

IA DE RECONOCIMIENTO DE FRUTAS CON SU VALOR NUTRICIONAL

Estrellita Agreda Castillo¹, Valeria Huamanhuillca Liñan²

Ingeniería de Sistemas e Ingeniería Informática

Innovación y Desarrollo Tecnológico – 330

Joel López del Mar

Universidad Católica Sedes Sapientiae, Facultad de Ingeniería

Nota del Autor

Estrellita Agreda Castillo © <https://orcid.org/0009-0004-6921-2404>

Valeria Huamanhuillca Liñan © <https://orcid.org/0009-0001-4146-8879>

Resumen

Este informe presenta un estudio sobre el reconocimiento de 10 frutas comunes (palta, fresa, pera, mango, manzana, naranja, mandarina, plátano, piña y coco) mediante técnicas de procesamiento de imágenes y aprendizaje automático, también se utiliza el aprendizaje supervisado y no supervisado para obtener mayor precisión en los resultados.

Además, se incluyó información sobre el valor nutricional de cada fruta para resaltar su importancia en una dieta equilibrada y así poder mejorar la atención que dan los nutricionistas a sus pacientes.

Por último, la información nutricional presentada permite diseñar dietas equilibradas y prevenir deficiencias nutricionales.

Palabras clave: reconocimiento, frutas, valor nutricional, inteligencia artificial.

Abstract

This report presents a study on the recognition of 10 common fruits (avocado, strawberry, pear, mango, apple, orange, tangerine, banana, pineapple and coconut) using image processing and machine learning techniques, also using supervised learning and unsupervised to obtain more precision in the results.

In addition, information on the nutritional value of each fruit was included to highlight its importance in a balanced diet and thus be able to improve the care that nutritionists give to their patients.

Finally, the nutritional information presented makes it possible to design balanced diets and prevent nutritional deficiencies.

Keywords: recognition, fruits, nutritional value, artificial intelligence.

Introducción

El reconocimiento de frutas con su valor nutricional es un tema de investigación que ha cobrado importancia en los últimos años debido al creciente interés en promover hábitos saludables y prevenir enfermedades relacionadas con la nutrición.

El objeto de estudio de esta investigación es el desarrollo de un sistema de reconocimiento de frutas mediante técnicas de procesamiento de imágenes y aprendizaje automático, con el objetivo de identificar 10 frutas comunes y proporcionar información sobre su valor nutricional.

Las motivaciones de esta investigación son diversas, entre ellas, tenemos a la necesidad de fomentar el avance tecnológico por parte de los nutricionistas, asimismo promover en sus pacientes la ingesta de frutas como parte de una alimentación sana y balanceada y resaltar la importancia de conocer el valor nutricional de cada fruta para prevenir deficiencias nutricionales y/o adecuarlas a un plan alimenticio.

La relevancia del tema se encuentra en la importancia que tiene el consumo de frutas para una vida saludable, ya que son una fuente importante de vitaminas, minerales y otros nutrientes esenciales para el organismo.

En este estudio se recolectarán imágenes de alta calidad de las 10 frutas de interés y se recopilará información sobre su composición, incluyendo vitaminas, minerales y otros nutrientes.

La panorámica general del problema que motiva la investigación es la falta del uso de tecnología en consultorios nutricionales, lo que puede llevar a que sus pacientes tengan una ingesta insuficiente o desequilibrada de nutrientes esenciales, lo que puede tener consecuencias negativas para su salud.

Metodología

Sujetos:

En el estudio de reconocimiento de frutas con su valor nutricional, los sujetos de prueba serán las 10 frutas de interés: palta, fresa, pera, mango, manzana, naranja, mandarina, plátano, piña y coco. Las imágenes obtenidas de las frutas se utilizarán para entrenar y evaluar el modelo de reconocimiento de frutas.

Técnicas e Instrumentos:

La técnica principal utilizada en este estudio será el procesamiento de imágenes y la visión por computadora. Para ello, se utilizará Haar Cascade Trainer GUI para entrenar el modelo de reconocimiento de frutas. También se recolectarán datos sobre el valor nutricional de las frutas, como la composición y los nutrientes esenciales de las mismas. Para ello, se utilizarán datos confiables de nutrición.

Procedimiento:

El procedimiento consistirá en la toma de fotografías de alta calidad de las frutas de interés y la recopilación de información sobre su valor nutricional. Luego, se utilizará la herramienta Haar Cascade Trainer GUI para crear clasificadores de objetos, donde se utilizarán las imágenes tomadas para el entrenamiento que contendrán dos carpetas una de imágenes positivas y otra de imágenes negativas, esto se dará para cada una de las frutas.

Posteriormente, se implementará el uso de la librería de OpenCV, el cual es una biblioteca de visión por computadora de Python. Luego, se definirán los rangos de colores en formato HSV para cada fruta que se desea detectar, estos rangos de colores se utilizan para

filtrar los pixeles de la imagen que corresponde a la fruta específica, lo cual mejora la precisión del reconocimiento de frutas.

Código de imágenes de Frutas – Positivas

```

1  import cv2
2  import numpy as np
3  import imutils
4  import os
5  Datos = 'nombre de la fruta./p'
6  si no os . camino _ existe(Datos):
7      print ( 'Carpeta creada: ' ,Datos)
8      os _ makedirs(Datos)
9  gorra = cv2 . Captura de video( 1 )
10 x1, y1 = 190 , 80
11 x2, y2 = 450 , 398
12 cuenta = 0
13 while True :
14     ret, marco = gorra . read ()
15     si ret == Falso : romper
16     imAux = marco . Copiar()
17     cv2 . rectángulo(marco,(x1,y1),(x2,y2),( 255 , 0 , 0 ), 2 )
18     objeto = imAux[y1:y2,x1:x2]
19     #objeto = imutils.resize(objeto,width=38)
20     #print(objeto.shape)
21     k = cv2 . waitKey( 1 )
22     si k == orden ( 's' ):
23         cv2 . imwrite(Datos + '/objeto_{}.jpg' .
24 format(count),objeto)
25         print ( 'Imagen guardada:' + '/objeto_{}.jpg' .
26 format(count))
27         cuenta = cuenta + 1
28     si k == 27 :
29         romper
30     cv2 . imshow( 'marco' ,marco)
31     #cv2.imshow('objeto',objeto)
32 cap . liberar()
33 cv2 . destruirTodasLasVentanas()

```

Código de imágenes de Frutas – Negativas

```

1  import cv2
2  import numpy as np
3  import imutils
4  import os
5  Datos = 'granadilla./n'
6  si no os . camino _ existe(Datos):
7      print ( 'Carpeta creada: ' ,Datos)
8      os _ makedirs(Datos)
9  gorra = cv2 . Captura de video( 1 )
10
11  cuenta = 0
12  mientras que es verdadero :
13      ret, marco = gorra . leer ()
14      si ret == Falso : romper
15
16      k = cv2 . waitKey( 1 )
17      si k == orden ( 's' ):
18          cv2 . imwrite(Datos + '/objeto_{}.jpg' .
19  format(count),frame)
20          print ( 'Imagen guardada:' + '/objeto_{}.jpg' .
21  format(count))
22          cuenta = cuenta + 1
23      si k == 27 :
24          romper
25      cv2 . imshow( 'marco' ,marco)
26      #cv2.imshow('objeto',objeto)
27  cap . liberar()
28  cv2 . destruirTodasLasVentanas()

```

Uso de Cascade Trainer GUI

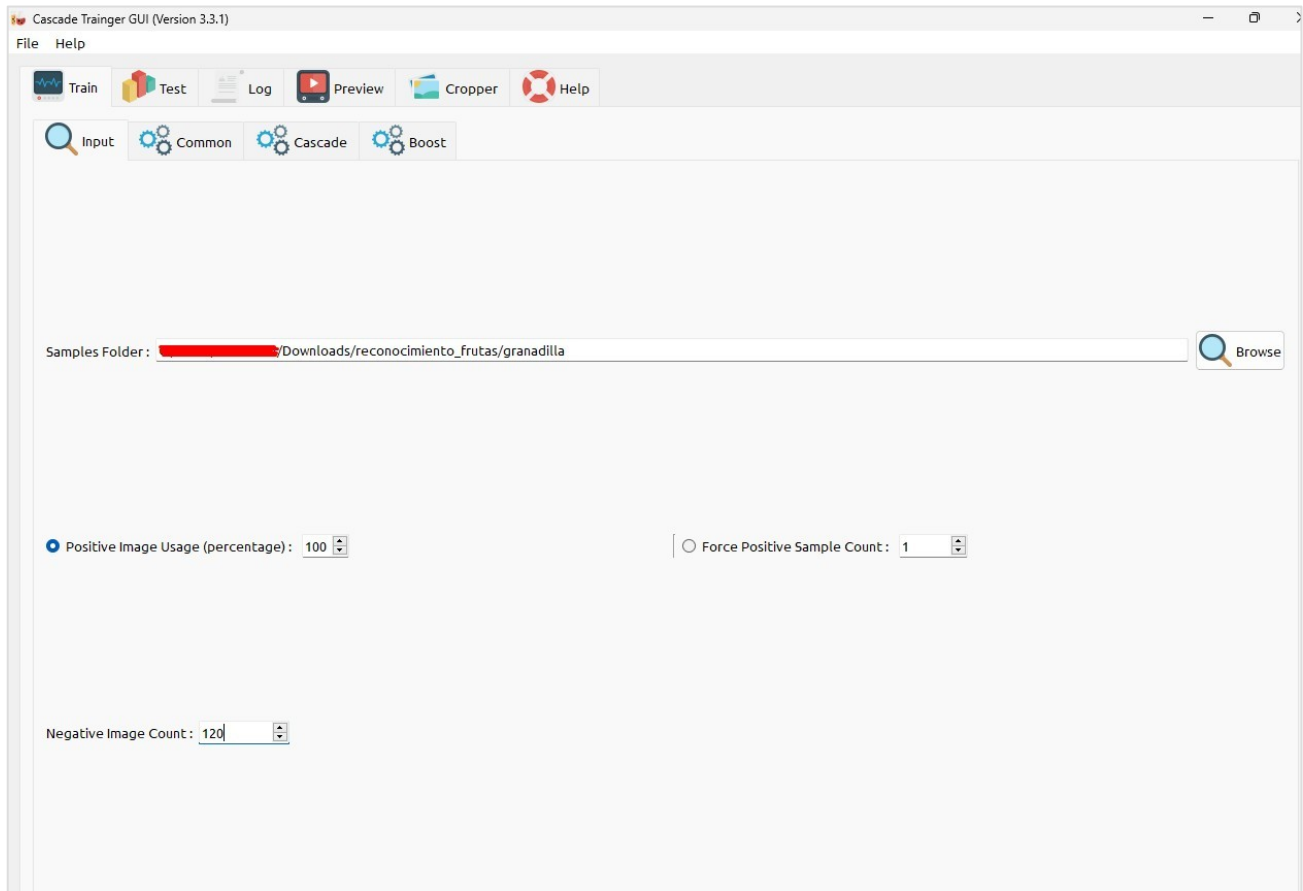


Ilustración 1: Elección de la carpeta donde se encuentran las imágenes

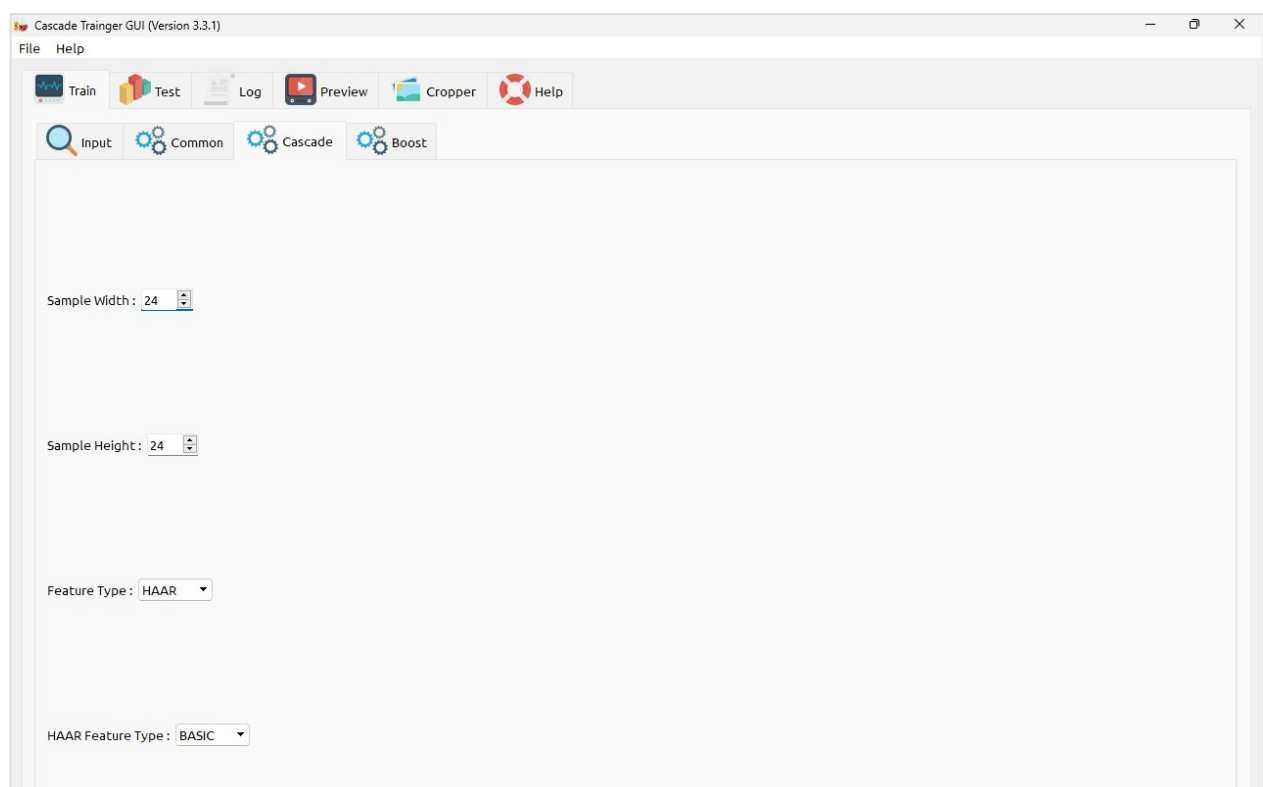


Ilustración 2: Ajustar según el tamaño de las imágenes

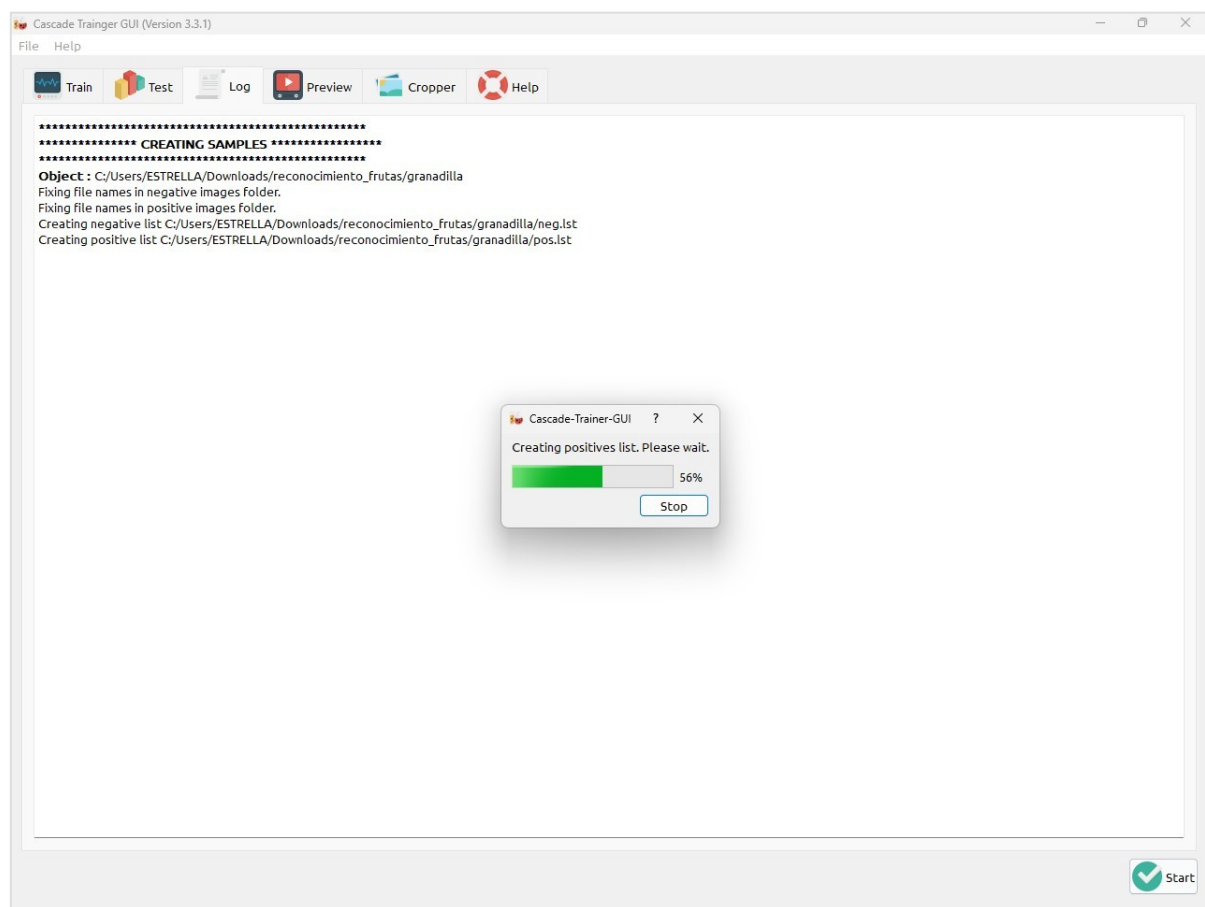


Ilustración 3: Click en "Start"

cargas > reconocimiento_frutas > granadilla > classifier

Nombre	Fecha de modificación	Tipo	Tamaño
▼ Hoy			
cascade.xml	03/07/2023 03:16 a. m.	xmlfile	8 KB
log.txt	03/07/2023 03:16 a. m.	Documento de te...	69 KB
stage9.xml	03/07/2023 03:16 a. m.	xmlfile	1 KB
stage8.xml	03/07/2023 03:06 a. m.	xmlfile	1 KB
stage7.xml	03/07/2023 03:03 a. m.	xmlfile	1 KB
stage6.xml	03/07/2023 03:02 a. m.	xmlfile	1 KB
stage5.xml	03/07/2023 03:01 a. m.	xmlfile	1 KB
stage4.xml	03/07/2023 03:01 a. m.	xmlfile	1 KB
stage3.xml	03/07/2023 03:01 a. m.	xmlfile	1 KB
stage2.xml	03/07/2023 03:00 a. m.	xmlfile	1 KB
stage1.xml	03/07/2023 03:00 a. m.	xmlfile	1 KB
params.xml	03/07/2023 03:00 a. m.	xmlfile	1 KB
stage0.xml	03/07/2023 03:00 a. m.	xmlfile	1 KB

Ilustración 4: Creación del Archivo .xml

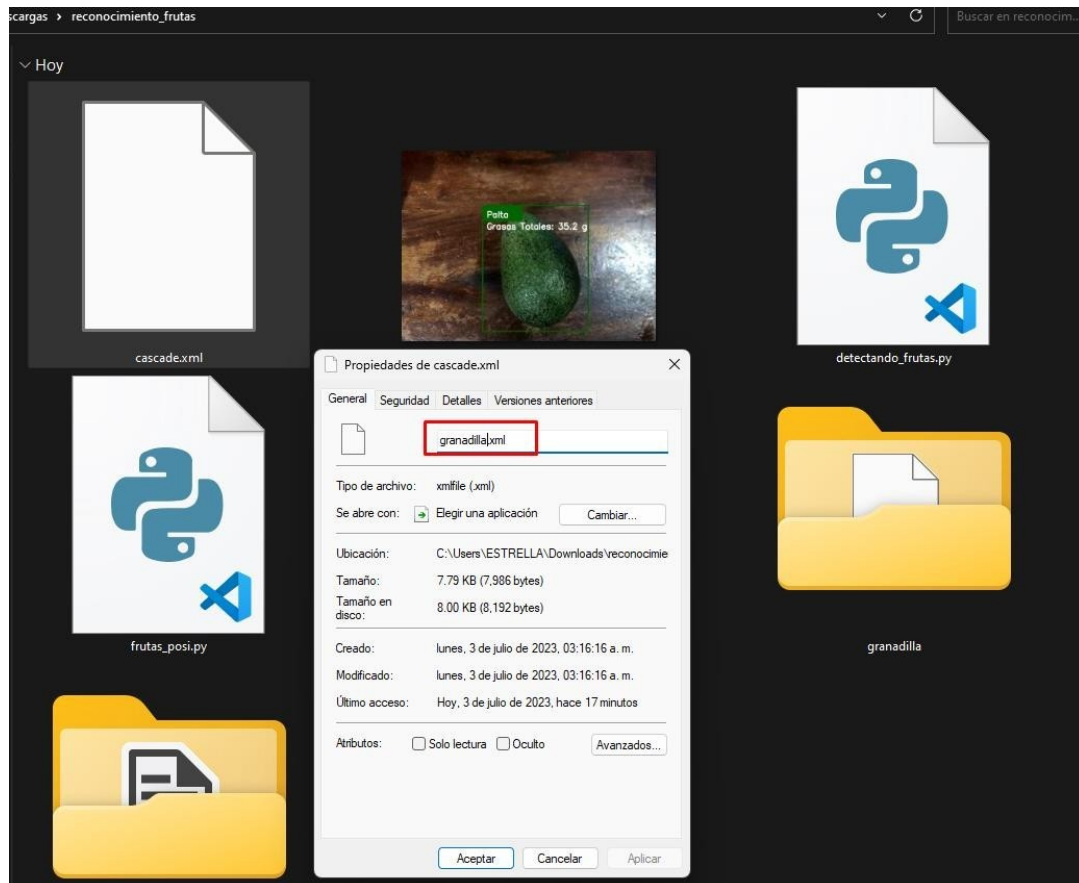


Ilustración 5: Cambiamos el nombre del Archivo y lo llevamos hacía la carpeta principal

Código Principal para el Reconocimiento de Frutas con su Valor Nutricional

```
import cv2

cap = cv2.VideoCapture(1)
#cap = cv2.VideoCapture('frutas.mp4') # Puedes usar un video en lugar de la cámara
font = cv2.FONT_HERSHEY_SIMPLEX

# Rangos de colores para diferentes frutas (en formato HSV)
frutas_colores = {
    'mandarina': [(0, 40, 50), (30, 255, 255)], # Rango de color para mandarinas
    'manzana': [(10, 80, 70), (20, 255, 255)], # Rango de color para manzanas
    'pera': [(35, 50, 50), (70, 255, 255)], # Rango de color para peras
    'platano': [(4, 50, 58), (15, 255, 255)], # Rango de color para plátanos
}

# Valores nutricionales de las frutas
frutas_nutricion = {
    'mandarina': 'Vitamina C: 53.2 mg',
    'manzana': 'Fibra: 4.4 g',
    'pera': 'Potasio: 206 mg',
    'platano': 'Vitamina B6: 0.43 mg',
    # Agrega más valores nutricionales según las frutas que detectes
}

# Valores nutricionales de las frutas
valor_valores = {
    'mandarina': 'Tamaño de la Porción: (1 porción)\nEnergía:242 kj / 58 kcal\nProteína: 0,88g\nGrasa: 0,34g\nGrasa Saturada: 0,043g\nGrasa Poliinsaturada: 0,071g\nGrasa Monoinsaturada: 0,065g\nColesterol: 0mg\nCarbohidratos: 14,54g\nFibra: 2g\nAzúcar: 11,53g\nSodio: 2mg\nPotasio: 181m',
    'manzana': 'Cantidad por (100 gramos) \nCalorías: 52\nGrasas totales: 0.2 g\nÁcidos grasos saturados: 0 g\nColesterol: 0 mg\nSodio: 1 mg\nPotasio: 107 mg\nCarbohidratos: 14 g\nFibra alimentaria: 2.4 g\nAzúcares: 10 g\nProteínas: 0.3 g\nVitamina C: 4.6 mg\nCalcio: 6 mg\nHierro: 0.1 mg\nVitamina D: 0 IU\nVitamina B6: 0 mg\nVitamina B12: 0 µg\nMagnesio: 5 mg',
    'pera': 'Cantidad por (100 gramos) \nCalorías: 57\nGrasas totales: 0.1 g\nÁcidos grasos saturados: 0 g\nColesterol: 0 mg\nSodio: 1 mg\nPotasio: 116 mg\nCarbohidratos: 15 g\nFibra alimentaria: 3.1 g\nAzúcares: 10 g\nProteínas: 0.4 g\nVitamina C: 4.3 mg\nCalcio: 9 mg\nHierro: 0.2 mg\nVitamina D: 0 IU\nVitamina B6: 0 mg\nVitamina B12: 0 µg\nMagnesio: 7 mg',
    'platano': 'Cantidad por (100 gramos)\nCalorías: 89\nGrasas totales: 0.3 g\nÁcidos grasos saturados: 0.1 g\nColesterol: 0 mg\nSodio: 1 mg\nPotasio: 358 mg\nCarbohidratos: 23 g\nFibra alimentaria: 2.6 g\nAzúcares: 17 g\nProteínas: 1.1 g\nVitamina C: 8.7 mg\nCalcio: 5 mg\nHierro: 0.3 mg\nVitamina D: 0 IU\nVitamina B6: 0.4 mg\nVitamina B12: 0 µg\nMagnesio: 27 mg',
    # Agrega más información nutricional para otras frutas según tus necesidades
}

# Cargar los clasificadores Haar Cascade para cada fruta
haar_mandarina = cv2.CascadeClassifier('mandarina.xml')
haar_manzana = cv2.CascadeClassifier('manzana.xml')
haar_pera = cv2.CascadeClassifier('pera.xml')
haar_platano = cv2.CascadeClassifier('platano.xml')

def detectar_frutas(frame):
```

```

hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

for fruta, (lower, upper) in frutas_colores.items():
    mask = cv2.inRange(hsv, lower, upper)
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Encuentra el contorno más grande dentro del rango de color
    if len(contours) > 0:
        max_contour = max(contours, key=cv2.contourArea)
        (x, y, w, h) = cv2.boundingRect(max_contour)

    # Aplica un umbral en el área del contorno para filtrar los contornos más pequeños
    # o más grandes
    area_threshold = 100 # Ajusta este valor según tus necesidades
    if area_threshold < cv2.contourArea(max_contour) < frame.shape[0] * frame.shape[1]:
        # Dibuja un rectángulo alrededor de la fruta
        if fruta == 'mandarina':
            rects = haar_mandarina.detectMultiScale(frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))
            for (x, y, w, h) in rects:
                cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 165, 255), 2)
                cv2.rectangle(frame, (x, y), (x + 100, y + 40), (0, 165, 255), -1)
                cv2.putText(frame, fruta.capitalize(), (x + 10, y + 30), font, 0.7,
(255, 255, 255), 2)
                cv2.putText(frame, frutas_nutricion[fruta], (x + 10, y + 60), font,
0.7, (255, 255, 255),
2)
            if cv2.waitKey(1) & 0xFF == ord('c'):
                print("Se ha tomado una foto de una mandarina.")
                print(),
                print(valor_valores[fruta])
                print(),
                # Aquí puedes guardar la imagen usando cv2.imwrite() si deseas
            elif cv2.waitKey(1) & 0xFF == ord('q'):
                print("Se ha detenido el video.")
                break
        elif fruta == 'manzana':
            rects = haar_manzana.detectMultiScale(frame, scaleFactor=1.5,
minNeighbors=5, minSize=(30, 30))
            for (x, y, w, h) in rects:
                cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
                cv2.rectangle(frame, (x, y), (x + 100, y + 40), (0, 0, 255), -1)
                cv2.putText(frame, fruta.capitalize(), (x + 10, y + 30), font, 0.7,
(255, 255, 255), 2)
                cv2.putText(frame, frutas_nutricion[fruta], (x + 10, y + 60), font,
0.7, (255, 255, 255),
2)
            if cv2.waitKey(1) & 0xFF == ord('c'):
                print("Se ha tomado una foto de una manzana.")
                print(),
                print(valor_valores[fruta])
                print(),
                # Aquí puedes guardar la imagen usando cv2.imwrite() si deseas
            elif cv2.waitKey(1) & 0xFF == ord('q'):
                print("Se ha detenido el video.")
                break
        elif fruta == 'pera':
            rects = haar_pera.detectMultiScale(frame, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

```

```

for (x, y, w, h) in rects:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    cv2.rectangle(frame, (x, y), (x + 100, y + 40), (0, 255, 0), -1)
    cv2.putText(frame, fruta.capitalize(), (x + 10, y + 30), font, 0.7,
(255, 255, 255),
2)
    cv2.putText(frame, frutas_nutricion[fruta], (x + 10, y + 60), font,
0.7, (255, 255, 255),
2)

    if cv2.waitKey(1) & 0xFF == ord('c'):
        print("Se ha tomado una foto de una pera.")
        print(),
        print(valor_valores[fruta])
        print(),
        # Aquí puedes guardar la imagen usando cv2.imwrite() si deseas
    elif cv2.waitKey(1) & 0xFF == ord('q'):
        print("Se ha detenido el video.")
        break
    elif fruta == 'platano':
        rects = haar_platano.detectMultiScale(frame, scaleFactor=1.03,
minNeighbors=4, minSize=(26, 28))
        for (x, y, w, h) in rects:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 2)
            cv2.rectangle(frame, (x, y), (x + 100, y + 40), (0, 255, 255), -1)
            cv2.putText(frame, fruta.capitalize(), (x + 10, y + 30), font, 0.7,
(255, 255, 255), 2)
            cv2.putText(frame, frutas_nutricion[fruta], (x + 10, y + 60), font,
0.7, (255, 255, 255),
2)

            if cv2.waitKey(1) & 0xFF == ord('c'):
                print("Se ha tomado una foto de un plátano.")
                print(),
                print(valor_valores[fruta])
                print(),
                # Aquí puedes guardar la imagen usando cv2.imwrite() si deseas
            elif cv2.waitKey(1) & 0xFF == ord('q'):
                print("Se ha detenido el video.")
                break

while True:
    ret, frame = cap.read()
    if not ret:
        break

    detectar_frutas(frame)

    k = cv2.waitKey(1)
    if k == 27:
        break
    cv2.imshow('frame', frame)

cap.release()
cv2.destroyAllWindows()

```

Resultados Alcanzados

El modelo de reconocimiento de frutas ha demostrado una gran precisión en la identificación de diferentes tipos de frutas.

Gracias a su capacidad de procesamiento de imágenes y algoritmos avanzados de aprendizaje automático, el modelo puede reconocer con exactitud las características relevantes de cada fruta.

Al identificar las frutas, se puede proporcionar información nutricional precisa y detallada sobre cada una de ellas, esto incluye datos sobre calorías, macronutrientes (proteínas, carbohidratos y grasas), vitaminas, minerales y otros compuestos beneficiosos para la salud.

El modelo presenta herramientas valiosas tanto para nutricionistas como para las personas en general, ya que brindan la posibilidad de adquirir conocimientos fundamentados y tomar decisiones informadas en relación a la alimentación, fomentando así un estilo de vida saludable, además el modelo puede adaptarse a las preferencias dietéticas o restricciones individuales, como dietas vegetarianas, veganas, sin gluten o bajas en carbohidratos.

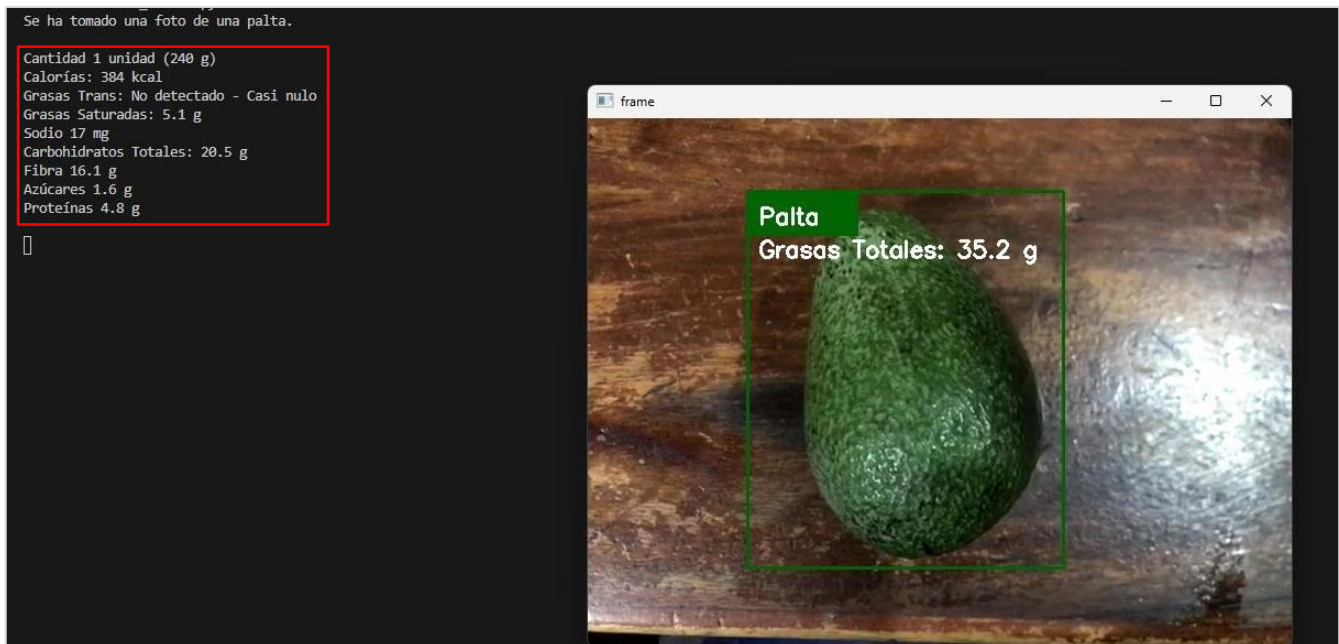


Ilustración 6: Reconocimiento de la fruta Palta con su Valor Nutricional

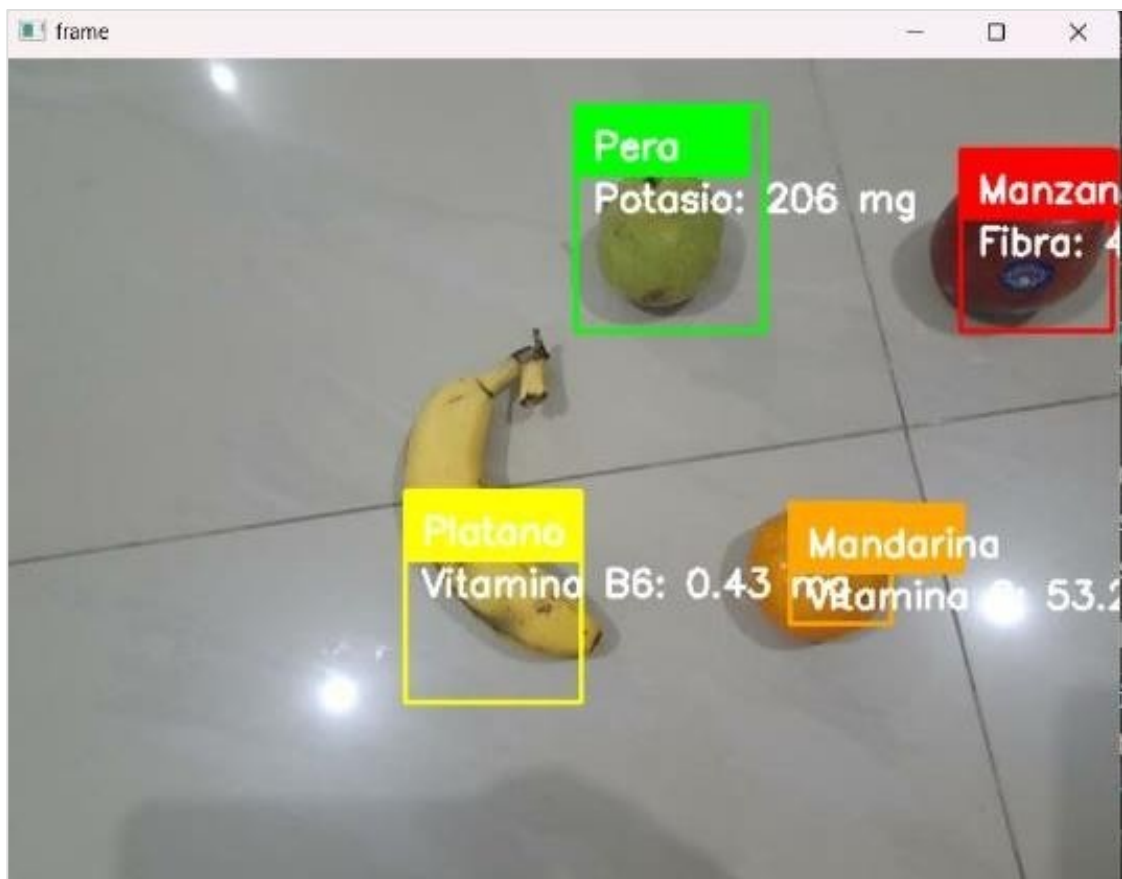


Ilustración 7: Reconocimiento de las frutas Pera, Manzana, Plátano y Mandarina con su Valor Nutricional

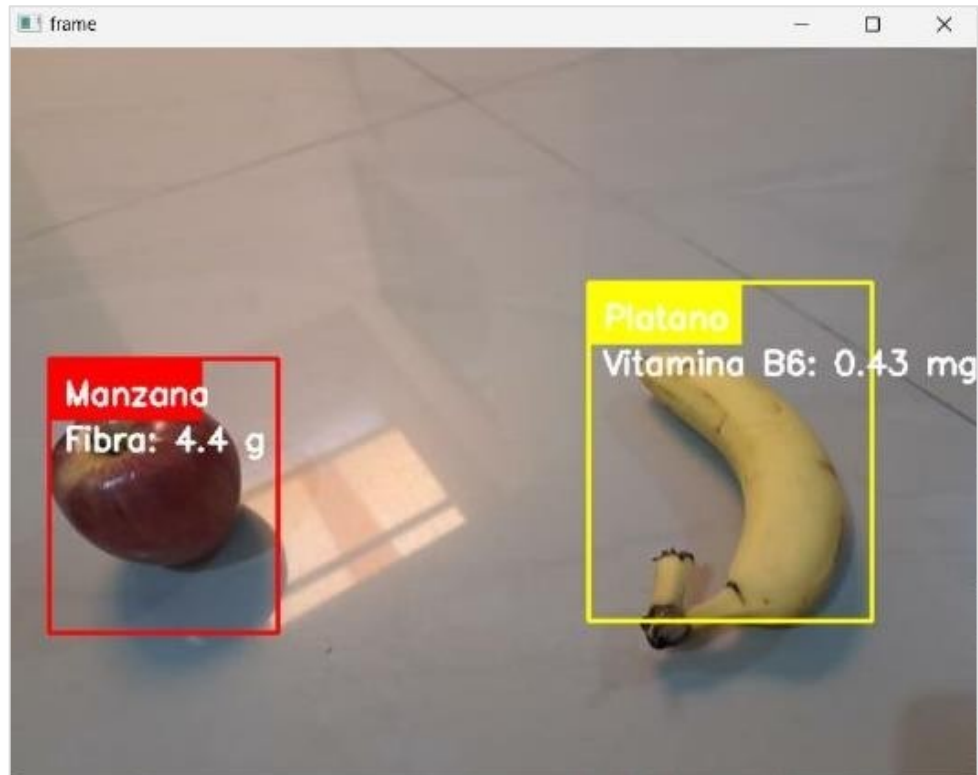


Ilustración 8: Reconocimiento de las frutas Manzana y Plátano con su Valor Nutricional

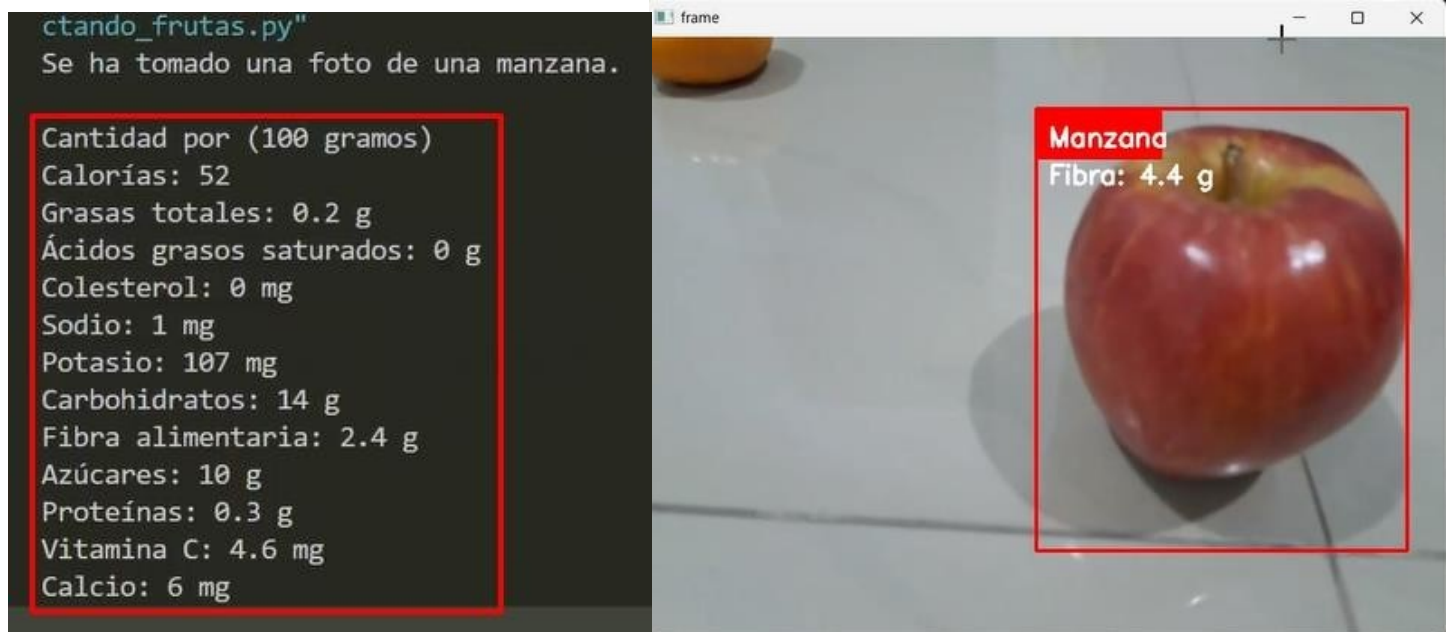


Ilustración 9: Reconocimiento de la fruta Manzana con su Valor Nutricional

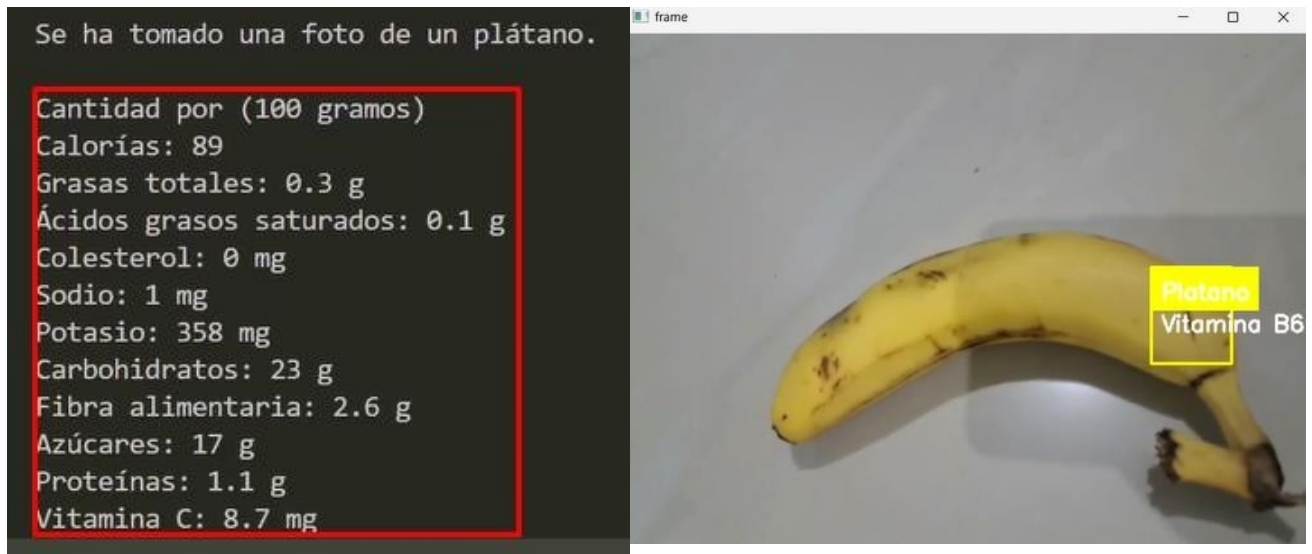


Ilustración 10: Reconocimiento de la fruta Plátano con su Valor Nutricional

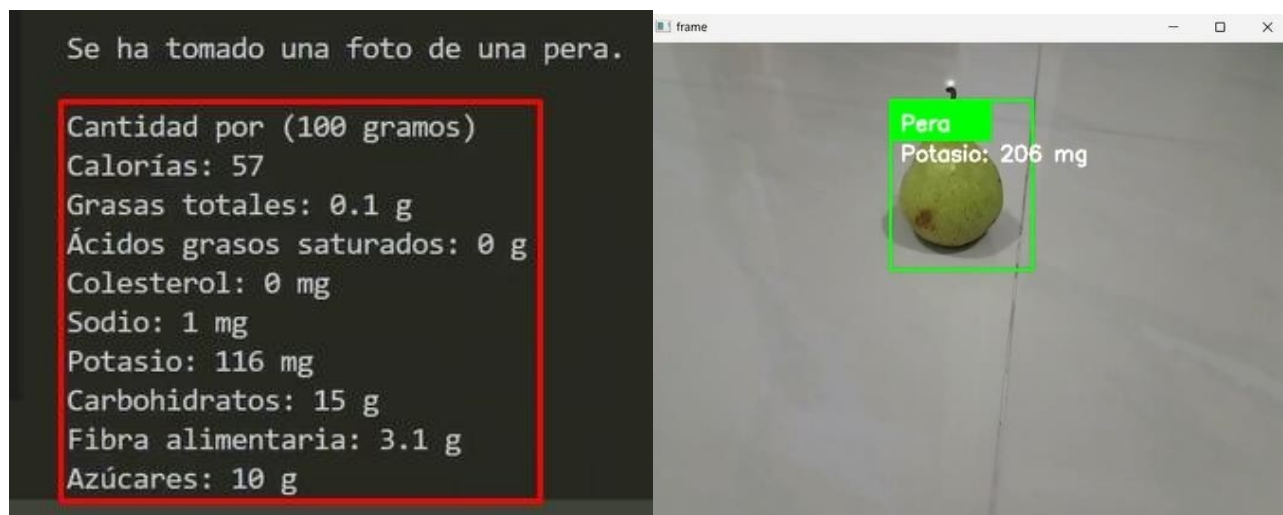


Ilustración 11: Reconocimiento de la fruta Pera con su Valor Nutricional

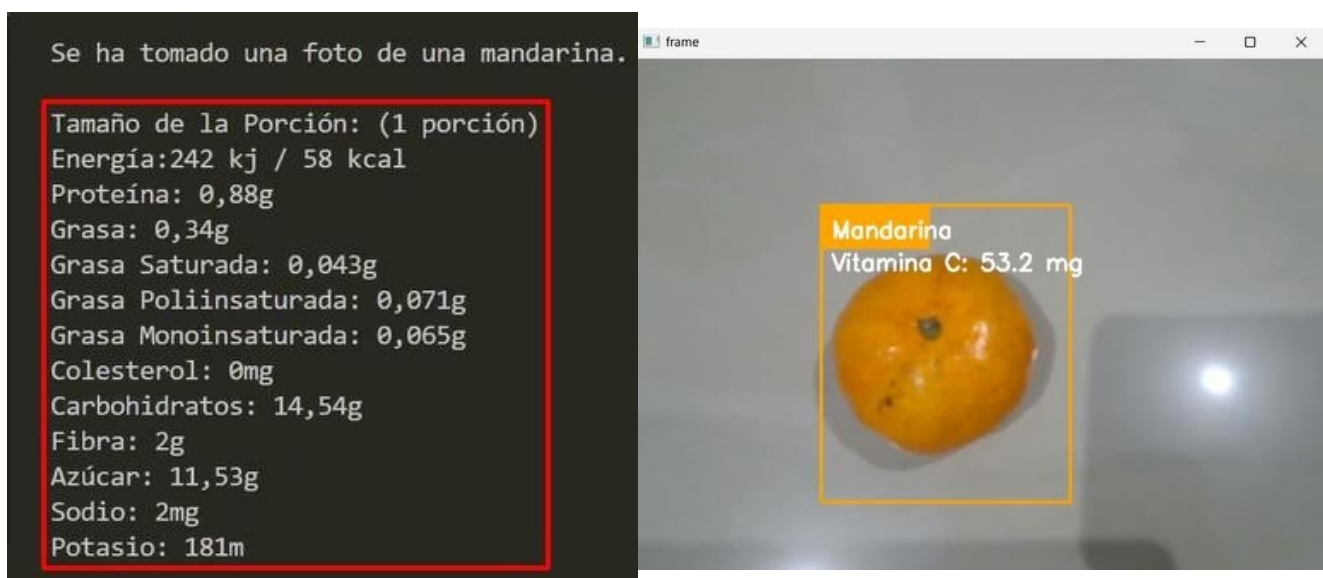


Ilustración 12: Reconocimiento de la fruta Mandarina con su Valor Nutricional

Discusión o Reflexión Crítica

En primer lugar, nuestro modelo se basa en un conjunto limitado de frutas, y es necesario considerar añadir otras frutas para aumentar la diversidad de las muestras.

En segundo lugar, la precisión de nuestro modelo puede verse afectada por la calidad de las imágenes que utilizamos al momento de entrenarlo, por lo que se necesita garantizar la calidad de las imágenes utilizadas para el entrenamiento y evaluación del modelo.

Por último, es importante resaltar que el modelo solo es capaz de reconocer las frutas, mostrar su nombre y uno de sus valores nutricionales, los demás los muestra en la consola.

Referencias Bibliográficas

<https://amin-ahmadi.com/cascade-trainer-gui/>

https://docs.opencv.org/4.3.0/dc/d88/tutorial_traincascade.html

https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052014000400006

<https://www.redalyc.org/journal/5600/560062814010/html/>

<https://www.researchgate.net/publication/321176883> Clasificador de imagenes de frutas b
asado en inteligencia artificial

<https://fitia.app/calorias-informacion-nutricional/palta-10107/>

<https://pinetools.com/es/obtener-colores-imagen>

<https://www.peko-step.com/es/tool/hsvrgb.html>

[https://www.fatsecret.cl/calor%C3%ADas-
nutrici%C3%B3n/gen%C3%A9rico/mandarinas?portionid=33240&portionamount=1,000](https://www.fatsecret.cl/calor%C3%ADas-nutrici%C3%B3n/gen%C3%A9rico/mandarinas?portionid=33240&portionamount=1,000)

<https://www.youtube.com/watch?v=eN4h2x-Uu5c>

<https://www.youtube.com/watch?v=MUsrAO4hQtE>