

Artículo Original

Simulación de un sistema de control de calidad de galletas usando visión por computadora.

Oscar Héctor Fernández Díaz



¹Universidad Nacional del Este. Facultad Politécnica.
Ciudad del Este, Paraguay.

<https://orcid.org/0009-0004-6769-3523>

Autor correspondiente: hectlp1@gmail.com

Para citar este artículo:

Fernández Díaz, O. H. (2024). Simulación de un sistema de control de calidad de galletas usando visión por computadora. *UCOM Scientia*, 2(1), 46-62.

Fecha de recepción: 14/12/2023

Fecha de aceptación: 01/02/2024

Resumen

La implementación de sistemas de control en la industria alimenticia es esencial para garantizar la satisfacción del cliente, reducir costos, cumplir con regulaciones y normativas, mejorar la competitividad y optimizar los procesos de producción.

Los sistemas de control de calidad basados en sensores de visión permiten obtener un control de las características físicas sin entrar en contacto con el producto, permitiendo tener un proceso más higiénico. En este trabajo, se construyó una estación de control de calidad en el simulador CoppeliaSim. Esta estación cuenta con sensores de visión y de proximidad, se agregó el modelo 3D de la galleta, a la cual se realizaron variaciones para simular defectos. Las imágenes captadas del simulador son enviadas a MATLAB, en donde se desarrolló un código capaz de extraer las características de interés para el control de calidad de las galletas, que son: área, circularidad, color y cantidad de chocolate. En el procesamiento de las imágenes se utilizaron técnicas de segmentación por color, operaciones morfológicas, etiquetado, extracción de propiedades de objetos y análisis de canales de color. Las galletas son mostradas en una interfaz gráfica indicando si cumplieron o no con los requisitos del proceso con algunos parámetros de interés, además, los datos de las galletas que no cumplen con los requisitos son almacenados en una variable. El sistema presenta una velocidad de inspección de 8 galletas por segundo, con una eficacia del 93 % con un grupo de 80 galletas.

Palabras clave: control de calidad, visión por computadora, galletas, Matlab, CoppeliaSim.

Original Article

Simulation of a cookie quality control system using computer vision.

Abstract

The implementation of control systems in the food industry is essential to guarantee customer satisfaction, reduce costs, comply with regulations and standards, improve competitiveness and optimize production processes. Quality control systems based on vision sensors allow control of the physical characteristics without coming into contact with the product, allowing a more hygienic process. In this work, a quality control station was built in the CoppeliaSim simulator. This station has vision and proximity sensors, the 3D model of the cookie was added, to which variations were made to simulate defects. The images captured from the simulator are sent to MATLAB, where a code capable of extracting the characteristics of interest for the quality control of the cookies, which are area, circularity, color and quantity of chocolate, was developed. Color segmentation techniques, morphological operations, labeling, object property extraction, and color channel analysis were used to process the images. The cookies are displayed in a graphical interface indicating whether or not they met the process requirements with some parameters of interest, in addition, the data of the cookies that do not meet the requirements are stored in a variable. The system has an inspection speed of 8 cookies per second, with an efficiency of 93% with a group of 80 cookies.

Keywords: quality control, computer vision, cookies, Matlab, CoppeliaSim

1. Introducción

Las galletas durante el proceso de cocción sufren variaciones en el tamaño, forma y color (Čačić et. al., 2014), antes de que estas galletas pasen a la máquina empaquetadora se deben verificar que no estén quemadas o sean demasiado grandes o pequeñas porque pueden dañar la maquinaria, creando un mayor problema en la línea de producción (Davies, 2012). Los productos alimenticios como las galletas se producen en grandes cantidades al día, lo cual una rápida inspección se vuelve esencial. La decisión humana en el control de calidad es muy variable, costosa y requiere mucho tiempo. En estos casos se recomienda implementar un sistema automatizado capaz de realizar el control de calidad, sin embargo, antes de realizar el proceso real es importante que el sistema sea probado en un simulador para disminuir errores y pérdidas.

El objetivo de este trabajo es el de simular un sistema de control de calidad de galletas, específicamente elaborar una estación de control de calidad en el simulador CoppeliaSim y agregar el modelo de la galleta a controlar, realizar un programa para la inspección de las galletas en Matlab y evaluar el sistema. Se evalúa el sistema en términos de velocidad de inspección y eficacia del sistema.

Las características de interés de las galletas que se desean conseguir aplicando técnicas de procesamiento de imagen son:

- Área
- Circularidad
- Color
- Cantidad de chispas de chocolate

2. Materiales y métodos

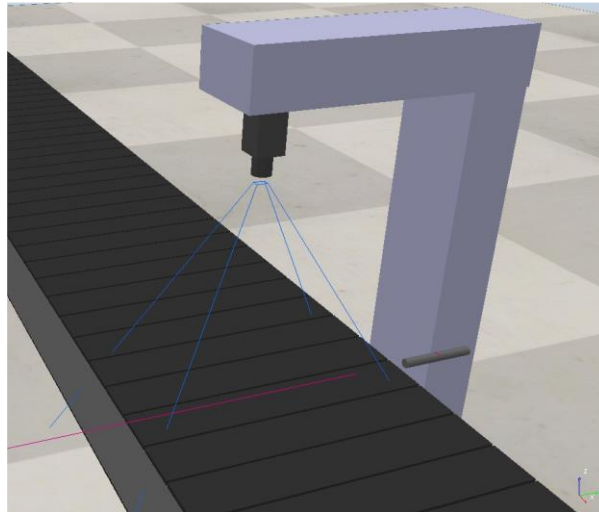
El trabajo de investigación está compuesto de 6 partes o etapas:

- Construcción de la estación de control de calidad en el simulador.
- Agregar modelo de la galleta.
- Conexión entre el simulador CoppeliaSim y MATLAB.
- Código de procesamiento de imagen.
- Establecer requisitos o tolerancias de producción.
- Ejecutar sistema de control de calidad y recolectar datos.

2.1 Construcción de la estación de control de calidad en el simulador

En el software CoppeliaSim se dispone la cinta transportadora, el sensor de visión y sensor de proximidad. Se pueden modificar las características de la cinta transportadora como largo, ancho, velocidad y aceleración, y la resolución del sensor visión. La resolución del sensor visión utilizada es de 300x500 píxeles y se sitúa a 20 cm de la cinta transportadora.

Figura 1. Elementos dispuestos en CoppeliaSim.



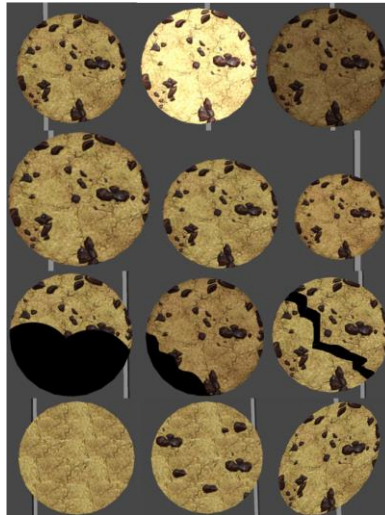
Fuente: Elaboración propia (2023).

2.2 Agregar modelo de la galleta

La galleta tiene 5 cm de diámetro y un ancho de más o menos de 0.2 cm.

Se modificó el modelo de la galleta de referencia para simular defectos que representan a galletas quemadas o crudas, tamaños distintos, rotas y sin o pocas chispas de chocolate.

Figura 2. Galletas modificadas para simular defectos.



Fuente: Elaboración propia (2023).

Las galletas fueron dispuestas en filas de cuatro en el simulador, con una separación de 2 cm entre cada una.

Figura 3. Galletas acomodadas en filas de 4 en el simulador.



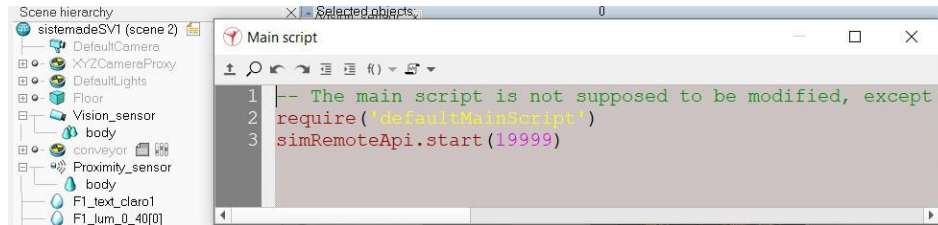
Fuente: Elaboración propia (2023).

2.3 Conexión entre el simulador CoppeliaSim y MATLAB

La interconexión entre ambos softwares facilita el intercambio de datos, como la lectura del sensor de proximidad y la imagen capturada por el sensor de visión. La versión utilizada es CoppeliaSim EDU 4.3.0.

En el software CoppeliaSim se agrega el comando `simRemoteApi.start(19999)` en el Main script, ubicado en la escena principal del software. Este comando permite iniciar la conexión por parte de CoppeliaSim.

Figura 4. Ventana *Main script*, donde se agrega el comando.



Fuente: Elaboración propia (2023).

En Matlab se agrega 6 archivos que se encuentran en la carpeta de instalación del simulador CoppeliaSim. Estos archivos son ejemplos, librerías y documentos que realizan la conexión entre ambos softwares. Al realizar el código en Matlab se debe agregar al inicio los comandos siguientes: (ver figura 5).

Figura 5. Comando que permite la conexión a CoppeliaSim por parte de Matlab.

```
sim=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
sim.simxFinish(-1); % just in case, close all opened connections.
clientID=sim.simxStart('127.0.0.1',19997,true,true,5000,5);
```

Fuente: Elaboración propia (2023).

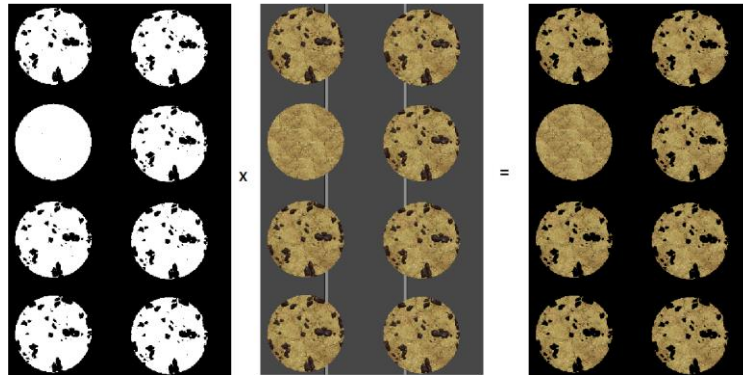
2.4 Código de procesamiento de imagen

El código tiene como objetivo verificar algunas características de las galletas, como el área, circularidad, color y cantidad de chispas de chocolate. El procesamiento de imagen sigue el siguiente orden:

2.4.1 Segmentar

La segmentación consiste en dividir una imagen en partes o regiones, diferenciando el fondo del objeto de interés. Es aplicado una segmentación por color utilizando el modelo de color HSV, en el cual solo se deja pasar el color marrón de las galletas. La máscara binaria obtenida por la segmentación se multiplica por la imagen RGB recibida del sensor visión, para obtener la imagen segmentada (Ver figura 6).

Figura 6. Máscara binaria multiplicada por la imagen original para obtener la imagen segmentada.

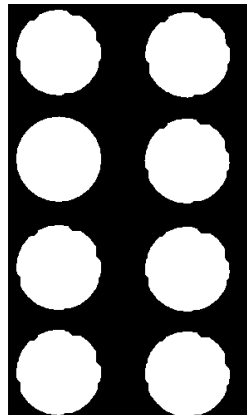


Fuente: Elaboración propia (2023).

2.4.2 Área, circularidad y cantidad de galletas

El área sirve para identificar el tamaño de la galleta y la circularidad si está roto o deforme. De la máscara binaria generada en la segmentación se pueden extraer características de las galletas, primeramente, se eliminan las chispas de chocolate aplicando la operación morfológica de cierre que permite rellenar huecos pequeños en una imagen a la vez que se conservan la forma, el tamaño de los huecos y objetos grandes de la imagen (fig. 7).

Figura 7. Máscara binaria donde se aplicó la operación de cierre.



Fuente: Elaboración propia (2023).

Luego, se etiqueta las galletas con la función `bwlabel` (Solomon y Breckon, 2011), y se obtiene la cantidad de galletas. Las propiedades se obtienen aplicando la función `regionprops` (Solomon y Breckon, 2011) a la imagen etiquetada, devolviendo una tabla con las propiedades de interés (Tabla 1).

Tabla 1. Propiedades de interés generada por regionprops.

1	2		3				4
Area	Centroid		BoundingBox				Circularity
8069	61.5022	59.1586	10.5000	7.5000	102	103	0.9738
8178	61.5249	187.1919	10.5000	135.5000	102	103	1.0079
8035	61.5522	317.8136	10.5000	266.5000	102	102	0.9692
8063	61.3382	443.9452	10.5000	392.5000	102	103	0.9664
8020	216.0439	189.6247	164.5000	138.5000	103	102	0.9656
8051	216.7224	61.7907	165.5000	10.5000	102	103	0.9711
8024	216.1436	319.7795	165.5000	268.5000	102	102	0.9723
8086	216.2523	445.9008	165.5000	394.5000	102	103	0.9730

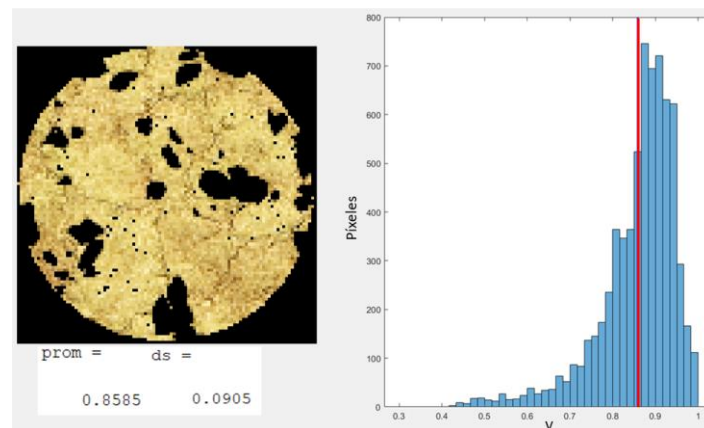
Fuente: Elaboración propia (2023).

2.4.3 Color

Sirve para identificar si una galleta esta cruda, cocinada o quemada. Se utiliza el modelo de color HSV, en específico el canal de color V, que se encarga del brillo de los pixeles. El análisis se realiza a cada galleta.

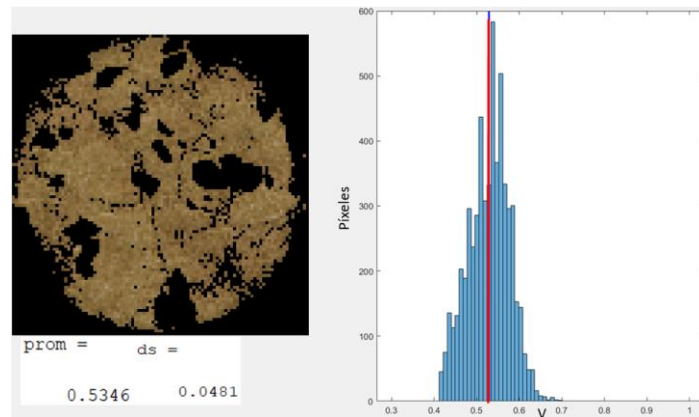
Se toma el promedio del canal V para identificar la cocción de la galleta. El promedio de V para galletas crudas o claras tiende a 1 (ver figura 8), las galletas quemadas tienden a 0.4 (ver figura 9) y las cocinadas correctamente 0.65 (ver figura 10).

Figura 8. Gráfico de cantidad de píxeles por cada valor de V de la galleta clara, línea roja que indica el promedio, valores de promedio (prom) y desviación estándar (ds).



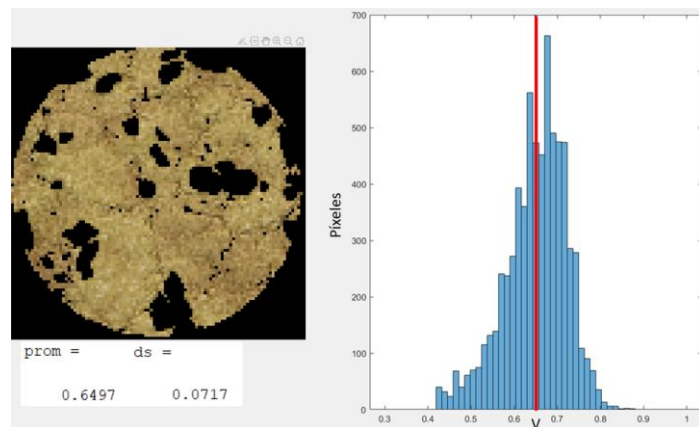
Fuente: Elaboración propia (2023).

Figura 9. Gráfico de cantidad de píxeles por cada valor de V de la galleta oscura, línea roja que indica el promedio, valores de promedio (prom) y desviación estándar (ds).



Fuente: Elaboración propia (2023).

Figura 10. Gráfico de cantidad de píxeles por cada valor de V de la galleta de referencia, línea roja que indica el promedio, valores de promedio (prom) y desviación estándar (ds).



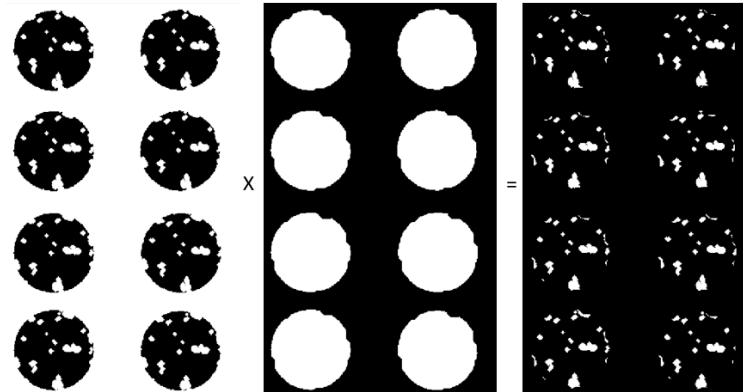
Fuente: Elaboración propia (2023).

2.4.4 Cantidad de chispas de chocolate

Se utiliza la máscara binaria para contar la cantidad de chispas de chocolate presente en cada galleta. Se aplica una operación morfológica de cierre a la máscara para eliminar el ruido o píxeles que no son chispas de chocolate. Las chispas de chocolate aparecen como píxeles negros o valor 0, debemos hacer que las chispas de chocolate se vuelvan píxeles blancos o valor 1 para poder contarlos con la función bwlable (Solomon y Breckon, 2011).

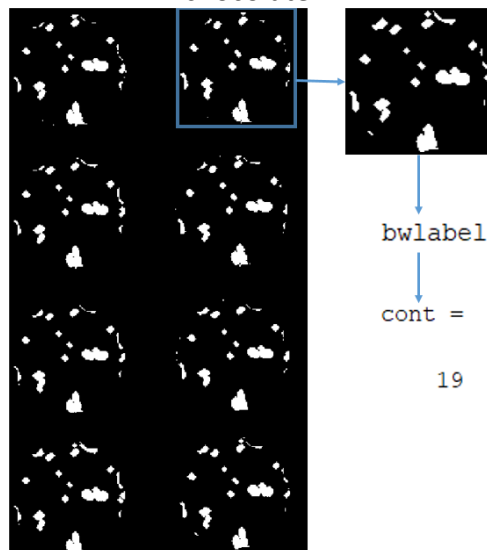
Multiplicamos la inversa de la máscara binaria, a la cual se aplicó la operación morfológica, con la máscara binaria sin chispas de chocolate (ver figura 11). La cantidad de chispas de chocolate se obtiene utilizando la función `bwlabel`. (ver figura 12).

Figura 11. Multiplicación de máscaras binarias para obtener solo las chispas de chocolate.



Fuente: Elaboración propia (2023).

Figura 12. Recorte de una galleta y obtención de la cantidad de chispas de chocolate.



Fuente: Elaboración propia (2023).

2.5 Establecer requisitos y tolerancias de producción

- Área = $8000 \pm 10\%$ píxeles, $19.6 \pm 1.96 \text{ cm}^2$.
- Circularidad > 0.9 .

- Color = 0.65 ± 7.7 %, promedio del canal V del modelo HSV.
- Cant. de chispas de chocolate > 7 unidad.

2.6 Ejecutar sistema de control de calidad y recolectar datos

La ejecución del simulador y el código se realizó en un computador con las siguientes características:

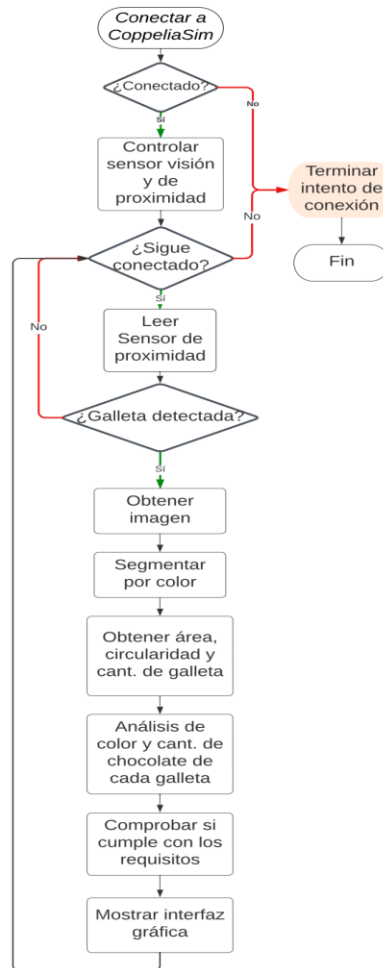
- Procesador I3-10100F.
- Gráfica RTX 3060.
- Memoria RAM 16 GB.

3. Resultados y discusión

3.1 Diagrama de flujo del código de Matlab

Al ejecutar el código desde Matlab, primeramente, se realiza la conexión al simulador CoppeliaSim. Una vez realizada la conexión, se comienza a leer el sensor de proximidad hasta que se detecte un grupo de galletas. Al detectarlo, se envía la imagen del sensor visión del simulador a Matlab, donde se realiza el procesamiento de imagen como segmentación, obtención de las características, análisis del color, y luego se compara los valores obtenidos con los requisitos. Por último, se muestra una interfaz gráfica.

Figura 13. Diagrama de flujo del código de Matlab.

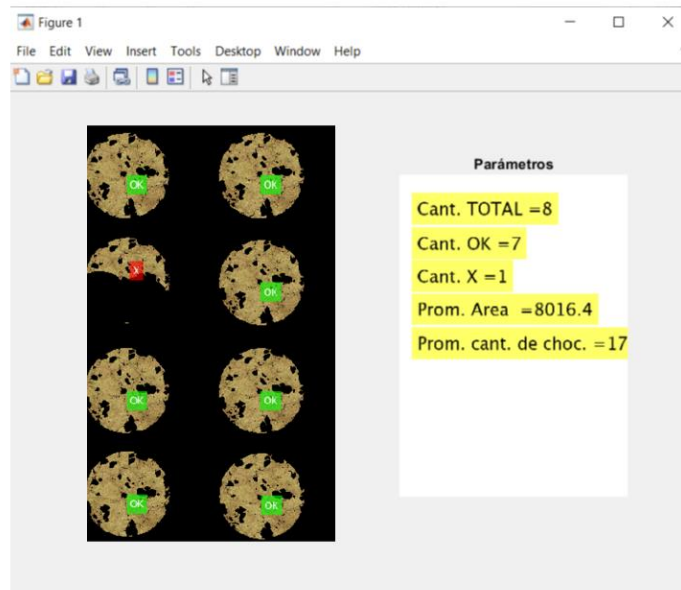


Fuente: Elaboración propia (2023).

3.2 Interfaz gráfica de usuario

La interfaz consiste en una ventana de MATLAB donde se muestran dos imágenes al usuario, primero la imagen segmentada de las galletas con símbolos (OK o X) indicando si cumplen o no con los requisitos y segundo, los parámetros de interés.

Figura 14. Interfaz gráfica del usuario.



Fuente: Elaboración propia (2023).

3.3 Tabla de galletas defectuosas

Los datos de las galletas que no cumplen con los requisitos de producción, son almacenados en la variable ERR en forma de tabla. En la tabla 2, se observa que se almacenan el área, circularidad, promedio de V, desviación estándar de V y la cantidad de chispas de chocolate.

Tabla 2. Variable ERR, con formato de tabla.

	1	2	3	4	5
	Area	Circularidad	V	V_S	Cant_choc
1	8053	0.9636	0.8210	0.0941	19
2	7997	0.9611	0.8584	0.0900	18
3	3650	0.5928	0.6512	0.0717	14
4	7734	0.9035	0.5289	0.0477	24
5	8055	0.9687	0.9373	0.0892	19
6	8060	0.9632	0.9368	0.0910	22
7	2898	0.9868	0.6419	0.0766	6
8	6896	0.8530	0.5864	0.0617	16
9	7671	0.9560	0.6473	0.0729	15
10	5121	0.9759	0.6489	0.0709	13
11	8177	1.0077	0.6490	0.0664	0
12	7772	0.9673	0.6503	0.0719	18
13	7783	0.9645	0.6502	0.0716	15
14	7789	0.9535	0.6506	0.0715	15
15	7459	0.8637	0.4758	0.0321	26
16	3507	0.5770	0.6500	0.0727	13

Fuente: Elaboración propia (2023).

3.4 Evaluación del sistema

Se evaluó el sistema utilizando un total de 80 galletas. De estas, 40 cumplían con los requisitos de producción, y las 40 galletas restantes no lo hacían.

En el análisis se consideró como falso positivo cuando un observador indica que la galleta analizada cumple con los requisitos de producción, pero el sistema muestra lo contrario, y viceversa para falso negativo.

Tabla 3. Resultados de la evaluación del sistema.

	Galletas (Un.)
OK	43
X	37
Falso Positivo	1
Falso Negativo	4

Fuente: Elaboración propia (2023)

De las 80 galletas, 5 fueron lecturas erróneas. El falso positivo fue una galleta que tenía un tamaño muy próximo del límite permitido. Los falsos negativos fueron de galletas con grietas que lo parten por la mitad y ovaladas. El código demostró tener una eficacia de 93 %.

3.5 Velocidad de inspección

Se registró el intervalo de tiempo transcurrido desde la detección del primer grupo de galletas hasta la aparición del siguiente grupo. Cada grupo consta de 8 unidades de galletas. No es posible verificar 8,2 galletas, al sistema se le da una velocidad promedio de inspección de 8 galletas por segundo. El tiempo se tomó sin mostrar la escena del simulador, ya que la cantidad de elementos en escena variaban produciendo variaciones en el consumo de recursos computacionales, como resultado, se producían variaciones en los tiempos.

Tabla 4. Tiempos de cada grupo, tiempo total, cantidad de galletas totales inspeccionadas y velocidad de inspección.

Grupo	Tiempo (seg.)
1	0,9988
2	0,9551
3	0,9699
4	0,9709
5	0,9708
6	0,9519
7	0,9841
8	0,9833
9	0,9556
10	0,9998
Tiempo Tot. (seg.)	9,7402
Galleta Tot.	80
Vel. Inspección (galletas/seg)	8,2134

Fuente: Elaboración propia (2023).

4. Conclusiones

En el trabajo realizado, se logró simular un sistema de control de calidad utilizando sensores de visión con los softwares MATLAB y CoppeliaSim.

El código desarrollado es capaz de eliminar la cinta transportadora de la imagen, dejando solo visible las galletas y obteniendo las características de interés. El método utilizado para el reconocimiento de galletas quemadas o crudas mostró ser bastante efectivo, ya que analiza un canal de color del modelo HSV correspondiente al brillo.

La interfaz gráfica del usuario permite observar las galletas y en qué estado se encuentran, además de parámetros de interés. La velocidad de inspección del sistema es de 8 galletas por segundo, con una eficacia del 93 %. El código no produce resultados consistentes cuando las galletas presentan una fisura que las divide por la mitad o si tienen una forma ovalada.

5. Declaración de financiamiento

La presente investigación se llevó a cabo con financiación propia.

6. Declaración de conflictos

El autor declara no tener conflicto de interés.

7. Declaración de autor

El autor aprueba la versión final del artículo.

8. Contribución del autor

Autor

Oscar Héctor Fernández Díaz

Contribución

Investigación, redacción y desarrollo del sistema de control de calidad.

9. Referencias

Čačić, F., Lukinac, J., Budžaki, S., & Koceva Komlenić, D. (2014). *Quality inspection of cookies using computer vision*. En M. Jukić. (Ed.), *7th International Congress Flour-Bread'13, 9th Croatian Congress of Cereal Technologists* (pp. 182-187). Josip Juraj Strossmayer University of Osijek. <https://urn.nsk.hr/urn:nbn:hr:109:576464>



Davies, E. R. (2012). *Computer and machine vision: theory, algorithms, practicalities*. 4^a ed. Academic Press.

Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley-Blackwell.