

Development of an algorithm to detect the Bloom effect on the blueberries.

León León, RyanAbraham ¹; Gómez Avalos, Nayelly Massiel ¹; Lafitte Chong, Adriana Cristina ¹; ¹Industrial Engineering Degree, Universidad Privada del Norte, Trujillo, Perú, ryan.leon@upn.edu.pe, N00203619@upn.pe, N00204144@upn.pe

Abstract - This report focused on the recognition of blueberries using the segmentation method and the post selection of the blueberries identified as good. The image change method from RGB to HSV was applied, which allowed a better discrimination of blueberries in relation to the background and other elements present in the image.

Successful segmentation was achieved by creating masks based on the values of hue, saturation and value (HSV). Classification based solely on the absolute size of blueberries may not be enough to guarantee an accurate and consistent classification. With the Hough transformation method, it will be possible to better discriminate blueberries by approaching them in a circular way. However, an image filter was used with acceptable Bloom concentration ranges for greater accuracy.

In conclusion, this report highlights the success in the recognition of blueberries through the segmentation method and the presentation of the blueberries selected as good. With an average effectiveness of 98%, favorable results can be obtained so that the process can be used to mention as valid for the objectives set.

Keywords- SEGMENTATION, CONTOUR SEARCH, RGB LAYER AND HSV

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Desarrollo de un algoritmo para detectar el efecto Bloom en los arándanos.

León León, Ryan Abraham ¹ ; Gómez Avalos, Nayelly Massiel ¹ ; Lafitte Chong, Adriana Cristina ¹ ;

¹Industrial Engineering Degree, Universidad Privada del Norte, Trujillo, Perú,
ryan.leon@upn.edu.pe, N00203619@upn.pe, N00204144@upn.pe

El presente informe se enfocó en el reconocimiento de arándanos utilizando el método de segmentación y la posterior selección de los arándanos identificados como buenos. Se aplicó el método de cambio de imagen de RGB a HSV, lo cual permitió una mejor discriminación de los arándanos en relación con el fondo y otros elementos presentes en la imagen.

Se logró una segmentación exitosa mediante la creación de máscaras basadas en los valores de matiz, saturación y valor (HSV). La clasificación basada únicamente en el tamaño absoluto de los arándanos puede no ser suficiente para garantizar una clasificación precisa y consistente. Con el método de la transformación de Hough se podrá discriminar de mejor forma los arándanos aproximándolos de forma circular. Sin embargo, se utilizó un filtro de imagen con rangos de concentración de Bloom aceptables para su mayor precisión.

En conclusión, este informe destaca el éxito en el reconocimiento de arándanos mediante el método de segmentación y la presentación de los arándanos seleccionados como buenos. Con una efectividad promedio de 98% se puede obtener resultados favorables de forma que se pueda utilizar el proceso a mencionar, cómo válido para los objetivos planteados.

Palabras Claves - SEGMENTACIÓN, BÚSQUEDA DE CONTORNOS, CAPA RGB Y HSV

I. INTRODUCCIÓN

Los arándanos son frutas ampliamente consumidas en todo el mundo a causa de su agradable sabor y por los grandiosos beneficios que brinda para la salud. Sin embargo, la calidad de los arándanos frescos se puede ver comprometido por muchos factores, uno de ellos es el fenómeno conocido como “Bloom”. La importancia de este recubrimiento ceroso permite conocer la conservación de calidad y frescura del fruto.

La detección del efecto Bloom en los arándanos cumple un papel sumamente importante en la agroindustria sobre todo en Perú, puesto que el blueberry es altamente demandado a nivel mundial siendo Perú uno de los principales proveedores en el mundo de este maravilloso fruto. Sin embargo, en las empresas hay técnicas que actualmente carecen de alta precisión en la selección de arándanos con Bloom, produciendo un incremento de sus costos, generando pérdidas.

En el presente trabajo se desarrolló un algoritmo para detectar el efecto Bloom en los arándanos, proponiéndola como una alternativa de implementación efectiva, que permite evaluar la calidad del arándano en tiempo real, optimizando procesos de selección. Para ello, se optó por trabajar con técnicas de visión artificial con el software Python en el cual se identifique mediante imágenes, las condiciones en las que se encuentra esta fruta diferenciando así los arándanos con y sin efecto Bloom; una vez ejecutado ello, se procederá a calcular la precisión del algoritmo en base a los resultados de las pruebas a ejecutar de este. Para lograr ejecutar la programación requerimos de conocimientos previos, así podremos desarrollar un algoritmo que detecte el efecto Bloom de los arándanos a analizar en base a su aspecto

físico. Así mismo [1] mencionan los beneficios de su proyecto aplicado en alimentos. El autor desarrolló una báscula capaz de clasificar frutas, con lo cual obtuvo solución ante la identificación y clasificación de productos agrícolas en las tiendas comerciales. Por otro lado, [2] empresas exportadoras de alimentos ejecutan su proceso de selección de manera manual y visual por el ser humano, ante ello, se ejecutó un sistema que busca estandarizar el lote cosechado sin que sea afectado por factores externos como el cansancio y fatiga visual de los operadores; adicionalmente a ello, el modelo de visión por computadora permite recopilar información de la muestra mucho más rápido, la misma que al ser analizada permitirá tomar una decisión adecuada con respecto a la selección del producto. Por otro lado [3] la aplicación de tecnología en el área de calidad de alimentos es de suma importancia, y cuando se trata de alimentos que se exportan aún más, es por ello por lo que en su trabajo de investigación midió la eficiencia entre la visión por computadora y el método de clasificación, en este caso de paltas, por un experto en el área, obteniendo como resultado un 91.5% a favor de la tecnología.

El programador Python será la herramienta principal para la ejecución de este trabajo, ya que con la ayuda de este se logró ejecutar el algoritmo para dar solución a nuestra problemática.[4] Para un proyecto se realizó una aplicación en el programador Python que tenga la capacidad de controlar los múltiples equipos del laboratorio de ingeniería eléctrica. Como resultado se obtuvo una mejor interacción con los equipos de laboratorio gracias al lenguaje de programación Python y el desarrollo de interfases gráficas PyQt pues permite un fácil manejo de los artefactos en mención. [5] La herramienta Python permitió diseñar e implementar un sistema de alertas que brinden soporte en la toma de decisiones en el área de planta del sector agrícola, obteniendo como resultado del uso de modelos predictivos un aumento del 80% de la eficacia en lo que respecta la toma de decisiones en dicha área.

Respecto a la identificación por color, [6] la investigación realizada por estudiantes de ingeniería de sistemas de la universidad César Vallejo, busca describir un sistema de control de un robot fabricado por Tekniker para recoger de forma autónoma una pelota de color rojo, amarillo o azul en un recinto abierto. Obteniendo como resultado una eficiencia de 71.1%, en caso de presentar una correcta iluminación este porcentaje aumenta a 87.4%. Apoyando lo mencionado, [7] ejecuta algoritmos matemáticos y de procesamiento de imagen capaces de reconocer a través de la escala de colores RGD de una imagen, la composición química propuesta y así poder extraer los parámetros de interés concluyendo con lo siguiente, se elaboró un diseño que permite a usuarios con distintas necesidades y a la vez se desarrollaron distintos algoritmos capaces de minimizar el tiempo de procesamiento de cada muestra, cumpliendo las expectativas previstas. [8] Aportando a ello, también es relevante que desde el área de Seguridad también la tecnología sea de tanto aporte; la delincuencia que actualmente existe en nuestra sociedad fue el incentivo que permitió crear un reconocedor facial en Python que le permitirá a múltiples hogares e instituciones incrementar el nivel de seguridad en sus locaciones.

[9] plantea un sistema de visión artificial con la finalidad de detectar del café su fruto para lograr producirlo; para lograr este fin se ejecutó el desarrollo de dos algoritmos. El primero se encarga de la clasificación, y el segundo detecta la presencia de broca; dicho trabajo obtuvo una eficiencia del 87%. Además, [10] utiliza de la misma manera visión artificial, adicionalmente a ello, se incorporó el uso de redes neuronales, con la finalidad de seleccionar las particularidades de las frutas a analizar; para el desarrollo de este se emplearon librerías como lo es OpenCV. El manejo de frutas es primordial, [11] señala que el desarrollo de una herramienta que permita la detección de frutas, más específicamente la granadilla, fue logrado gracias al reconocimiento de imágenes, obteniendo como resultado un 92,6% de eficiencia.

Por otra parte, [12] buscan crear un modelo de inteligencia artificial basada en CNN, el cual pueda deducir de manera rápida una prueba de COVID 19. Atraves de una CNN podemos diagnosticar diferentes enfermedades aplicando imágenes, lo cual ayudará a aligerar la atención de los pacientes con una detección temprana de síntomas del COVID 19, será de gran ayuda a optimizar el tiempo debido a la gran cantidad de pacientes sintomáticos y asintomáticos. Así mismo, [13] en su trabajo busca saber qué parámetros hacen posible la supremacía del cerebro en unas determinadas tareas frente a los computadores. Los autores llegaron a la conclusión que las neuronas son mucho más simples, lentas y menos fiables que una CPU, sin embargo, existen tareas que el cerebro resuelve eficazmente, por ejemplo: respuestas ante estímulos del entorno, reconocimiento del habla, etc. Tienen como objetivo buscar que las ANS aborden todos los problemas que el cerebro humano resuelve con rapidez y eficiencia. Para lograr ello sería conveniente construir sistemas que simulen la estructura de las redes neuronales biológicas con la finalidad de alcanzar una funcionalidad similar.

[14] tiene como finalidad en su trabajo de investigación dar una alternativa de solución al control de vehículos que ingresan al centro histórico de Arequipa. Para ello se busca elaborar un algoritmo en Python utilizando redes neuronales que dé como resultado un registro del número de placa de los conductores que infringen la ordenanza municipal N° 927-MPA. En primer lugar, nos habla de la detección de placas usando redes neuronales convolucionales; la segunda parte consta en reconocer el texto de las placas localizadas y finalmente crear una tabla en Excel indicando si un vehículo debió o no ingresar al centro histórico. También [15] expone en su proyecto el uso de redes neuronales convolucionales ante la insuficiencia en procesos de reposición de productos y actualización y mantención de etiquetado de los productos que presenta un supermercado de estatus estrato económico medio-bajo. Una vez realizado y aplicado el programa, se demostró que el uso de esta tecnología es eficiente para solucionar los problemas expuestos.

Por otro lado, [16] ejecuta el desarrollo de un sistema capaz de reconocer objetos a partir de una imagen, por ejemplo, una fotografía. El objetivo es que el sistema sea capaz de identificar el objeto en cuestión y proporcionar información relevante, como su nombre, existencia, precio, ubicación, entre otros datos. Para lograr esto, se utilizará un algoritmo basado en FLANN y SIFT, y como soporte una Raspberry Pi 2 para su funcionamiento óptimo. [17] creó un robot autónomo utilizando visión inteligente Faster r-cnn en el programador Python con la capacidad de brindar una eficiente dosificación de herbicidas para el control malezas en el sector agrícola con la finalidad de prevenir la propagación de especies malignas en los cultivos a causa del inadecuado practicas por parte de los trabajadores. [18] Se comenta que el mundo de la inteligencia artificial, visión por computador, entre otros, se convierte en el pilar de este proyecto. El trabajo de investigación ejecutó un software que identifique

manuscritos aplicando redes neuronales en el programa Python con un éxito de 97.4%.

El objetivo de este trabajo radica en la ejecución de un algoritmo que permita una selección de arándanos de buena calidad, para ello se necesita realizar la toma de la imagen de forma continua, establecer valores de clasificación más objetivas y menos subjetivos como tamaño y color y finalmente realizar una clasificación por colores y tonalidades, obteniendo los valores de opacidad y tamaño para la clasificación adecuada.

II. MATERIALES Y MÉTODOS

Adquisición de imágenes y establecimiento de parámetros

Para la adquisición de nuestras imágenes ,se realizó una fotografía de cada distribución de arándano mediante una cámara de resolución alta, cabe mencionar que el tamaño de la imagen no tiene limitación, debido a que luego se tiene que redimensionar la foto existente. Se hizo un muestreo con 50 fotografías de arándanos azules, entre los cuales se eligieron con diferentes porcentajes de Bloom, se eligió trabajar con los azules debido a que el mercado proporciona comúnmente este fruto en color azul.

Se eligió una tonalidad de 55 de 255 para bit gris y tamaño promedio entre 20 y 35 pixeles con lo cual se realizó la programación del sistema.

Cambio de tipo de imagen RGB a HS

El modelo de color HSV es una forma más intuitiva de representar los colores, ya que se basa en la forma en que los seres humanos perciben y describen los colores. Aquí está una breve descripción de cada componente:

Matiz (H): Representa el tono o la posición cromática del color en una escala circular, que va desde 0° a 360°. Los valores cercanos a 0° y 360° corresponden al rojo, mientras que los valores intermedios representan otros colores en el espectro.

Saturación (S): Indica la pureza o intensidad del color. Valores más altos de saturación indican colores más vivos y vibrantes, mientras que valores más bajos se acercan al gris o al blanco.

Valor (V): Representa la luminosidad o brillo del color. Valores más altos de valor indican colores más claros, mientras que valores más bajos se acercan al negro.

Entendiendo los valores R, G y B como valores de intensidad en las capas rojo, verde y azul de la imagen RGB, para cambiar de formato RGB a HSV se emplea `cv2.cvtColor(image, cv2.COLOR_BGR2HSV)` e utilizará el siguiente método para nuestra aplicación [19].

Se puede apreciar el cambio de capas en la figura 1

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases} \quad (3)$$



Fig 1 Imagen en RGB y HSV

Creación de máscaras para selección

Las máscaras de color en una capa HSV (Matiz, Saturación, Valor) se utilizan para seleccionar píxeles o regiones de una imagen que cumplan con ciertos criterios de color en el espacio HSV. Una máscara de color es una imagen binaria en la que los píxeles seleccionados se representan como blancos (valor 255) y los píxeles no seleccionados se representan como negros (valor 0) [20].

Una vez convertida la imagen de RGB a HSV, el uso de una máscara de color permitió realizar selecciones o segmentaciones precisas basadas en propiedades de matiz, saturación y valor de los píxeles. A continuación, se muestran los rangos de elección de píxel, la máscara resultante se puede visualizar en la figura 2.



Fig 2 Imagen filtrada con la máscara de color (imagen)

Uso de filtro de tipo gaussiano

Con el uso de la función cv2.GaussianBlur en Python, del paquete OpenCV, permitió aplicar un suavizado gaussiano a una imagen. Este suavizado se logra convolucionando la imagen con un kernel gaussiano, lo que ayuda a reducir el ruido y los detalles no deseados.

La sintaxis básica de cv2.GaussianBlur es la siguiente:

$$\begin{aligned} & \text{For } i, j, k = \text{len}(\text{image}) \\ & \text{blur } [i, j, k] = \text{image } [i, j, k] * \exp(i+j+k) \\ & \text{end} \end{aligned} \quad (5)$$

Es importante tener en cuenta que al aplicar el suavizado gaussiano, se pierden algunos detalles finos de la imagen, pero se obtiene una versión más suavizada y menos propensa al ruido [21]. La elección adecuada del tamaño del kernel y el valor de sigmaX que para nosotros es 1 depende del tipo de imagen y de los objetivos específicos del procesamiento, el resultado se puede ver en la figura 3.



Fig 3 Máscara suavizada con filtro Gaussiano

Búsqueda de círculos dentro de la imagen

Con el uso de la función cv.HoughCircles en Python, La función cv2.HoughCircles() devuelve una lista de círculos detectados, donde cada círculo se representa como un arreglo de tres valores: las coordenadas x e y del centro del círculo, y el radio del círculo. Posteriormente se iteró sobre esta lista para acceder a cada círculo detectado y realizar acciones adicionales según sea necesario [22].

$$(x - x_{center})^2 + (y - y_{center})^2 = r^2 \quad (6)$$

Es importante tener en cuenta que los resultados de detección de círculos pueden variar según los parámetros utilizados y las características de la imagen de entrada. Debemos ajustar los parámetros que requiere a función y realizar pruebas adicionales para obtener los mejores resultados en la presente investigación.

$$\begin{aligned} & \text{If image } [i, j, k] > \text{rango} \\ & \text{Mask } [i, j, k] = \text{image } [i, j, k] \\ & \text{end} \end{aligned} \quad (4)$$

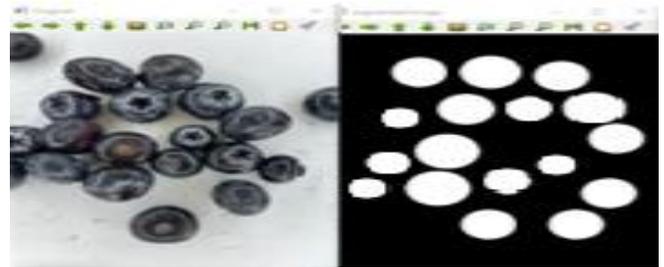


Fig 4 Máscara suavizada con filtro Gaussiano

Detección de bordes

El uso de la función cv2.findContours en Python, del paquete OpenCV, permite encontrar los contornos de una imagen binaria o una máscara. Los contornos son los límites de las regiones o figuras en una imagen.

La sintaxis básica de cv2.findContours es la siguiente [23]:

$$E = \frac{\sum_{i=1}^{I_D} \left(\frac{1}{1 + \alpha d(i)^2} \right)}{\max(I_D, I_I)} \quad (7)$$

Donde:

d(i): distancia entre píxeles

Id: Cantidad de píxeles que detecta el programa

Ii: Cantidad de píxeles reales

Alpha(a): Factor de re escalado

Utilizamos cv.houghcircles en el procesamiento para el enfoque de reconocimiento de formas circulares. Este proceso es iterativo, por lo que los arándanos irán apareciendo uno tras otro como se visualiza en la figura 5.

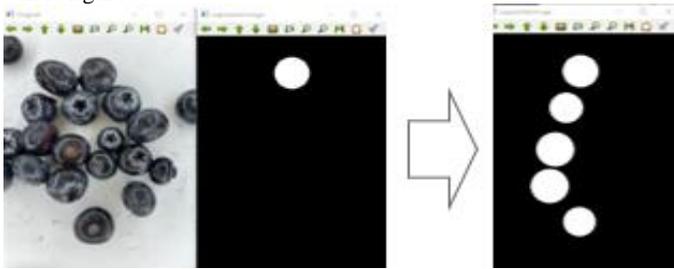


Fig 5 Máscara con ubicaciones ocupadas (izq: imagen, cent y der: blurred_mask)

Búsqueda de contornos y áreas

Se utilizó una iteración de búsqueda de contornos de las formas circulares.

La función cv2.findContours ayudó a encontrar el área utilizada por el Bloom dentro de un círculo de radio de 17px y cv2.area la cual es una función útil para calcular el área de cada uno de los contornos o curvas, proporcionando información cuantitativa sobre la forma y las características de una región en una imagen [24].

Se utilizaron valores aceptables de área Bloom, de forma que solo se filtren los arándanos con la cantidad de Bloom aceptable, los cuales se pueden ver en la figura 6.

```

For i, j, k = len(image)
blur [i, j, k] =image [i, j, k] * mask [i, j, k]
end
    
```

La función cv2.bitwise_and devuelve una nueva imagen que resulta de la operación "y" a nivel de bit entre src1 y src2, aplicada a los píxeles donde la máscara no es cero [25].

Esta función se utilizó para realizar operaciones de máscara, se mostró la región de interés de la imagen original basadas en condiciones lógicas.

Es importante tener en cuenta que las imágenes de entrada deben ser la imagen original y la máscara creada, las cuales tienen el mismo tamaño.

Utilizamos la siguiente fórmula para dibujar los círculos que vemos en la figura 7.

$$(x-h)^2 + (y-k)^2 + (z - l)^2 = r^2 \quad (9)$$



Fig 7 Arándanos que fueron seleccionados

Por lo tanto, en el presente procesamiento se puede visualizar que primeramente se extrajo la imagen en RGB de los arándanos, posteriormente se obtuvo la máscara general en función a las circunferencias obtenidas, luego con el uso de la capa de iluminación de la figura en HSV se logró adquirir cada área ocupada por el arándano en la imagen, con el valor en gris de cada píxel se puede diferenciar quienes están mejor iluminados o con abundante Bloom y solo seleccionar dichos arándanos.

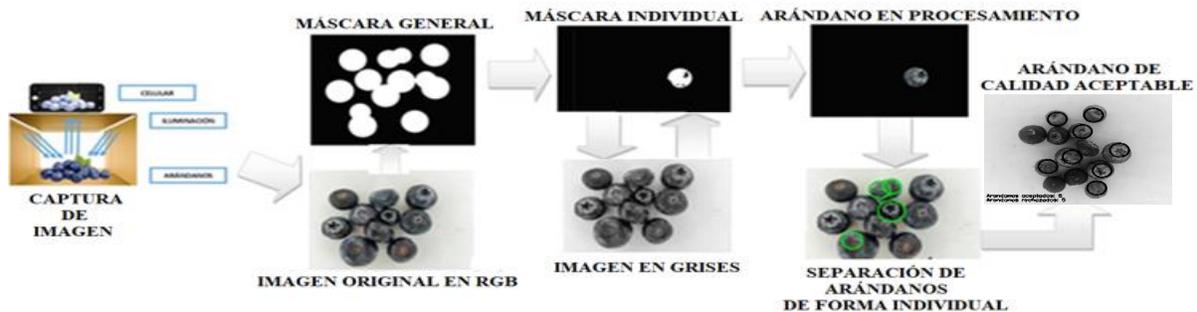


Fig 8 Arquitectura de la visión artificial

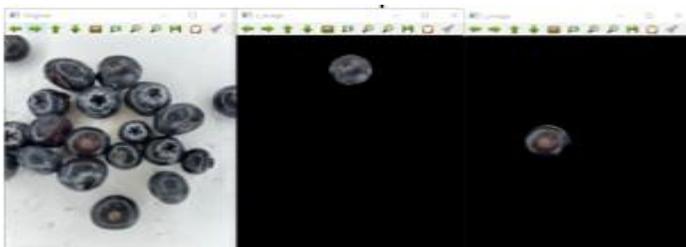


Fig 6 Arándano recortado individualmente uno tras otro

Separación de arándanos en la imagen real

Finalmente, en los arándanos que pasen los valores establecidos, se dibuja una circunferencia sobre dicho arándano, procedimiento que se puede visualizar en la figura 8.

III. RESULTADOS

Se pueden destacar los siguientes resultados favorables:

- Mayor precisión en la detección de arándanos: El enfoque de segmentación utilizando el espacio de color HSV permite una mejor discriminación de los arándanos en comparación con el espacio de color RGB. Esto se debe a que el espacio de color HSV separa la información de color, brillo y saturación, lo que facilita la

identificación de los arándanos en función de su matiz y saturación característicos.

- Segmentación precisa mediante máscaras: La creación de máscaras basadas en las intensidades de HSV favorables proporciona una segmentación más precisa de los arándanos. Al establecer rangos específicos para el matiz, la saturación y el valor, se puede delimitar con mayor precisión los píxeles que corresponden a los arándanos en la imagen.

- Detección de contornos circulares: La búsqueda de contornos circulares después de la segmentación permite identificar de manera efectiva las regiones de arándanos en la imagen. La aproximación de los contornos mediante cv2.area ayuda a simplificar la forma de los contornos, facilitando su análisis y extracción de características.

- Selección de los arándanos encontrados: La selección de los arándanos encontrados en base al área de los contornos detectados según el color del Bloom proporciona una forma eficiente de obtener muestras individuales de arándanos. Esto puede ser útil para análisis posteriores, como la clasificación o el cálculo de características específicas de los arándanos.

La precisión para la visión artificial se basa en predicciones positivas y falsas proporcionadas por la prueba de la programación con imágenes tomadas al producto, y verificando el resultado. Es decir, es la relación entre predicciones positivas del programa (óptimas) y predicciones generales manuales.

Donde:

$$PRECISIÓN = \frac{n(APP)}{n(APM)} * 100 \quad (10)$$

nAPP: Número de pruebas positivas por programa

nAPM: Número de pruebas positivas visualmente

En la tabla 1, se representa la aplicación de la ecuación de precisión a diversas cantidades de muestras de forma ascendente y sus porcentajes de precisión:

TABLA I
RESULTADOS

Nº PRUEBA	ARANDANOS MANUAL		ARANDANOS PHYTON		PRECISIÓN	
	BUENOS	MALOS	BUENOS	MALOS	POSITIVOS	NEGATIVOS
1	9	4	9	4	100	100
2	8	5	8	5	100	100
3	8	5	9	5	87.5	100
4	8	5	9	5	87.5	100
5	9	4	9	4	100	100
6	7	4	7	4	100	100
7	9	4	10	4	88.9	100
8	8	5	8	4	100	80
9	8	5	8	5	100	100
10	8	5	8	4	100	80
11	8	5	8	5	100	100
12	8	5	8	5	100	100
13	8	5	8	5	100	100
14	8	5	8	4	100	80
15	8	5	8	5	100	100
16	8	5	8	5	100	100
17	8	5	8	5	100	100
18	8	5	8	5	100	100
19	8	5	8	5	100	100
20	8	5	8	5	100	100
21	8	5	8	5	100	100
22	8	5	8	5	100	100
23	8	5	8	5	100	100
24	8	5	8	5	100	100
			PROMEDIO		98	98

Lo mismo se puede visualizar en el desarrollo de tendencia en las figuras 9 y 10, donde se representa la tendencia de arándanos reconocidos como buenos y malos.

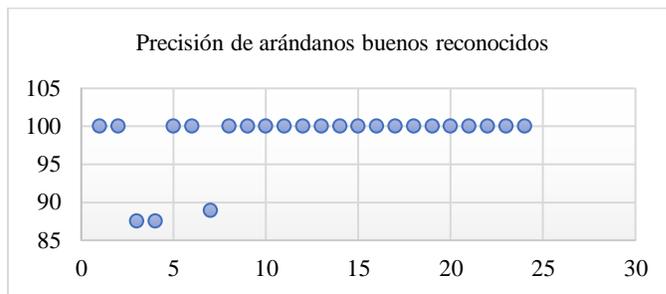


Fig. 9 Distribución de tendencia de precisión de arándanos reconocidos como buenos



Fig. 10 Distribución de tendencia de precisión de arándanos reconocidos como malos

IV. DISCUSIÓN

Mediante la extracción de video se pudo comprobar la efectividad de aplicación de la clasificación del arándano, según la concentración del Bloom. Se menciona, que al aplicar las diferentes muestras el proyecto tiene un porcentaje de precisión de 98% el cual no tiene variación por más que se implemente nuevas muestras en comparación con la investigación.

Según [26] se asignan etiquetas a los defectos en cada arándano. Se genera un archivo en formato *.xml que contiene las coordenadas de cada defecto identificado en cada imagen. Luego, se convierte dicho archivo a un formato *.csv, el cual representa una hoja de cálculo donde se encuentran registradas todas las imágenes junto con sus respectivas coordenadas, tanto para las imágenes de entrenamiento como para las de prueba. Se transforman estos archivos de entrenamiento y prueba en un formato TFRecord y finalmente se configuran los parámetros de la red neuronal. Como precisión obtuvo un 24 al 40.5% en su algoritmo. La creación de su sistema de visión artificial utilizando redes neuronales convolucionales y aplicando el modelo Faster R-CNN se demuestra que tiene menor precisión que el uso de máscaras y filtros con visión artificial usados en el presente trabajo, teniendo una ventaja relevante de 57.5% a favor.

La estrategia aplicada fue diferente ya que, en lugar de introducir a imagen como tal, primeramente, se realiza un preprocesamiento de la imagen con la cual se hacen todas las detecciones anteriormente mencionadas, método que no se realiza en el trabajo anterior.

En concordancia con [6], mencionan que la implementación de la herramienta Phyton logró mejorar el grado de precisión siendo esta mayor a 80% en los indicadores relacionados con la producción y gestión de materiales utilizando la metodología CRISP-DM. En cambio, nuestra codificación tuvo imágenes claras y buena

iluminación, las cuales nos permitió detectar los colores morados del arándano y blancos del Bloom, además de utilizar un fondo blanco que nos permite utilizar un rango de colores mínimo en la capa RGB y capa V de la imagen HSV.

Se tomó en cuenta las investigaciones mencionadas porque son parte del área de calidad, siendo la implementación de la visión artificial en arándanos un proyecto nuevo que beneficiará a su comercialización.

V. CONCLUSIÓN

En conclusión, se implementó un entorno con fondo claro y buena iluminación, que nos permita tomar fotos sin ninguna complicación de los arándanos, utilizando OpenCV se pudo conectar una cámara de alta resolución con Python, de forma que el trabajo sea remoto.

Así mismo, se trabajó en la segmentación de los arándanos utilizando el método de cambio de imagen de RGB a HSV demostró ser efectiva en la discriminación de los arándanos del fondo y otros objetos en la imagen desde la capa V de a imagen en HSV. La creación de máscaras basadas en los valores de matiz, saturación y valor (RGB) permitió resaltar los cuerpos morados y suprimir otros elementos no deseados, la combinación de la máscara de color y la capa V, obtuvimos una imagen con los contornos de los arándanos. Esto condujo a una segmentación más precisa y un mejor aislamiento de los arándanos en las imágenes.

Para la selección únicamente de objetos circulares se utilizó la metodología de la transformada de Hough, la cual nos ayudó en gran medida para analizar cada uno de los arándanos por separado y la necesidad de mejorar la clasificación basada en tamaños relativos de los arándanos en relación con la imagen y la selección de valores en la máscara HSV. Estas conclusiones proporcionan una base sólida para futuras mejoras en el algoritmo de segmentación y clasificación de arándano.

Para la selección de los arándanos buenos, fue importante realizar un ajuste cuidadoso de los valores de área ocupada por el Bloom, para asegurarse de que se evite la inclusión de objetos no deseados en la clasificación.

Finalmente, el informe destaca el éxito en el reconocimiento de arándanos mediante el método de segmentación y la presentación de los arándanos seleccionados como buenos; ya que como resultado se obtuvo una efectividad promedio de 98%, lo cual demuestra que es una alternativa de implementación efectiva para las empresas dedicadas a la agroindustria para evaluar la calidad del arándano en tiempo real, optimizando procesos de selección.

REFERENCIA BIBLIOGRÁFICA

- [1] MONTOYA HOLGUIN, Christian, Jimmy Alexander CORTÉS OSORIO y José Andrés CHAVES OSORIO. *Sistema automático de reconocimiento de frutas basado en visión por computador*. En línea. Ingeniare. Revista chilena de ingeniería, vol. 22 (octubre de 2014), n. °4, pp.504-516. ISSN 0718-3305. Disponible en: <https://doi.org/10.4067/s0718-33052014000400006>.
- [2] JAVIER CORTÉZ, Julio Ernesto. *Diseño de un sistema mecatrónico para seleccionar paltas*. En línea. Master's thesis. Pontificia Universidad Católica del Perú, 2016. Disponible en: <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6739>.
- [3] FONDEVILA RICARDO, Kevin Daniel. *Uso de redes neuronales convolucionales en python para lectura de pruebas rápidas de covid-19*. En línea. Repositorio Universidad de Guayaquil, 2022. Disponible en: <http://repositorio.ug.edu.ec/handle/reduq/65222>.
- [4] GARCÍA VIGIL, Manuel Enrique. *Aplicación de visión por computadora en la clasificación según categorías de calidad de palta (persea americana mill) variedad hass*. En línea. Universidad Privada Antenor Orrego. 2019. Disponible en: <https://repositorio.upao.edu.pe/handle/20.500.12759/5572>.
- [5] CELI CALDERON, Mario Sebastián. *Control de un generador de pulsos, una fuente de voltaje-corriente y matriz por medio del protocolo GPIB programado en Python*. En línea. Universidad San Francisco de Quito. 2013. Disponible en: <https://repositorio.usfq.edu.ec/bitstream/23000/2749/1/108833.pdf>.
- [6] MENDOZA VÁSQUEZ, Eduardo Arturo. *Implementación de herramientas Python en el proceso de producción de cultivos agrícolas del fundo "San Juan de Buenavista"*. En línea. Repositorio de la Universidad César Vallejo. 2021. Disponible en: <https://repositorio.ucv.edu.pe/handle/20.500.12692/59522>.
- [7] ARISTONDO ETXEBERRIA, Jon. *Algoritmo de reconocimiento de forma y color para una plataforma robótica*. En línea. Universidad del país Vasco. 2010. Disponible en: <https://www.ehu.es/documents/1545039/1570316/10jariston-do.pdf>.
- [8] ROS SANTAOLALLA, Iván. *Aplicación Android para el reconocimiento de patrones de color*. En línea. Universidad de Burgos. 2014. Disponible en: <https://riubu.ubu.es/handle/10259/3544>.
- [9] HERRERA, J. C., S. M. MEDINA, K. BELEÑO y O. E. GUALDRÓN. *Diseño de un sistema automático de selección de frutos de café mediante técnicas de visión artificial*. En línea. Revista UIS Ingenierías, vol. 15 (enero de 2016), n.º 1, pp. 7-14. ISSN 2145-8456. Disponible en: <https://doi.org/10.18273/revuin.v15n1-2016001>.
- [10] CONSTANTE PRÓCEL, Patricia Nataly y Andrés Marcelo GORDÓN GARCÉS. *Diseño e implementación de un sistema de visión artificial para clasificación de al menos tres tipos de frutas*. En línea. Repositorio Digital Institucional de la Escuela Politécnica Nacional. 2015. Disponible en: <https://bibdigital.epn.edu.ec/handle/15000/11368>.
- [11] ESCOBAR FIGUEROA, Diego y Edgar ROA GUERRERO. *Sistema de visión artificial para la identificación del estado de madurez de frutas (granadilla)*. En línea. Redes de Ingeniería, vol. 7 (junio de 2016), n.º 1, p. 78. ISSN 2248-762X. Disponible en: <https://doi.org/10.14483/udistrital.jour.redes.2016.1.a08>.
- [12] CRUZ CARRILLO, Henry Omar, Juan Manuel MENESES CHAUS, Martina ECKERT y José Fernán MARTÍNEZ ORTEGA. *Seguimiento automático para RGB y detección de objetos en color*. En línea. Universidad Politécnica de Madrid, Archivo Digital UPM. 2015. Disponible en: <https://oa.upm.es/42294/>.
- [13] ARTOLA MORENO, Álvaro. *Clasificación de imágenes usando redes neuronales convolucionales en Python*. En línea. Universidad de Sevilla, depósito de investigación. 2019. Disponible en: <https://idus.us.es/handle/11441/89506>.

- [14] BALLÓN BENAVENTE, Bruno. *Diseño de un algoritmo de reconocimiento de placas utilizando redes neuronales para optimizar el uso de la Ordenanza Municipal N°927-MPA*. En línea. Universidad Católica de San María. 04/11/2020. Disponible en: <https://repositorio.ucsm.edu.pe/items/0f136c9f-8fde-4a63-b477-644297e9f915>.
- [15] MORAN BARREIRO, Emmanuel Fernando. *Diseño e implementación de una red neuronal convolucional para reconocimiento de productos de una empresa de retail*. En línea. Repositorio DSpace. 2018. Disponible en: <https://www.dspace.espol.edu.ec/xmlui/handle/123456789/47416>.
- [16] GONZALEZ OSORIO, Guadalupe Jonathan. *Reconocimiento de objetos utilizando Open CV y Python en una Raspberry Pi 2 en una tlapalería*. En línea. Universidad Autónoma del Estado de México, repositorio institucional. 10/02/2017. Disponible en: <http://ri.uaemex.mx/handle/20.500.11799/68150>.
- [17] CORNEJO OROSCO, Renato Gianmarco y Aurelio Alejandro DELGADO LUNA. *Diseño de robot agrícola para fumigación automática de malezas en terrenos llanos usando visión inteligente FASTER R-CNN en plataforma Python*. En línea. Universidad Ricardo Palma, repositorio institucional. 2021. Disponible en: <https://repositorio.urp.edu.pe/handle/20.500.14138/4553>.
- [18] TRUJILLO QUEVEDO, Alejandro. *Sistema de reconocimiento de caracteres manuscritos usando redes neuronales convolucionales implementado en Python*. En línea. Universidad de Sevilla. depósito de investigación. 2017. Disponible en: <https://idus.us.es/handle/11441/63838>.
- [19] *OpenCV: changing colorspace*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/3.4/df/d9d/tutorial_py_colorspaces.html.
- [20] *OpenCV: hough circle transform*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/3.4/da/d53/tutorial_py_houghcircles.html.
- [21] *OpenCV: smoothing images*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html.
- [22] *OpenCV: contours: getting started*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html.
- [23] *OpenCV: contour features*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/4.x/dd/d49/tutorial_py_contour_features.html.
- [24] *OpenCV: structural analysis and shape descriptors*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/4.x/d3/dc0/group_imgproc_shape.html#ga2c759ed9f497d4a618048a2f56dc97f1.
- [25] *OpenCV: arithmetic operations on images*. En línea. OpenCV documentation index. [s. f.]. Disponible en: https://docs.opencv.org/3.4/d0/d86/tutorial_py_image_arithmetics.html.
- [26] NARCISO HORNA, Wily Aldo y Edgar Andre MANZANO RAMOS. *Sistema de visión artificial basado en redes neuronales convolucionales para la selección de arándanos según estándares de exportación*. En línea. Campus, vol. 26 (diciembre de 2021), n.º 32, pp. 155–166. ISSN 2523-1820. Disponible en: <https://doi.org/10.24265/campus.2021.v26n32.01>.