

## **PROYECTO FINAL IMÁGENES**

### **IDENTIFICACION Y DISCRIMINACIÓN DE AGUACATES COMO OBJETO DE INTERÉS**

#### **I. INTRODUCCIÓN**

El aguacate es considerado como el oro verde de la economía colombiana por [1], siendo el cuarto país exportador en el mundo, en el último año las exportaciones de aguacate pasaron de US\$10 millones a US\$35 millones, este cultivo centra las esperanzas de comercio en el país.

Sin embargo, el aguacate y las frutas en general provenientes de Colombia han sido calificadas medianamente en 5.9 de 10 en el ranking de calidad europeo, a comparación de otros países como Perú y Chile que tiene puntuación de 7.1 y 7.4 respectivamente [2]. Lo que representa una oportunidad de mejora en los estándares y procesos de calidad que le permitan al país la permanencia y escalamiento en el mercado internacional.

Actualmente, el proceso de cultivo y clasificación del aguacate para exportación se hace de manera artesanal, lo cual representa un alto porcentaje de error en el cálculo de los tiempos de transporte en la exportación y tiempos de maduración, por lo cual en el presente proyecto se propone realizar el análisis por medio de procesamiento de imágenes, donde se identifique y discrimine la forma del aguacate entre otros posible objetos y como futuro trabajo de investigación se propone clasificar por color en un entorno de luz controlada para evitar variaciones en la toma de muestras.

## II. DIAGRAMA DE FLUJO DE LA SOLUCIÓN

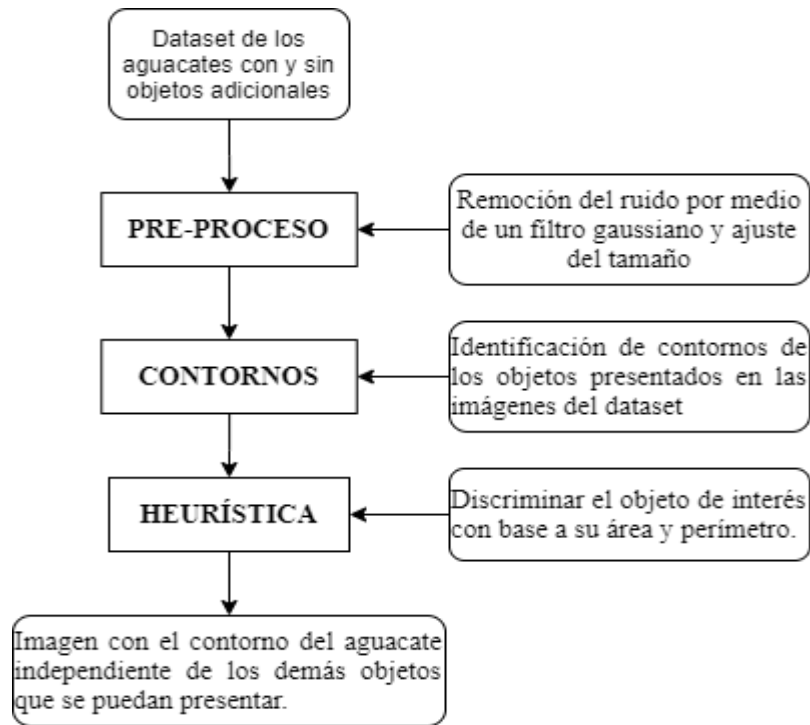


Figura 1. Diagrama de flujo del proyecto.

## III. DESARROLLO

### Cambio de tamaño de la imagen

Este bloque se desarrolla con el objetivo de cambiar el tamaño de la imagen del aguacate, ya que su visualización en el *software* SPYDER no se veía de la manera correcta a la imagen ser de muy alta definición. De esta forma, se procede a cambiar el tamaño de la imagen, donde se le indica el porcentaje al cual se quiere ver. En este proyecto en particular se reduce el tamaño a un 10%, sin perder las dimensiones de la imagen.

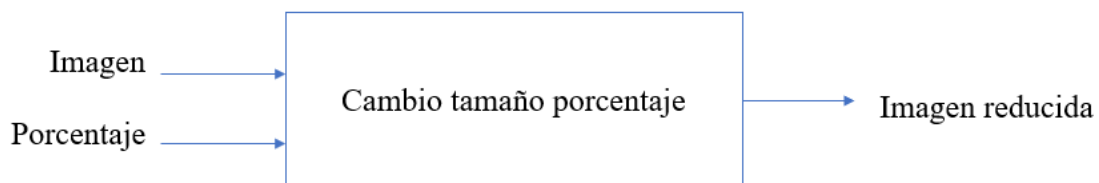


Figura 2. Cambio tamaño porcentaje.

Entradas:

- Imagen original de gran tamaño
- Porcentaje de la imagen que se quiere obtener para su posterior procesamiento.

Salida: Imagen reducida al porcentaje requerido.

### Filtro Gaussiano por color

Este bloque se encarga de tomar la imagen reducida del aguacate y aplicarle un filtro gaussiano por color, donde retorna una imagen a blanco y negro, resaltando como color blanco los objetos de la imagen que corresponden al color elegido, en este caso, un rango significativo con la intención de identificar el aguacate.

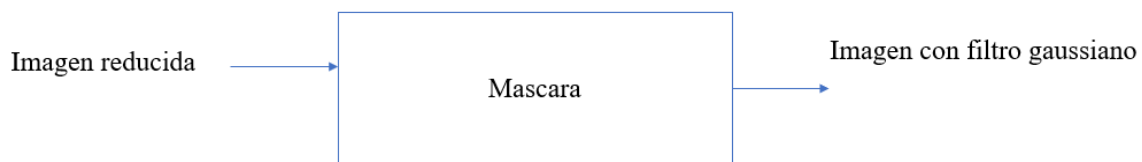


Figura 3. Mascara.

Entrada: Imagen reducida.

Salida: Imagen blanco y negro sometida al filtro gaussiano.

### Identificación y procesamiento de contornos

Este bloque se encarga de identificar los contornos de la imagen filtrada, donde seguidamente mediante heurística se determina por el área y el perímetro del contorno, cual es el contorno del aguacate. Para de esta forma identificarlo y posteriormente graficarlo en la imagen original del aguacate.

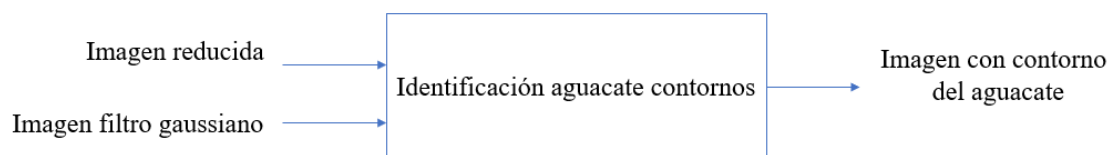


Figura 4. Identificación aguacate contornos.

Entradas:

- Imagen reducida: Imagen que fue cambiada de tamaño de la original para facilidad de visualización.
- Imagen Filtro Gaussiano: Imagen que fue sometida a un filtro de ruido para obtener la máscara, donde se identifica solo un rango de colores.

Salida: Imagen reducida ilustrando el contorno del aguacate identificado.

En este proceso se identificó que el área del aguacate debe estar entre 10mil y 40mil pixeles cuadrados de área del contorno y un perímetro de 500 a 1800 pixeles. Esto se identifico determinando las diferentes áreas y perímetros de diferentes fotos de aguacates del data set. Así que de esta forma se logra filtrar los contornos que realmente son útiles, y finalmente si se detecta mas de un objeto se tiene prioridad al área, debido a que pueden existir otros objetos con el mismo perímetro.

### Proceso conjunto

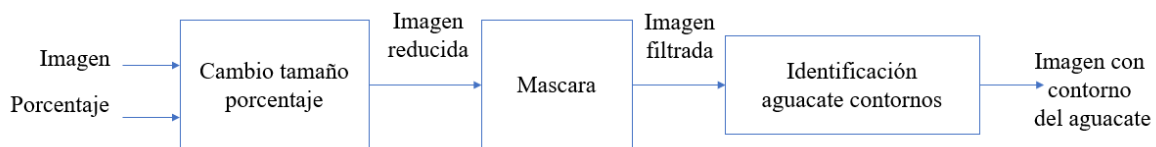


Figura 5. Unificación bloques

Finalmente, como se puede observar en la figura, el proceso completo se caracteriza por ingresar una imagen que contenga un aguacate y un porcentaje para reducir la imagen, y posteriormente se graficara la misma imagen, identificando y graficando el contorno del aguacate detectado.

## IV. RESULTADOS

Se realizo la prueba con diferentes fotografías de aguacates tomadas en un laboratorio, tomadas de otro trabajo de grado. Donde al tener este data set de imágenes, el código elige de forma aleatoria, una de las imágenes y posteriormente se muestra la identificación de este.

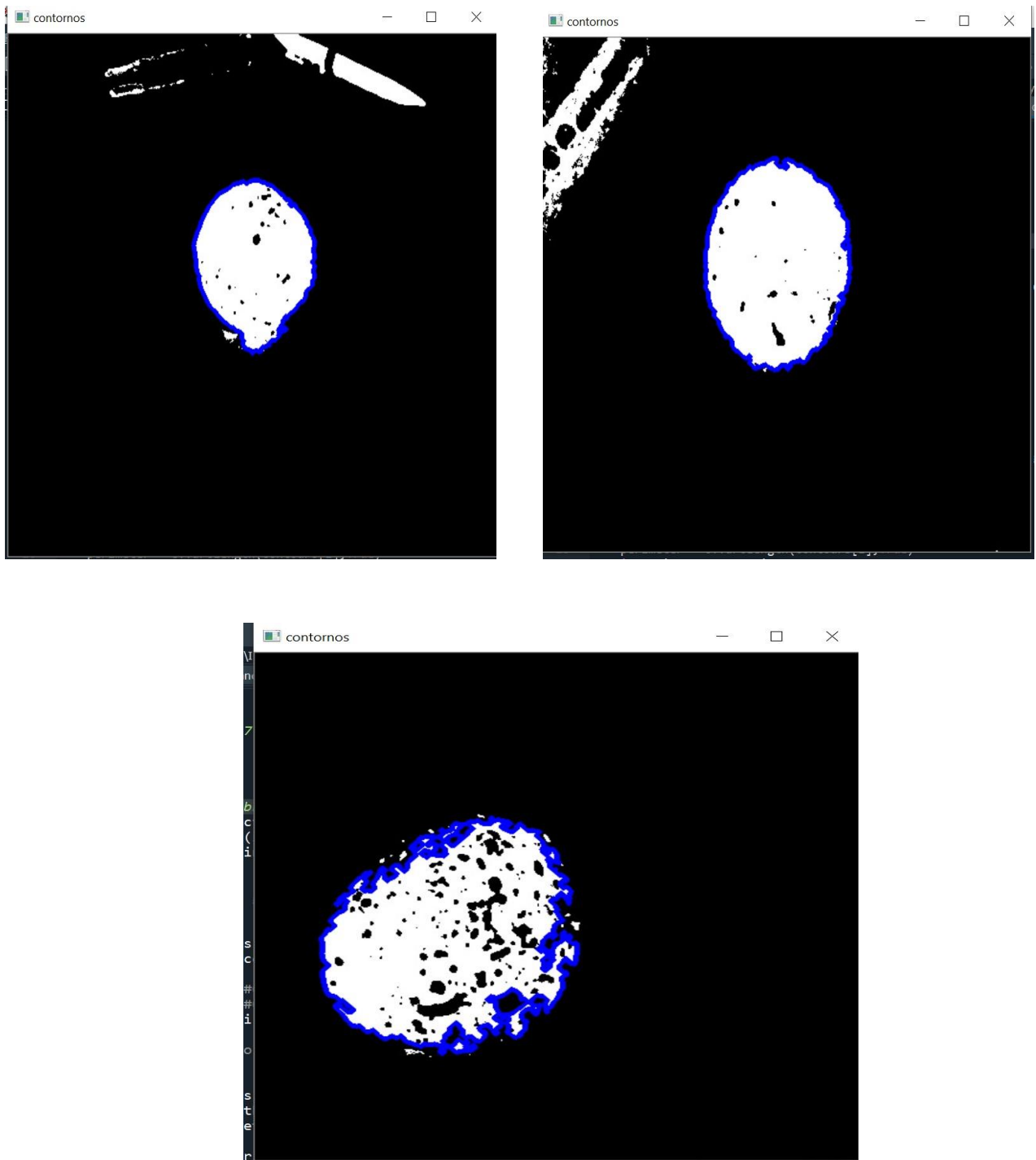


Figura 6. Imagen blanco y negro sometida al filtro gaussiano.

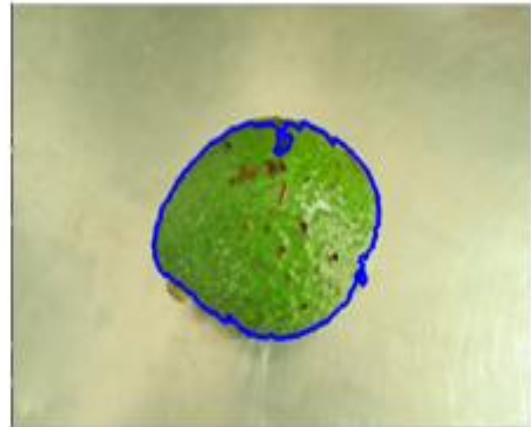


Figura 7. Discriminación de los aguacates con distintos objetos.

En la figura se puede observar que reconoce el objeto de manera satisfactoria, donde se omiten los contornos de otros objetos como se puede observar en la imagen, así como de esferos, cargadores, cables, etc. Donde fue necesario realizar un filtrado por color, pero, aun así, se identificaban contornos de sombras u otros objetos. Así que fue necesario filtrar por área y perímetro para finalmente obtener los contornos únicamente de los aguacates.

## **V. CONCLUSIONES**

1. Colombia tiene un potencial gastronómico importante que se está empezando a explotar, beneficiando económica y culturalmente el país. Sin embargo, aún no se han implementado en gran medida la tecnología para mejorar y automatizar los procesos para de esta manera volver más atractivo el producto colombiano e incrementar los ingresos provenientes de exportaciones. Con base a lo anterior y desde la ingeniería con el uso de nuevas tecnologías tenemos el compromiso de contribuir con el mejoramiento continuo de procesos.
2. Para realizar el proceso de identificación y discriminación del aguacate se hace relevante aplicar un filtro, en este caso se decide aplicar el filtro GaussianBlur debido a que su respuesta al impulso es una función gaussiana, por lo tanto, tienen la propiedad de no sobrepasar una entrada de función escalonada y minimizar el tiempo de subida y bajada, filtrando un kernel de 5x5 y un ajuste de tamaño al data set con el objetivo de que solo se obtengan los objetos de un color predeterminado y posteriormente hallar los contornos del mismo.
3. Para identificar el objeto deseado, en este caso el aguacate de los otros objetos que posiblemente se encuentren en las imágenes del dataset empleado, se requiere inicialmente predeterminar el área y perímetro aproximado del objeto de interés, estableciendo un umbral de la medida y de esta manera se hace la analogía constante y se discrimina lo que corresponde o no al aguacate.

## **VI. REFERENCIAS**

[1] Revista Dinero “Aguacate: el oro verde de la economía colombiana” disponible en: <https://www.dinero.com/edicion-impresa/informe-especial/articulo/aguacate-exportacion-y-mercado-en-colombia/243434>.

[2] J. Ramirez Gil, H. López, and J. Henao Rojas, “Causes of Hass Avocado Fruit Rejection in Preharvest, Harvest, and Packinghouse: Economic Losses and Associated Variables,” in MDPI, 2019.

## VII. CODIGO

```
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 30 12:21:43 2020

@author: nnico
"""

import cv2 as cv
import numpy as np
import glob
from random import sample

# Cambia al tamaño de la imagen
def cambio_tamano_porcentaje(image,scale_percent):
    width = int(image.shape[1] * scale_percent / 100)
    height = int(image.shape[0] * scale_percent / 100)
    dim = (width, height)

    img = cv.resize(image, dim, interpolation = cv.INTER_AREA)
    return img

# Aplica la mascara de bordes segun color
def mascara(img):
    blurred_frame = cv.GaussianBlur(img, (5, 5), 0)
    hsv = cv.cvtColor(blurred_frame, cv.COLOR_BGR2HSV)
    lower_blue = np.array([38, 86, 0])
    upper_blue = np.array([121, 255, 255])
    mask = cv.inRange(hsv, lower_blue, upper_blue)
    return mask

# Identifica el contorno segun area y perimetro
def identificacion_aguacate_contornos(img,mask):
    imgray = mask
    #imgray = cv.cvtColor(mask, cv.COLOR_BGR2GRAY)
    ret, thresh = cv.threshold(imgray, 127, 255, 0)
    contours, hierarchy = cv.findContours(thresh, cv.RETR_TREE,
cv.CHAIN_APPROX_NONE)

    # Encontrando por area
    cnt_area_grande = -1
    #max_area = -1
```



```

for i in range(len(contours)):
    area = cv.contourArea(contours[i])
    if area>10000 and area <40000: # Area del aguacate
        #cnt_area = contours[i] #Guarda el contorno que corresponde al rango
        #max_area = area #Guarda el area que corresponde al rango
        cnt_area_grande = i #Guarda el numero del contorno del aguacate

# Encontrando por perimetro
cnt_perimetro_grande = -1
#max_perimeter = -1
for i in range(len(contours)):
    perimeter = cv.arcLength(contours[i],True)
    if perimeter>500 and perimeter <1800: #Perimetro aguacate
        #cnt_perimeter = contours[i] #Guarda el contorno que corresponde al
rango
        #max_perimeter = perimeter #Guarda el perimetro que corresponde al
rango
        cnt_perimetro_grande = i #Guarda el numero del contorno del aguacate

# Si el contorno elegido es diferente para perimetro y area, se tiene como
prioridad
# el area debido a que el numero establecido es mas cercano
if cnt_area_grande != cnt_perimetro_grande:
    cnt_perimetro_grande = cnt_area_grande

# Se dibuja el contorno identificado
cv.drawContours(img, contours, cnt_area_grande, (255,0,0), 3)
#cv.drawContours(img, contours, cnt_perimetro_grande, (255,0,0), 3)

cv.imshow('contorno_aguacate',img)
cv.waitKey(0)
cv.destroyAllWindows()

# PROCEDIMIENTO -----

scale_percent = 10 # Porcentaje de cambio de tamaño de la imagen

imagenes = glob.glob('*.JPG') # Imagenes de agacates disponibles
images=sample(imagenes,1) # Eleccion de una imagen

```

```
#image = cv.imread('4.jpg')
```

```
for fname in images:
```

```
    image = cv.imread(fname)
```

```
    img = cambio_tamano_porcentaje(image,scale_percent)
```

```
    mask = mascara(img)
```

```
    identificacion_aguacate_contornos(img,mask)
```