'rugosity': rugosity,
    'number_of_contours': number_of_contours,
    'area': shape_features['area'],
    'perimeter': shape_features['perimeter'],
    'aspect_ratio': shape_features['aspect_ratio'],
    'height': shape_features['height'],
    'width': shape_features['width'],
    'circularity': shape_features['circularity'],
    'mean_normalized_contour_x': shape_features['mean_normalized_contour_x'],
    'mean_normalized_contour_y': shape_features['mean_normalized_contour_y'],
    'std_normalized_contour_x': shape_features['std_normalized_contour_x'],
    'std_normalized_contour_y': shape_features['std_normalized_contour_y'],
    'contrast': contrast,
    'correlation': correlation,
    'energy': energy,
    'homogeneity': homogeneity
}

for i, hu_moment in enumerate(shape_features['hu_moments']):
    features[f'hu_moment_{i+1}'] = hu_moment

for i, h_range in enumerate(color_ranges):
    features[f'hue_range_{i+1}_start'] = h_range[0]
    features[f'hue_range_{i+1}_end'] = h_range[1]
```

Ilustración 4: Código del diccionario

Características generales:

- fruit_type
- category
- image_name
- rugosity
- number_of_contours

Características geométricas:

- area
- perimeter

- aspect_ratio
- height
- width
- circularity
- mean_normalized_contour_x
- mean_normalized_contour_y
- std_normalized_contour_x
- std_normalized_contour_y

Momentos Hu:

- hu_moment_1
- hu_moment_2
- hu_moment_3
- hu_moment_4
- hu_moment_5
- hu_moment_6
- hu_moment_7

Características de color:

- hue_range_1_start
- hue_range_1_end
- hue_range_2_start
- hue_range_2_end
- hue_range_3_start
- hue_range_3_end
- hue_range_4_start
- hue_range_4_end
- hue_range_5_start
- hue_range_5_end

Características de textura:

- contrast
- correlation
- energy
- homogeneity

6.2 PROCESO DE ALMACENAMIENTO

Una vez creado el diccionario para cada imagen, este se agrega a una lista llamada data. Esta lista acumula los diccionarios de todas las imágenes procesadas, creando una colección completa de características extraídas que facilita su posterior análisis.

Al finalizar la recopilación de todos los diccionarios en la lista, esta se convierte en un DataFrame utilizando la biblioteca pandas. El DataFrame organiza las características de manera tabular, donde cada fila corresponde a una imagen y cada columna a una característica específica. Esta estructura es ideal para realizar análisis más

avanzados, como filtrado, agrupamiento o la utilización de modelos de machine learning.

Finalmente, el DataFrame puede ser exportado a un archivo CSV utilizando el método `to_csv()`. Este archivo CSV facilita el almacenamiento de los datos de forma estructurada, permitiendo un acceso fácil y eficiente. En la siguiente imagen se puede apreciar una pequeña parte de esta estructura.

	fruit_type	category	image_name	rugosity	number_of_contours	area	perimeter	aspect_ratio	height	width
0	platano	platano_bien	IMG_20241029_130154.jpg	253.041027	30	220.5	1867.138293	1.883929	224	422
1	platano	platano_bien	IMG_20241029_130136.jpg	300.392883	26	112.0	618.666081	1.041176	170	177
2	platano	platano_bien	IMG_20241029_130112.jpg	296.440448	25	188.0	1898.846882	1.082840	338	366
3	platano	platano_bien	IMG_20241029_130101.jpg	286.637281	10	240.0	2246.944421	1.931174	247	477
4	platano	platano_bien	IMG_20241029_130048.jpg	342.070948	14	151.5	803.702663	1.129707	239	270
5	platano	platano_bien	IMG_20241029_130029.jpg	237.184075	18	328.0	1582.459143	1.249057	265	331
6	platano	platano_bien	IMG_20241029_130015.jpg	228.442043	14	50.0	261.705624	1.584615	65	103

Ilustración 5: Fichero csv

7. IMPLEMENTACIÓN

La parte de visión por computador ha sido fundamental en este proyecto y se ha implementado en dos etapas clave. Es importante destacar que el proceso de análisis de imágenes es idéntico en ambas etapas; la diferencia radica en la forma y el propósito de su uso. Las etapas con las siguientes:

o Extracción de características para el entrenamiento del modelo

En esta primera etapa, se utiliza un conjunto de imágenes previamente recopiladas y almacenadas en un dataset en Google Drive. Este dataset, creado manualmente, contiene imágenes etiquetadas que sirven como base para el entrenamiento de un modelo de aprendizaje automático.

El procesamiento de estas imágenes y la extracción de características se realiza en Google Colab, aprovechando su capacidad para ejecutar código en la nube y su integración con Google Drive. Colab facilita el acceso al dataset y proporciona un entorno interactivo y eficiente para ejecutar las tareas de análisis.

En este entorno, las imágenes se procesan una a una, extrayendo características relevantes como medidas geométricas, texturas, rugosidad, colores y momentos de Hu. Estas características se organizan en un archivo CSV que se almacena en Google Drive que será necesario para el entrenamiento de un modelo.

o Extracción de características en tiempo real

Esta etapa corresponde a la fase final del proyecto, en la que ya está desarrollado y en uso. En este punto, se centra en la captura de imágenes en tiempo real mediante una cámara conectada al sistema.

Para ello, se desarrolló un archivo independiente (proyecto.py) que utiliza el mismo código de procesamiento de imágenes empleado en la etapa de entrenamiento. Cuando se coloca una fruta frente a la cámara, el sistema captura una imagen al presionar una tecla, procesa la imagen, y extrae las mismas características que se calcularon en el dataset inicial. Estas características se almacenan temporalmente en un diccionario estructurado, el que posteriormente se convierte en un DataFrame, que se pasa directamente al modelo de clasificación para que realice su predicción en tiempo real.

Estos archivos se encuentran en el repositorio GitHub dentro de la carpeta Proyecto que se muestra a continuación: <https://github.com/IreeneGG/VisionComputador.git>

8. BIBLIOGRAFIA

- o *opencv-python*. (2024, 17 junio). PyPI. <https://pypi.org/project/opencv-python/>
- o *Mahotas: Computer Vision in Python — mahotas 1.4.16 documentation*. (s. f.). <https://mahotas.readthedocs.io/en/latest/>
- o Apuntes ofrecidos en Clase en la asignatura -*Visión por computador*
- o ChatGPT. (2024, 7 octubre). ChatGPT Español sin registro. <https://chatgpt.es/>