

**Desarrollo de un algoritmo de visión artificial para el reconocimiento de la pera de agua en fase óptima en su recolección aplicado en industria****Development of an artificial vision algorithm for the recognition of the water pear in optimal phase in its collection applied in industry****Desenvolvimento de um algoritmo de visão artificial para reconhecimento da pera de água em fase ótima em sua coleta aplicado na indústria**Escobedo Gómez Jesús E.<sup>1</sup>, Cruz Díaz Baby Elder<sup>2</sup>**Resumen**

El objetivo del presente trabajo es desarrollar un algoritmo de detección para el reconocimiento de la pera en su fase óptima para su recolección aplicado a la industria, mediante la utilización de diferentes técnicas de visión artificial y procesamiento de imágenes. Los objetivos específicos son, describir las técnicas de visión artificial y procesamiento de brillo, además de emplear programas para el desarrollo del algoritmo. El reconocimiento de la pera se realiza en tiempo real, utilizando el brillo adecuado y con equipos tecnológicos. El algoritmo reconoció a la pera en su fase óptima para la recolección en un 98% y con un tiempo de detección promedio de 7.63 min. Esta presente investigación pretende sumar evidencia con respecto a los anteriores trabajos de investigación que desarrollaron algoritmos de detección para alguna fruta, además de rescatar información para futuras investigaciones que pretendan mejorar los procesos industriales en empresas las cuales tengan como producto final la importación o exportación de frutas.

**Palabras claves:** Algoritmo, Visión Artificial, Procesamiento de brillo.

**Abstract**

The objective of this work is to develop a detection algorithm for the recognition of the pear in its optimal phase for harvesting applied to the industry, through the use of different artificial vision and image processing techniques. The specific objectives are to describe artificial vision and brightness processing techniques, in addition to using programs for the development of the algorithm. Pear recognition is carried out in real time, using the appropriate brightness and technological equipment. The algorithm recognized the pear in its optimal phase for harvesting by 98% and with an average detection time of 7.63 min. This research aims to add evidence with respect to previous research works that developed detection algorithms for some fruit, in addition to rescuing information for future research that aims to improve industrial processes in companies whose final product is the import or export of fruits.

**Keywords:** Algorithm, Computer Vision, Gloss Processing.

**Resumo**

O objetivo deste trabalho é desenvolver um algoritmo de detecção para reconhecimento da pêra em sua fase ótima de colheita aplicado à indústria, através da utilização de diferentes técnicas de visão artificial e processamento de imagem. Os objetivos específicos são descrever técnicas de visão artificial e processamento de brilho, além de utilizar programas para o desenvolvimento do algoritmo. O reconhecimento da pêra é realizado em tempo real, utilizando brilho e equipamentos tecnológicos adequados. O algoritmo reconheceu a pêra em sua fase ideal para colheita em 98% e com tempo médio de detecção de 7,63 min. Esta pesquisa visa agregar evidências a respeito de pesquisas anteriores que desenvolveram algoritmos de detecção para algumas frutas, além de resgatar informações para futuras pesquisas que visem melhorar processos industriais em empresas cujo produto final seja a importação ou exportação de frutas.

**Palavras-chave:** Algoritmo, Visão Computacional, Processamento de Brilho.

<sup>1</sup> Escuela de Ingeniería Industrial. Estudiante. Universidad Privada del Norte. Trujillo. Perú. [N00218816@upn.pe](mailto:N00218816@upn.pe). <https://orcid.org/0009-0000-5746-9449>

<sup>2</sup> Escuela de Ingeniería Industrial. Magister. Universidad Católica de Trujillo. Trujillo. Perú. [b.cruz@uct.edu.pe](mailto:b.cruz@uct.edu.pe). <https://orcid.org/0009-0008-0229-4264>

## **Introducción**

La investigación realizada en una empresa de exportación de uvas, tuvo por finalidad aumentar la productividad ya que no se contó con una eficiente gestión de inventarios, por lo que existió diferentes problemas, los principales son: la rotura de stock de productos de alta rotación y la falta de planificación de las compras, porque no hubo un manejo adecuado en el inventario en la cantidad y tiempo de pedido, esto causó que no se cuente con la cantidad suficiente para abastecer al área de producción; el deficiente control de calidad de los productos, ya que el personal no se encontró capacitado para el proceso de empaque de las uvas porque la mayoría de los trabajadores realizaban sus labores de manera empírica; el personal se demoraba debido a la dificultad en el manejo de la máquina que arma las cajas de madera y cartón, por lo que ocurrió retrasos en la producción, demoras más del tiempo estimado y que la producción del armado de cajas no haya sido el esperado y así se generó desperdicios de la uva fresca ya que la producción programada para exportación ya no fue útil.

El estado de las frutas es de vital importancia para su recolección final, debido a esto, las empresas que trabajan con estos tipos de productos suelen tener ciertos estándares de calidad que el ojo humano a simple vista no puede observar, he aquí la necesidad de la visión artificial que ayuden al operador al cumplimiento de todos los parámetros que requiere los productos de primera necesidad como son las frutas. Desde hace algún tiempo las empresas industriales están modernizando sus procesos, implementando tecnología basa en la visión por computadora, debido a que muchas empresas que exportan e importan productos alimenticios que requieren estándares de calidad de mayor exigencia que otros productos (Atencio y Sánchez, 2009). El factor más frecuente que el consumidor puede utilizar para determinar si compra o no los productos es el aspecto visual de este, causante de ello, el color de una fruta es un buen parámetro visual al evaluar si el estado o fase de maduración es el adecuado para su compra y finalmente para su consumo (Padron, 2016), esto genera que las industrias agrícolas que exportan e importan sus productos por todo el mundo, prioricen en tener una gestión de calidad en todos sus procesos que intervengan para el producto final, al no hacerlo presentarán una baja competitividad hacia el mercado global y con riesgo de no tener esa aceptación por el consumidor final. Debido a esto, llevar un buen control de calidad es de mucha importancia, sin embargo, esta actividad aún no está automatizada en todos sus procesos de producción, debió a ello, es el operario el que se sigue encargado de tomar la decisión, y esta puede variar y ser errónea (La Madrid, 2019).

Por un lado, el procesamiento digital de imágenes es utilizada en procesos internos por las grandes empresas, debido a la fuerte suma de inversión, esta tecnología trata de un conjunto de procesos que resalta información que contiene una imagen, utilizando como principal herramienta la computadora (Dominguez, 1996). El aumento de la utilización de procesamiento digital de imágenes en la industria se dio gracias a la necesidad de procesos más estandarizados, niveles de producción más altos y productos como mayores niveles de calidad, aunque la adquisición de esta técnica de procesamiento de imágenes se da a inicios del año 1940, en las industrias alimentarias se difunde con posterioridad (González, 2021), la inspección visual de productos utilizando algún algoritmo inteligente, requiere de un mayor desarrollo a las distintas dificultades que conlleva o puede presentar trabajar en tiempo real (Navarro y Reta, 2020). El análisis digital de imágenes tiene como tarea extraer mediciones, datos o información contenida en una imagen, por lo que el sistema se distingue al tener un parámetro de entrada una imagen, y cuyo resultado es una salida numérica, en lugar de otra imagen (Guadalupe, 2020), una imagen digital es una imagen función  $f(x, y)$ , y puede ser considerada como una matriz cuyos índices de renglón y columna identifican un punto en la imagen que sería un lugar bidimensional en el espacio. (Escalante, 2006)

Por otro lado, el color juega un papel muy para determinar en qué estado de madurez para su recolección se encuentra la fruta, se define como una percepción humana (Conde y Báez, 2000), en la maduración de la pera de agua y en otras frutas ocurren cambios de color en su exocarpio, que va desde el verde al amarillo, los cuales son provocados en los organelos celulares, la cual en su interior está compuestos de pigmentos como las flavonas, los carotenoides y las antocianinas, estos son los que sufren un anabolismo y generan al fruto colores secundarios sobre la base del color que debe predominar (Castro, Cerquera, y Gutiérrez, 2013), a esto se le conoce como síntesis o degradación de pigmentos, a diferencia de otras frutas, la temperatura que necesitan algunas variedades de peras para alcanzar una fase óptima de recolección y posteriormente para su consumo, está entre los  $-1.0$  y  $-0.5$  °C (Delfilippi y Torres, 2015).

Existen una multitud de lenguajes de programación, Python es un lenguaje que lleva a cabo el tratamiento de las imágenes, y utiliza diferentes librerías como NumPy y Matplotlib, este lenguaje de programación está presente en muchos sistemas operativos, como iOS, Windows, Linux, Mac o Android, debido a que tiene una sintaxis fácil de leer con una curva de aprendizaje moderada y con un código legible (Fernández, 2020). Por otro lado, Open CV (Open Source Computer Vision), es una librería desarrollada por Intel utilizada para el procesamiento de imágenes y la visión artificial en general, contiene funciones que se utiliza en la visión por computadora, como es la inspección de productos, personas u objetos en movimiento (Viera, 2017).

En la investigación de (Delgado, Herrera, Medina y Corredor, 2017) propuso una implementación de un sistema de procesamiento de imágenes utilizando un pequeño ordenador llamado Raspberry PI 2B, con el objetivo de describir el método para localizar fresas y determinar su madurez. Se usó herramientas como el algoritmo Python, adicional se utilizó las librerías de Open CV y un robot recolector de fresas. Tuvo como resultados la identificación de todas las fresas y el reconocimiento de la madurez en un 76.6%.

Así mismo (Moreno, Trejo, Lizárraga, Cabal y Carrillo, 2017) sugirió la aplicación de procesamiento de imágenes para determinar en qué estado de madurez se encuentra una cereza, utilizando un software llamado Matlab en su versión R2013a con el toolbox para el procesamiento de imágenes y capturas de la cereza en distintas etapas de maduración. Tuvo como resultados que la metodología propuesta para segmentar las imágenes fue de alta efectividad con un tiempo de 4.30 segundos, logrando así identificar una cereza en cierto grupo sin la ayuda de un ser humano.

De tal forma (Castro, Cerquera y Gutiérrez, 2013), propuso la utilización de técnicas de procesamiento digital de imágenes para determinar el color del exocarpio como indicador del desarrollo fisiológico de la guayaba pera, con el objetivo de desarrollar el análisis de coordenadas colorimétricas de imágenes digitales en tres tipos de cosecha y con una temperatura de almacenamiento refrigerado de 7,5 °C y humedad relativa del 85%. Además, para determinar el color se utilizó espacios de color RGB y CIE-L \*A \*B. Los resultados arrojaron una evolución desde el verde intenso, recién recolectado, hacia verdes claros y amarillosos, de manera que se puede determinar la madurez de la fruta y apoyar en las etapas de cosecha y postcosecha de frutas.

El objetivo de este artículo es desarrollar un algoritmo de visión artificial para el reconocimiento de la pera de agua en fase optima en su recolección aplicado en industria, con la utilización de técnicas de visión artificial y de procesamiento de imágenes. Los objetivos específicos para el cumplimiento del general son, describir las técnicas empleadas como las técnicas de clasificación en cascada Haar y el procesamiento de brillo, además de emplear el lenguaje de Python para el desarrollo de algoritmo, el editor de código Visual Studio Code y el programa Cascade Trainer GUI, el cual nos sirve para poder testear y encontrar los parámetros óptimos en la imagen y de encargarse de realizar el proceso de clasificación en cascada Haar.

## **Materiales y Métodos**

Los materiales utilizados para el desarrollo del proyecto son; una laptop con la cual se utilizó para el desarrollo y funcionamiento del algoritmo es de la marca Lenovo con un procesador i5 de novena generación, además se usó la web cam que tiene integrada este ordenador para las tomas de capturas de imágenes positivas y negativas. Dichos materiales pertenecen a uno de los integrantes del proyecto. Por otro lado, el software empleado para la ejecución de nuestra red neuronal es el programa Python con la versión 3.11.3 y su Librería Open CV de manera gratuita, asimismo el editor de código compatible con Python empleado en nuestro proyecto es el Visual Studio code. Por último, se aprovechó el interfaz de Cascade Trainer GUI para el entrenamiento de las imágenes positivas y negativas.

En la Figura 1 se observa la elaboración de un esquema en 2D, la cual representa la preparación del objeto (pera) con su respectiva iluminación la cual será una luz led y la cámara para realizar la captura de una manera más profesional.

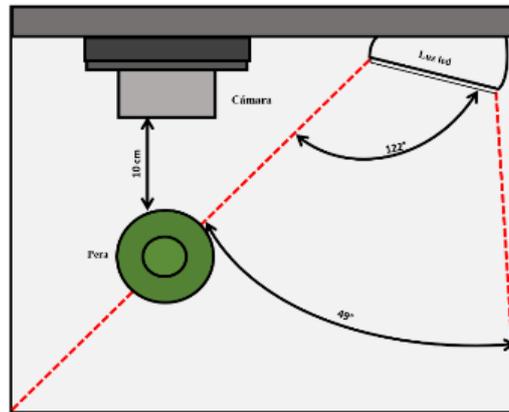


Figura 1. Esquema para la captura de imágenes.

Realizaron un estudio sobre las características físicas y fisiológicas de la pera, explicando que la fase óptima para su recolección ronda los 179 días de cultivo, con un color verdoso, un pH promediando los 4,16 y un peso de 306,30 g. Una vez pasado los días recomendados para su recolección, dichos valores comienzan a cambiar y perjudican el estado de composición de la fruta. Por lo cual, siguiendo el estudio se determinaron fases con respecto a los días de cosecha que se observa en la figura 2. Considerado que la recolección óptima debe ser en la fase de 179 días (Etapa 0) de la cosecha, por otro lado, si se recoge en las demás etapas podría llegar a descomponerse y se considera como un pera en malas condiciones.

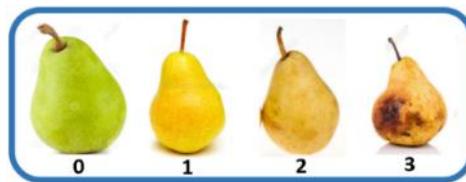


Figura 2. Etapas de maduración en la pera.

Para el correcto funcionamiento del programa, se debe importar las librerías necesarias para la ejecución del código, este se verá mostrado en el entorno de desarrollo de programa visual studio code que nos ayudará a editar nuestro código de manera rápida y con un interfaz ligero pero eficaz. Numpy es una librería matemática muy utilizada en el procesamiento de imágenes, dicha librería nos permitirá trabajar con matrices (N-Array) y trabajar con grandes cantidades de matrices multidimensionales, además cuenta con diferentes funciones para el procesamiento de imágenes (Bauckhage, 2015). También se utilizará la librería OS, esta librería nos permite acceder a muchas funcionalidades dependiente si se encuentran o no en el sistema operativo empleado (Python, s.f.). Luego está la librería Imutils que es un conjunto de diferentes funciones que sirven para el procesamiento de imágenes como por ejemplo en el cambio de tamaño, clasificación de contornos, rotación, entre otras (Chuquimarca, Pinzón y Rosales, 2021). Por último, importaremos la librería de OpenCV, dicha librería está dirigida directamente a la visión por computadora en tiempo real, debido a que contiene estructuras de datos, procesamiento y análisis de imágenes estructural (Arévalo, González y Ambrosio, 2004). Para entrenar nuestra red neuronal se requiere de la utilización del software Cascade Trainer GUI, es una herramienta complementaria para usar OpenCV, este programa cuenta con una interfaz gráfica muy fácil de intuir para el usuario (AMIN, 2016).

Las librerías que se mencionó anteriormente, desarrollan un papel muy importante en la creación de nuestra red neuronal, la librería Numpy en nuestro trabajo almacenará y accederá a los datos de un N-array, las matrices multidimensionales utilizadas tendrán el mismo dato para todos sus elementos, la librería Imutils nos ayudará con la visualización de la imagen y además con diferentes funciones para procesar las fotos negativas y positivas. La librería OS, en nuestro código nos permitirá redactar, leer y manipular rutas del destino de nuestro archivo y a la vez crear carpetas temporales. Por otro lado, programa Cascade Trainer GUI cumple la función de entrenar, probar y mejorar nuestro modelo de clasificación en cascada para las imágenes positivas y negativas.

Una vez importada las librerías a utilizar, se desarrollará la preparación de la muestra. El programa está

dividido en dos ventanas, llamadas “entrenamiento.py” y “detectando.py”, esta primera ventana tendrá un código que se encargará de tomar las imágenes positivas (pera apta) y negativas (peras no aptas), para posteriormente ser entrenadas. Para el entrenamiento se necesitará contar con una gran cantidad de imágenes ya sea cientos o miles, de esta manera al tener mayor cantidad de imágenes será mejor para la determinación de la pera ya sea apta o no apta.

El contenido del primer código iniciará con la creación de las carpetas “p” y “n” donde se almacenarán las capturas mencionadas anteriormente con la variable “Datos” como se observa en la Figura 3.

```
import cv2
import numpy as np
import os
import imutils

Datos = "n"
if not os.path.exists(Datos):
    print('Carpeta creada: ', Datos)
    os.makedirs(Datos)
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
```

Figura 3. Código de la toma de imágenes y creación de las carpetas.

Las carpetas creadas por el código contienen un conjunto de muestras positivas donde se visualice a la pera en buen estado, y otras muestras negativas donde no esté presente la pera o esté presente una pera en mal estado, dichas muestras no deben ser muy grandes, debido a que pueden provocar lentitud en la detección. En las figuras 4 y 5 se muestran las carpetas creadas con sus respectivas muestras.

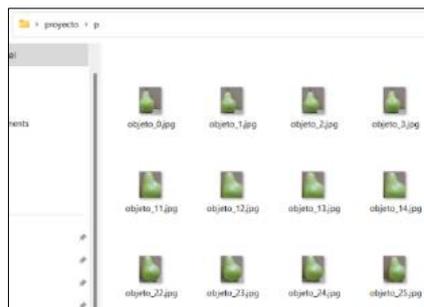


Figura 4. Carpeta con muestra positivas

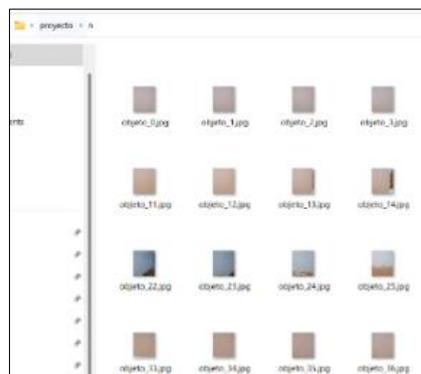


Figura 5. Carpeta con muestra negativas

Luego de tomar y guardar las muestras, se tomó en cuenta implementar la técnica de procesamiento de brillo, para calcular el brillo general se tuvo que convertir las imágenes del formato RGB a escalas de grises y se utilizó las siguientes ecuaciones matemáticas. La Ecuación (1) se observa que las variables N y M deben ser las dimensiones de ancho y alto de las imágenes tomadas anteriormente, P sería el valor del píxel que está situado en una posición cartesiana (i, j) de la imagen. Esta ecuación es utilizada en la probabilística cuando se quiere sumar todos los valores recorriendo el ancho y alto de una imagen y dividiéndolo entre la cantidad total de píxeles para luego encontrar un valor promedio (Delgado, Herrera, Medina, y Corredor, 2017).

$$B = \frac{1}{NM} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} P(i,j) \quad (1)$$

Por último, se tuvo que calcular un valor  $B_{sum}$ , restando el brillo deseado  $B_{des}$  y el brillo obtenido de la imagen a escala de grises [26]  $B$  como se aprecia en la Ecuación (2).

$$B_{sum} = B_{des} - B \quad (2)$$

Como se había explicado, imagen para a convertirse en escala de grises como se observa en la Figura 6.

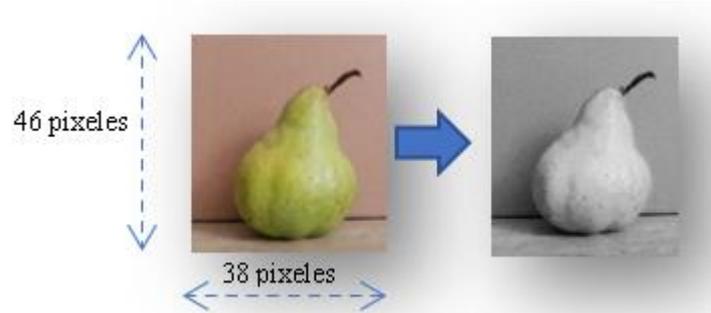


Figura 6. Conversión de la imagen a escala de grises.

En el algoritmo, se agregó todos estos resultados como se aprecia en la Figura 7.

```
gray = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)

objeto = peraClasificador.detectMultiScale
(gray, scaleFactor=6, minNeighbors=900,
minSize=(70, 78))
```

Figura 7. Código de escala de grises.

Para el entrenamiento de la muestra se utilizó la técnica denomina clasificador en cascada, esta técnica se basa en funciones de Haar, este es un método especializado en la detección de objetos (OpenCV, 2004), definida por Viola y Jones (Viola y Jones, 2001) y tiende a fundamentarse en un aprendizaje automatizado, donde la función cascada es entrenada por muchas imágenes positiva y negativas (Unipython, 2018). Se extrae rasgos de la imagen y se utiliza las características de Haar y organiza la imagen en diferentes features como se observa en la Figura 8, estos también llamados filtros generan diferencia en la intensidad de la imagen y generan distintas características (Guevara, Echeverry y Ardilla, 2008).

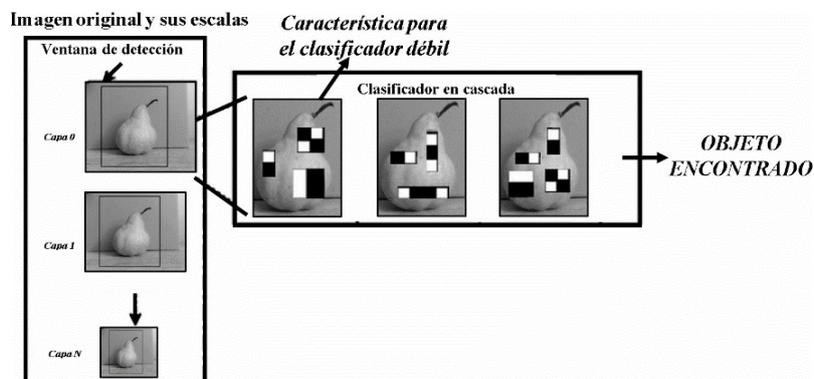


Figura 8. Algoritmo basado en cascada Haar.

Una vez almacenadas nuestra muestra de imágenes positivas y negativas en sus respectivas carpetas

creadas, se utilizará la herramienta Cascade Trainer GUI, la cual nos permitirá testear y dar mejoras a nuestro modelo de clasificación de cascada y se encargará de llevar todo el proceso de clasificación en cascada Haar. Básicamente proporciona una interfaz fácil de entender, además de proveer de un set de datos para las imágenes correspondientes, a partir de ello se obtiene como salida un modelo de cascada en formato XML. (Jeremías, 2020)

Luego de finalizar el entrenamiento, se procede a crear y ejecutar el código llamado “detectando.py”, aquí se abrirá la cámara que detectará la pera apta y dejará de lado la pera no apta, así mostrará el resultado en tiempo real. En la Figura 9 se muestra la arquitectura para la creación y aplicación de algoritmo.

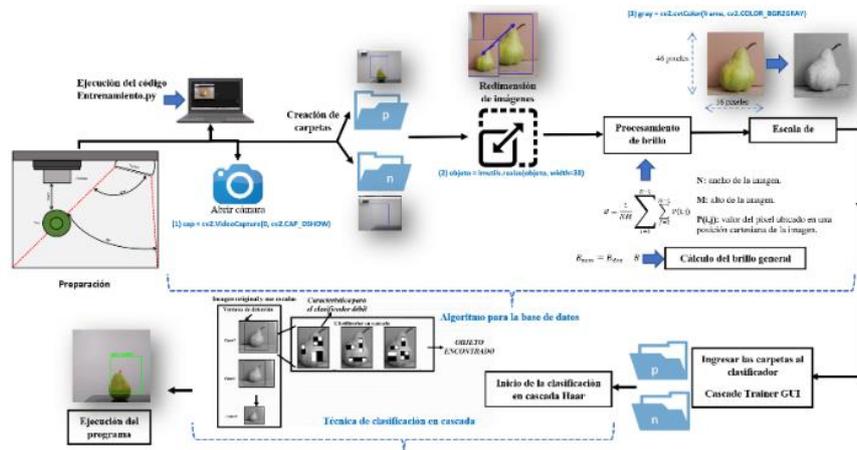


Figura 9. Arquitectura de la creación y aplicación del algoritmo.

## Resultados

La prueba se realizó con imágenes positivas e imágenes negativas. A cada una de estas imágenes se le aplicó las diferentes técnicas para el mejoramiento del brillo general y calidad.

**Tabla 1**  
Escala de brillo general para las imágenes

FORMATO	ECUACIÓN	ESCALA DE BRILLO
Imágenes positivas	$B_{sum} = B_{des} - B$	(70 - 78)
Imágenes negativas	$B_{sum} = B_{des} - B$	(42 - 50)

En la Tabla 1 se puede observar la escala para el brillo general en las imágenes positivas tiene a ser de 70 – 78, con lo cual permitirá una mayor calidad en el píxel encontrado y una mejor clasificación en Haar cascada. Por otro lado, para las imágenes negativas la escala de brillo es de 42 – 50, lo que significa una menor calidad de imagen.

Para encontrar la precisión del algoritmo se realizaron pruebas para el reconocimiento de la pera en buen estado, registrando la cantidad peras, intentos y aciertos observados en la Tabla 2.

**Tabla 2**  
Registro de intentos

Cantidad de imágenes positivas	Intentos	Aciertos	Efectividad (%)
20	100	94	94
40	100	95	95
50	100	98	98
60	100	98	98
80	100	98	98

Por último, en la Tabla 3 se registró el tiempo de detección promedio en 50 intentos para posteriormente

determinar la eficiencia promedio del programa dividiendo el tiempo de detección con el tiempo de trabajo como se muestra en la Tabla 4.

**Tabla 3**  
*Tiempo de detección*

Item	Intentos	Tiempo de detección promedio (min)
Pera óptima	50	7.63

**Tabla 4**  
*Eficiencia del programa*

Item	Ecuación	Eficiencia promedio (%)
Pera óptima	$eficiencia = \frac{tiempo\ de\ detección}{tiempo\ de\ trabajo}$	97

## Discusión

Al utilizar la técnica de procesamiento de brillo en las imágenes positivas y negativas, se pudo determinar la escala de brillo de 70 – 78 para las imágenes positivas y un rango de 42 – 50 para las imágenes negativas. Esto ayudó a mejorar la calidad y aumentar la efectividad del algoritmo. En el trabajo de (Delgado, Herrera, Medina y Corredor, 2017), al implementar un preprocesamiento de brillo, pudo reducir o aumentar valores en la imagen, mejorando la calidad y rango de colores de las fotos donde se aprecia el fruto.

Para poder encontrar el porcentaje de efectividad del algoritmo, se hicieron pruebas, registrando las cantidades de imágenes positivas donde se encuentra la pera óptima con sus respectivos intentos y aciertos, el resultado fue que para una cantidad de 50 imágenes positivas la efectividad en aciertos era del 98% y si se seguía aumentando las imágenes este porcentaje no aumentaba, dando así un 98% de efectividad final con 50 imágenes. Además, el tiempo de detección promedio fue de 7.63 ms, lo que indica que el algoritmo es confiable y desarrolla su ejecución de manera rápida. En el trabajo de (Moreno, Trejo, Lizárraga, Cabal y Carrillo, 2017), se segmentó en dos ítems, el primero fue el tiempo de segmentación por imagen arrojando un tiempo de 4.3048 segundos y el segundo fue el porcentaje de efectividad, donde de 407 imágenes. 396 fueron correctas y 11 incorrectas, obteniendo como resultado una efectividad del 97.30%. Esta diferencia se debió a la utilización de un programa que nos ayuda a testear parámetros y se encarga de realizar el proceso de clasificación en cascada Haar, mientras el estudio que obtuvo una efectividad del 97.30%

Este estudio suma evidencia adicional respecto a los anteriores artículos de investigación, donde utilizan y se estudia técnicas de procesamiento de imágenes y visión artificial, ya sea para reconocer objetos o en el estado en que se encuentra diferentes frutos. Estos resultados ayudan al desarrollo de la investigación con respecto al manejo de algoritmos tecnológicos que puedan mejorar los procesos en las diferentes industrias.

## Conclusiones

Se logró desarrollar un algoritmo de detección para el reconocimiento de la pera de agua en su fase óptima de recolección, a partir de las diferentes técnicas de procesamiento de imágenes como el procesamiento de brillo, la conversión a escala de grises y la redimensión de imágenes. Por otro lado, se implementó la metodología clasificación en cascada con base en Haar como la técnica de visión artificial en tiempo real.

Se utilizó programas como Python y sus diferentes librerías para la aplicación de Haar cascade, además de un editor de código como es el Visual Studio Code para la creación y ejecución del algoritmo, y, por

último, el programa Cascade Trainer GUI, que permite testear valores y mejorar el modelo de clasificación en cascada.

Finalmente, se demostró que el algoritmo funciona de manera rápida y efectiva, obteniendo como efectividad un 98% y un tiempo de detección de 7.63 ms, esto gracias a las técnicas aplicadas a las imágenes.

El algoritmo puede ser aplicado en las diferentes industrias para la detección de la pera de agua, solo hay que tener en cuenta los ajustes respectivos del ambiente de iluminación donde se realizaría la detección.

## Referencias

- AMIN. (2016). Cascade Trainer GUI. Obtenido de <https://amin-ahmadi.com/cascade-trainer-gui/>
- Arévalo, V., González, J., & Ambrosio, G. (2004). La librería de visión artificial opencv. aplicación a la docencia e investigación. Base Informática, 40, 61-66. Disponible en: [Microsoft Word - open-source.rtf \(uma.es\)](#)
- Atencio, P. S., & Sánchez, G. (2009). Enfoque geométrico para la estimación del peso del fruto mango de azúcar mediante procesamiento digital de imágenes. *Avances en Sistemas e Informática*, 6(3), 111-118.
- Bauchhage, C. (2015). "NumPy / SciPy Recipes for Image Processing: Creating Fractal Images", researchgate.net, Feb. 2015, <https://dx.doi.org/10.13140/2.1.2975.5685>.
- Castro, J., Cerquera, N., & Gutiérrez, N. (2013). Determinación del color del exocarpio como indicador de desarrollo fisiológico y madurez en la guayaba pera utilizando técnicas de procesamiento digital de imágenes. *EIA*, 10(19), 79-89.
- Chuquimarca, L., Pinzón, S., & Rosales, A. (2021). Detección de Mascarilla para COVID-19 a través de Aprendizaje Profundo usando OpenCV y Cascade Trainer GUI. *Revista Científica y Tecnológica UPSE*, 8 (1) pág. 68-73. DOI: 10.26423/rctu.v8i1.572.
- Conde, J., & Báez, J. (2000). Conversión de un conjunto de imágenes multiespectrales a un sistema RGB. *Revista Mexicana de Física*, 46(1), 45-51.
- Delfilippi, B., & Torres, C. (2015). Maduración en Peras. Calidad y condición en postcosecha. *Boletín Técnico POMACEAS*, 15(1), 2-8.
- Delgado, M., Herrera, D., Medina, L., & Corredor, J. (2017). Implementación de un sistema de procesamiento de imágenes integrado con Raspberry PI 2B para reconocimiento y recolección de fresas maduras. *Revista Politécnica*, 13(25), 75-85.
- Dominguez, A. (1996). Procesamiento digital de imágenes. Distrito Federal: Perfiles Educativos.
- Escalante, B. (2006). Procesamiento Digital de Imágenes.
- Fernández, S. (2020). Herramienta de Reconocimiento de Imágenes en Python. Universidad de Sevilla.
- González, M. (2021). Técnicas de procesamiento de imágenes aplicadas al monitoreo de procesos alimentarios. Tesis de maestría. Universidad de la República (Uruguay). Facultad de Ingeniería.
- Guadalupe, L. (2020). Procesamiento digital de imágenes. instituto tecnológico de Mérida.
- Guevara, M. L., Echeverry, J. D., & Urueña, W. A. (2008). Detección de rostros en imágenes digitales usando clasificadores en cascada. *Scientia et Technica*, XIV(38),1-5. ISSN: 0122-1701. Recuperado de: <https://www.redalyc.org/articulo.oa?id=84903801>
- Jeremías, E. (2020). Reconocimiento de objetos a través de la metodología Haar Cascades. *Artículos presentados a RADI. Tecnología de la información y comunicación*.
- La Madrid, L. (2019). Implementación de un algoritmo de control de calidad para la selección de productos agrícolas utilizando visión artificial. Universidad Nacional Pedro Ruiz Gallo.

Moreno, W., Trejo, M., Lizárraga, R., Cabal, E., & Carrillo, L. (2017). Determinación del estado de madurez de una cereza aplicando procesamiento de imágenes. *Jóvenes en la Ciencia*, 3(2), 2686-2689.

Navarro, J., & Reta, C. (2020). Procesamiento digital de imágenes en la creación de prototipo de sistema inteligente para detección y reconocimiento de texto en etiquetas de conductores eléctricos recubiertos. *Latin - American Journal of Computing*, 7(2), 94-105.

OpenCV. (2004). Clasificador en cascada. Disponible de [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html).

Padron, C. (2016). Procesamiento digital de imágenes. En *Determinación del color en muestras de alimentos y durante la maduración de frutos* (págs. 37-38). Valencia, Carabobo: Carlos Padron.

Python (s/f). <https://docs.python.org/es/3.10/library/os.html>.

Unipython. (2018). Detección de rostros, caras y ojos con Haar cascad. Obtenido de <https://unipython.com/deteccion-rostros-caras-ojos-Haar-cascad/>.

Viera, G. (2017). Procesamiento de Imágenes usando OpenCV aplicado en Raspberry Pi para la Clasificación del cacao. Universidad de Piura.

Viola, P., & Jones, N. (2001). Detección rápida de objetos utilizando una cascada potenciada de funciones simples. Obtenido de <https://ieeexplore.ieee.org/document/990517>.