



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN MECATRÓNICA**

**TEMA:**

**“SISTEMA DE CLASIFICACIÓN DE HUEVOS MEDIANTE  
UN ALGORITMO DE VISIÓN ARTIFICIAL”**

**AUTOR: SHERMAN STALIN DE LA TORRE MORALES**

**DIRECTOR: ING. IVÁN IGLESIAS NAVARRO**

**IBARRA-ECUADOR**

**2022**



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	1004302749		
<b>APELLIDOS Y NOMBRES:</b>	DE LA TORRE MORALES SHERMAN STALIN		
<b>DIRECCIÓN:</b>	IMBABURA - SAN ROQUE		
<b>EMAIL:</b>	<a href="mailto:ssdelatorrem@utn.edu.ec">ssdelatorrem@utn.edu.ec</a>		
<b>TELÉFONO FIJO:</b>	0967747566	<b>TELÉFONO MÓVIL:</b>	0967747566

DATOS DE LA OBRA	
<b>TÍTULO:</b>	“SISTEMA DE CLASIFICACIÓN DE HUEVOS MEDIANTE UN ALGORITMO DE VISIÓN ARTIFICIAL”
<b>AUTOR:</b>	DE LA TORRE MORALES SHERMAN STALIN
<b>FECHA:</b>	21/10/2022
<b>PROGRAMA:</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA:</b>	INGENIERO EN MECATRÓNICA
<b>DIRECTOR:</b>	IVÁN IGLESIAS NAVARRO



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**2. CONSTANCIAS**

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días del mes de octubre de 2022

**EL AUTOR:**

De La Torre Morales Sherman Stalin

C.I.: 1004302749



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**CERTIFICACIÓN**

En calidad de director del trabajo de grado “**SISTEMA DE CLASIFICACIÓN DE HUEVOS MEDIANTE UN ALGORITMO DE VISIÓN ARTIFICIAL**”, presentado por el egresado **SHERMAN STALIN DE LA TORRE MORALES**, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, a los 27 días del mes de septiembre de 2022

Ing. Iván Iglesias Navarro

DIRECTOR DE TESIS

# Agradecimientos

A mis padres por todo su apoyo incondicional espero que la vida me de la oportunidad de retribuirles todo lo que han hecho por mí.

A mis hermanos que son tan increíbles y únicos que con sus ocurrencias me alegran todos los días.

A toda mi familia quienes a pesar de mis errores y defectos los soportaron y continúan apoyándome.

A mi director Ingeniero Iván Iglesias por apoyarme y guiarme a lo largo de este proceso.

*Sherman Stalin De La Torre Morales*

# **Dedicatoria**

Dedico este trabajo a mis padres Carlos y Claudia, quienes con cariño, paciencia y esfuerzo me permitieron alcanzar una de mis metas en la vida.

# Resumen

El presente trabajo muestra el desarrollo de un algoritmo de visión artificial basado en código abierto que permite clasificar distintos huevos de gallina mediante la estimación del peso a través de una imagen en 2D; para lograr el objetivo en este trabajo el peso se calcula tomando en consideración únicamente dos factores geométricos que son el ancho y alto del producto. El algoritmo inicia con la adquisición de la imagen, para posteriormente realizar el procesamiento de la imagen, detectar los contornos, analizar las dimensiones y finalizar calculando el peso y clasificando el producto. El funcionamiento del algoritmo se basa en una escena de pruebas el cual se trata de un ambiente controlado que consta de una cámara web Genius FaceCam 1000X y una fuente de luz ubicadas de manera frontal a la base.

Los criterios de clasificación se basan en la norma técnica ecuatoriana para ovoproductos NTE INEN 1973. Para validar el funcionamiento del algoritmo se realizó las respectivas pruebas con 21 huevos de gallina seleccionados al azar obteniendo 19 coincidencias, teniendo como resultado una eficiencia de 90,47% en la clasificación, y un 97,32% de eficiencia en la estimación del peso.

**Palabras clave:** Visión artificial, procesamiento de imágenes, clasificación de huevos, estimación del peso.

# Abstract

The present work shows the development of an artificial vision algorithm based on open source that allows classifying different chicken eggs by estimating the weight through a 2D image; to achieve the objective in this work the weight is calculated taking into consideration only two geometric factors which are the width and height of the product. The algorithm starts with the acquisition of the image, to later process the image, detect the contours, analyze the dimensions and finally calculate the weight and classify the product. The operation of the algorithm is based on a test scene which is a controlled environment consisting of a Genius FaceCam 1000X webcam and a light source placed in front of the base.

The classification criteria are based on the Ecuadorian technical standard for egg products NTE INEN 1973. To validate the operation of the algorithm, the respective tests were carried out with 21 randomly selected hen eggs, obtaining 19 matches, resulting in an efficiency of 90.47% in classification and 97.32% efficiency in weight estimation.

**Key words:** Artificial vision, image processing, egg classification, weight estimation.



# Índice de Contenido

<b>Resumen</b> .....	VII
<b>Abstract</b> .....	VIII
<b>Índice de figuras</b> .....	XIII
<b>Índice de tablas</b> .....	XV
<b>Introducción</b> .....	1
<b>Planteamiento del problema</b> .....	1
Problema .....	1
Objetivos .....	3
Objetivo general .....	3
Objetivos específicos .....	3
Justificación .....	3
Alcance .....	4
<b>Capítulo 1</b> .....	5

<b>Revisión literaria</b> .....	5
1.1. Huevos de gallina.....	5
1.2. Clasificación de huevos en el Ecuador .....	6
1.3. Clasificadoras de huevos.....	7
1.3.1. Clasificadora de huevos Mobanette3 .....	7
1.3.2. Clasificadora de huevos Yemita.....	8
1.4. Python .....	9
1.5. OpenCV .....	10
1.6. Visión artificial .....	10
1.6.1. Sistema de visión artificial .....	10
1.6.2. Etapas en el proceso de visión artificial .....	11
1.7. Componentes de un sistema de visión artificial.....	13
1.7.1. Sistema de procesamiento .....	14
1.7.2. Sistema de iluminación .....	14
1.7.3. Sistema de captación .....	16
1.8. Aportes previos de la visión artificial para la clasificación de huevos por peso a partir del procesamiento de imágenes.....	17
1.9. Propuesta.....	18
<b>Capítulo 2</b> .....	19
<b>Metodología</b> .....	19

2.1.	Adquisición de la imagen del huevo .....	20
2.2.	Preprocesamiento de la imagen .....	23
1.	Conversión a escala de grises .....	23
2.	Filtro gaussiano .....	24
3.	Detección de bordes de Canny .....	24
4.	Filtro de dilatación .....	25
5.	Filtro de erosión .....	26
2.3.	Detección de contornos .....	26
2.3.1.	Orden de los contornos .....	27
2.4.	Análisis de dimensiones .....	28
2.4.1.	Análisis del tamaño en píxeles .....	29
2.5.	Cálculo del peso .....	31
2.6.	Librerías utilizadas .....	32
2.6.1.	NumPy .....	32
2.6.2.	Imutils .....	33
2.6.3.	SciPy .....	33
2.6.4.	Math .....	34
<b>Capítulo 3</b>	.....	<b>35</b>
<b>Resultados</b>	.....	<b>35</b>
3.1.	Análisis de resultados .....	39

<b>Capítulo 4</b> .....	42
<b>Conclusiones y trabajo futuro</b> .....	42
4.1. Conclusiones .....	42
4.2. Trabajos futuros .....	44
<b>Referencias</b> .....	45
<b>Anexo</b> .....	52
<b>Código del programa</b> .....	52

# Índice de figuras

<b>Figura 1.1.</b> <i>Huevo de gallina</i> .....	5
<b>Figura 1.2.</b> <i>Clasificadora Mobanette3</i> [10] .....	7
<b>Figura 1.3.</b> <i>Clasificadora Yemita</i> [11] .....	8
<b>Figura 1.4.</b> <i>Sistema de visión artificial</i> [20].....	11
<b>Figura 1.5.</b> <i>Imagen: a) Normal, b) Procesada (Histograma ecualizado), c) Características (Contorno)</i> .....	12
<b>Figura 1.6.</b> <i>Componentes físicos de un sistema de visión artificial</i> [25] .....	14
<b>Figura 1.7.</b> <i>Iluminación frontal</i> [28] .....	15
<b>Figura 1.8.</b> <i>Iluminación lateral</i> [28].....	16
<b>Figura 1.9.</b> <i>Cámara con sensor tipo CCD</i> [30].....	16
<b>Figura 2.1.</b> <i>Diagrama de flujo para la elaboración del algoritmo</i> .....	20
<b>Figura 2.2.</b> <i>Escenario para realizar pruebas</i> .....	22
<b>Figura 2.3.</b> <i>Imagen digital antes de ser procesada</i> .....	22
<b>Figura 2.4.</b> <i>Imagen en escala de grises</i> .....	23
<b>Figura 2.5.</b> <i>Imagen con filtro gaussiano</i> .....	24
<b>Figura 2.6.</b> <i>Detección de bordes mediante Canny</i> .....	25
<b>Figura 2.7.</b> <i>Imagen con filtro de dilatación</i> .....	25

<b>Figura 2.8.</b> <i>Imagen con filtro de erosión</i> .....	26
<b>Figura 2.9.</b> <i>Cuadro que bordea al objeto con sus respectivos puntos medios</i> .....	28
<b>Figura 2.10.</b> <i>Variables encontradas de ancho y alto</i> .....	30
<b>Figura 2.11.</b> <i>Peso y clasificación proporcionados por el algoritmo</i> .....	32

# Índice de tablas

<b>Tabla 1.1.</b> <i>Clasificación de huevos acorde a su tamaño y en función de su peso</i> .....	6
<b>Tabla 2.1.</b> <i>Características Genius FaceCam 1000X</i> .....	21
<b>Tabla 3.1.</b> <i>Peso y clasificación realizados de forma manual</i> .....	36
<b>Tabla 3.2.</b> <i>Peso y clasificación proporcionados por el algoritmo</i> .....	37
<b>Tabla 3.3.</b> <i>Eficiencia de clasificación con el algoritmo</i> .....	39
<b>Tabla 3.4.</b> <i>Porcentaje de error en la estimación del peso</i> .....	40
<b>Tabla 3.5.</b> <i>Porcentaje de error por categoría</i> .....	41

# **Introducción**

## **Planteamiento del problema**

### **Problema**

A nivel mundial la producción de huevos de gallina se ha incrementado entre los años 2000 y 2017 en un 56,6% generando que durante este periodo exista un crecimiento anual de 2,7%. En 2017 la producción mundial de huevos alcanzó los 1,417 billones de unidades, que corresponde a 80,1 millones de toneladas, un 1,4% más que en el mismo período del año anterior [1]. De acuerdo con lo mencionado se estima que para el año 2035 la producción mundial de huevos puede llegar a los 100 millones de toneladas por año [2]. En el año 2017 en el continente americano se produjeron alrededor de 286 billones de huevos siendo así que cerca del 20,2% de la producción total de huevos se da en América entre los países de Estados Unidos, México y Brasil quienes cuentan con una producción del 7,8%, 3,5% y 3,2% respectivamente [3].

En 2019 en el Ecuador el consumo de huevos al año per cápita (kilogramos/personas) fue de 226 unidades, cifra superior en un 6% a 2018, que se situó en 213 huevos [4]. La producción diaria de esta fuente básica de proteína para el consumo de un ecuatoriano fue de 9,38 millones de unidades en 2017, 10,01 millones de unidades en 2018 y 10,81 millones de unidades en 2019.



Siendo así que la producción de huevos en el Ecuador fue de 361'078.496 unidades semanales en 2019, lo que muestra que la producción anual promedio supera los 4 mil millones de huevos. Entre 2014 y 2015, las unidades de huevo producidas se aumentaron un 7%, pero a partir de los años siguientes se observa un decrecimiento de hasta un -18%, sin embargo, tanto la producción como el autoconsumo muestran un crecimiento de casi el 100% en huevos de mesa para el 2019. Es así que, de la producción total entre 2014 y 2019 se destina al autoconsumo el 7% y se comercializa el 86,2% de la producción anual total [5].

De acuerdo con los registros, de toda la cantidad de huevos producidos la mayoría se destina a la venta, por lo que es necesario que se realice un proceso de clasificación para que todos cumplen con los requisitos para su comercialización. Dada a la alta cobertura que demanda el consumo de huevos, las empresas deben contar con sistemas altamente automatizados que ayuden a mejorar sus procesos para de esta manera generar un elevado rendimiento en su capacidad de producción, debido a que, en la actualidad la clasificación de los huevos en las medianas y grandes empresas se realiza mediante el empleo de máquinas sumamente costosas, mientras que en las granjas este proceso se realiza aun de forma manual [6].

En todas las empresas, microempresas y granjas la clasificación de huevos se realiza de acuerdo con el tamaño que es proporcional al peso que este tenga, presentado la misma problemática en el proceso de clasificación del producto.

Durante el proceso de pesaje se detectan errores debido a fallas de equipos como la desestabilidad en la balanza que no proporciona un resultado fijo al momento de pesar el producto y fallas humanas como la lectura subjetiva que hace el trabajador después de jornadas extensas en esta labor. Esto está ocasionando que todas las empresas vean reflejado considerables pérdidas económicas.

Para dar solución a la problemática descrita, se propone diseñar un sistema que mediante un algoritmo de visión artificial realice la clasificación de huevos. Este podrá clasificar el producto de acuerdo con el peso que estos tengan; el desarrollo de este algoritmo se realiza con la finalidad de proporcionar una nueva forma de clasificación en lugar de la forma típica que es por medio de máquinas industriales o de forma manual.

## **Objetivos**

### **Objetivo general**

- Desarrollar un sistema de clasificación de huevos mediante un algoritmo de visión artificial.

### **Objetivos específicos**

- Determinar los requisitos y requerimientos funcionales del sistema.
- Determinar las condiciones adecuadas para la adquisición de imágenes.
- Diseñar un algoritmo de visión artificial para la clasificación de huevos.
- Validar el algoritmo.

## **Justificación**

En el continente americano se producen alrededor del 20,2% de los huevos que se producen a nivel mundial. Desde este punto de vista en la actualidad las empresas avícolas en general necesitan poseer sistemas altamente automatizados que permitan procesar una gran cantidad de sus productos para poder competir en el mercado [7].

Debido al gran aumento de producción de huevos de gallina que está teniendo la industria por el alto consumo del mismo, está ocasionando que todos los productores de huevos tanto en

galpones y grandes industrias tengan la necesidad de buscar alternativas que faciliten tareas como la clasificación y control de la calidad para poder comercializar el producto de una forma más rápida y eficiente para la empresa. Para esto se necesita que todos los pequeños y grandes productores automaticen sus procesos para de esta manera mejorar el rendimiento de sus sistemas de producción y clasificación [8]. El algoritmo que se va a desarrollar busca solventar una necesidad latente identificada en los pequeños y grandes productores de huevos, como es el control del peso del producto. Al desarrollar un algoritmo para la clasificación de huevos, que mediante el empleo de técnicas de visión artificial pueda estimar el peso, se pretende generar un método de clasificación nuevo y rápido, así como también una forma de clasificación a bajo costo, generando así que las pérdidas que los productores ven en sus ingresos sean muy bajas debido a que con esta nueva forma de clasificación se pretende reducir en gran medida la intervención de mano de obra lo que ocasionara que las fallas producidas se reduzcan drásticamente.

## **Alcance**

El proyecto que se va a realizar constará de una estructura en la que se colocará una cámara que estará en una posición fija. Para obtener una buena calidad de las imágenes la toma de las mismas se realizará en un ambiente controlado con condiciones de iluminación adecuada. La clasificación de huevos será mediante un algoritmo en código abierto basado en visión artificial que realizará una estimación del peso que tenga el producto. En el presente trabajo se realizará la clasificación un huevo a la vez. Para validar el funcionamiento del algoritmo se analizarán los resultados obtenidos en base a la Norma Técnica Ecuatoriana NTE INEN 1973:2013.

# Capítulo 1

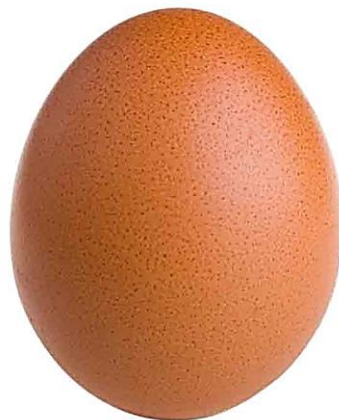
## Revisión literaria

### 1.1. Huevos de gallina

El huevo de gallina (Figura 1.1.) es uno de los alimentos al que casi toda la población puede acceder, su alto valor biológico, así como su elevada fuente de proteína y precios relativamente bajos, han ocasionado que en la actualidad en la industria alimenticia exista una alta demanda de su consumo, ya que está relacionada con una alimentación balanceada [1] [8].

#### **Figura 1.1.**

*Huevo de gallina*



## 1.2. Clasificación de huevos en el Ecuador

En el Ecuador la normativa NTE INEN 1973 rige los estándares de clasificación de huevos comerciales y ovoproductos en las granjas avícolas. De acuerdo con esta norma en cuanto a su masa (peso) unitario la tolerancia máxima por huevo es de  $\pm 1$  gramo para todos los tipos o tamaños de huevos [9] . En la tabla 1.1 se puede observar la clasificación de huevos con respecto a su peso.

**Tabla 1.1.**

*Clasificación de huevos acorde a su tamaño y en función de su peso*

Tipo (Tamaño)	Masa unitaria en g		Masa por decena en g		Masa por 30 huevos en g	
	Mínimo ( $\geq$ )	Máximo ( $<$ )	Mínimo ( $\geq$ )	Máximo ( $<$ )	Mínimo ( $\geq$ )	Máximo ( $<$ )
I Super gigante	76	---	912	---	2280	---
II Gigante	70	76	840	912	2100	2280
III Extragrande	64	70	768	840	1920	2100
IV Grande	58	64	696	768	1740	1920
V Mediano	50	58	600	696	1500	1740
VI Pequeño	46	50	552	600	1380	1500
VII Inicial	---	46	---	552	---	1380

### 1.3. Clasificadoras de huevos

Existen distintos tipos de máquinas clasificadoras, cada una de ellas presentan beneficios en cuanto a precio y forma de clasificación, a continuación, se describe las clasificadoras más asequibles que se pueden encontrar en el mercado.

#### 1.3.1. Clasificadora de huevos Mobanette3

La clasificadora de huevos Mobanette3 (Figura 1.2.) está diseñada para el uso en pequeñas granjas, es fácil de usar y cuenta con eficiencia y calidad para procesar 1.600 huevos/h; necesita poco mantenimiento y su consumo de energía es sumamente bajo.

#### Características

- Capacidad de clasificación de 1.600 huevos/h.
- Mesa de empaquetado para 4 tipos de huevos, con unidades que son fáciles de ajustar.
- Precisión de pesaje de las unidades de  $\pm 1$  gramo.

#### Figura 1.2.

*Clasificadora Mobanette3 [10]*



### 1.3.2. Clasificadora de huevos Yemita

La clasificadora de huevos Yemita (Figura 1.3.) garantiza la eficiencia y fiabilidad para clasificar 3.600 huevos/hora; cuenta con parámetros de clasificación modificables lo que lo hace apto para cualquier situación; además posee cuatro sectores donde se depositan los huevos los cuales sirven para almacenar el producto a pesar de que las cucharas que los mantiene se encuentren ocupadas.

#### Características

- Material de fabricación: aluminio, hierro, fibra de vidrio, yelco, vida útil de 30 años.
- Motor eléctrico de 1/6HP y funciona con 220v.
- Clasifica los huevos en 4 o 5 diferentes pesos.
- Puede clasificar 3.600 huevos/ h [11].

#### Figura 1.3.

*Clasificadora Yemita [11]*



En la actualidad las industrias buscan la manera de reemplazar la intervención humana por sistemas automatizados que garanticen la eficiencia en los distintos procesos, esto se debe principalmente a que dichos sistemas son más rápidos y eficientes.

La visión artificial es una tecnología novedosa que permite a dispositivos como computadoras adquirir y analizar una imagen de una escena real con el fin de obtener información para controlar máquinas o procesos. Es así que, actualmente existen muchas aplicaciones que se encuentran en áreas como: robótica, manufactura, vigilancia aérea, teledetección, medicina, clasificación y selección automática de productos agrícolas y alimentarios, entre otros [12].

Existen muchos lenguajes de programación que permiten desarrollar aplicaciones de visión artificial, pero en la actualidad Python es el preferido por desarrolladores ya que cuenta con distintas herramientas.

#### **1.4. Python**

Python es un lenguaje de programación de alto nivel que se rige bajo una licencia de software libre, cuenta con una amplia biblioteca estándar y distintas herramientas [13].

Su simplicidad y legibilidad para desarrollar códigos han hecho que este sea ampliamente utilizado para todo tipo de proyectos, pues su versatilidad, ha ayudado a realizar distintas tareas comunes sin necesidad de realizar programas desde cero, es por ello que Python se basa en dos premisas que lo ha convertido en la herramienta más popular, estas son:

- Filosofía DRY (Don't Repeat Yourself).
- Filosofía RAD (Rapid Application Development).

Así también, cuenta con una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a que aplicación como la visión artificial sea más fácil de



implementar, tal es el caso de OpenCV que es la librería principal para el desarrollo de trabajos de visión artificial [14] [15].

### **1.5. OpenCV**

OpenCV es una biblioteca de visión por computador de código abierto la cual se diseñó para lograr una mayor eficiencia computacional que se enfoca en realizar aplicaciones en tiempo real, esta librería permite realizar distintas tareas ya que contiene más de 500 funciones que abarcan muchos ámbitos de la visión, como la de inspección de productos, seguridad, interfaz de usuario, calibración de cámaras, robótica, entre otros. Así también, cuenta con más de 2500 algoritmos que permite desarrollar lo mencionado [16].

El módulo OpenCV se importa como: `import cv2`

### **1.6. Visión artificial**

Los estudios realizados con respecto al tema de visión artificial se basan en simular la capacidad visual de las personas por medio del uso de un computador; es así que, mediante la captura de una imagen o video se pretende analizar la información y características de los objetos para percibir y comprender los factores inmersos en el objeto de estudio [17].

Para lograr lo mencionado se utilizan técnicas que ayudan a adquirir, procesar, analizar y comprender las imágenes tomadas del mundo real, de tal forma que esta información sea comprensible por el ordenador, para que este pueda actuar según convenga en una determinada situación de la manera que lo haría una persona [18].

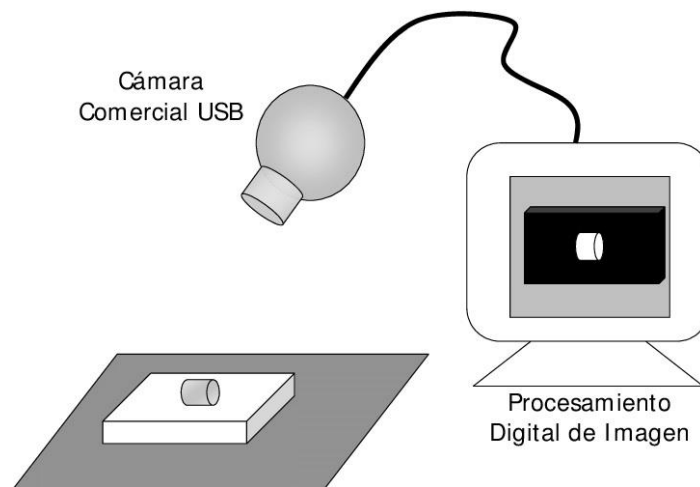
#### **1.6.1. Sistema de visión artificial**

De acuerdo con [19] un sistema de visión artificial hace referencia a un conjunto de elementos que se integran con la finalidad de extraer información del mundo físico por medio de imágenes

bidimensionales (ver Figura 1.4.). Para esto se hace uso de un computador el cual mediante un algoritmo desarrollado puede reconocer, analizar y entender la escena para la cual ha sido programada, es decir, procesar las imágenes para extraer información.

**Figura 1.4.**

*Sistema de visión artificial [20]*



**1.6.2. Etapas en el proceso de visión artificial**

Cada aplicación de visión artificial tiene sus propias características específicas, esto se debe principalmente a que el reconocimiento de imágenes depende en gran medida del objetivo que se debe cumplir. Desde este punto de vista, aunque no existen pasos establecidos a seguir, se puede decir que hay una serie de etapas que son comunes y habituales en el proceso de visión artificial, cada uno de las cuales se detallan a continuación.

**1.6.2.1. Adquisición de Imágenes**

Se considera como la etapa primordial debido a que se busca una imagen adecuada para el análisis actual y de las etapas siguientes. Cabe mencionar que el método de adquisición depende de dos factores, el primero, cual es el objeto de interés y el segundo, cuál es el propósito de estudio [21].

### 1.6.2.2. Pre-Procesamiento de imágenes

En esta etapa se busca mejorar la calidad de la imagen, para lo cual se usan técnicas que ayudan a eliminar imperfecciones tomadas en la etapa anterior, resaltar características, mejorar la intensidad, mejorar la calidad, entre otros.

### 1.6.2.3. Segmentación

Por medio de esta etapa se puede dividir la imagen capturada en las diferentes partes con la finalidad de extraer todas las características que se requieran de la imagen; con una segmentación adecuada se brinda una solución correcta y por ende se asegura que el sistema desarrollado es confiable [22].

### 1.6.2.4. Extracción de características

En esta etapa se busca extraer las características deseadas de forma acertada para su respectivo análisis; por lo general los datos que se extraen para ser analizados se basa en: color, distancias, áreas, ángulos, contornos, entre otros.

#### Figura 1.5.

*Imagen: a) Normal, b) Procesada (Histograma ecualizado), c) Características (Contorno)*



### **1.6.2.5. Análisis y reconocimiento de la imagen**

De manera general los sistemas de visión artificial tienen como objetivo:

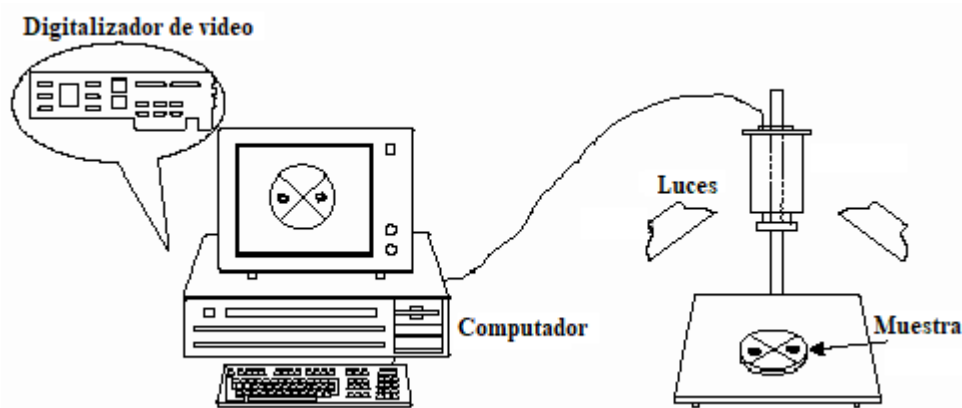
- **Identificación de objetos:** El propósito del sistema es el de identificar algún objeto que se encuentra en la imagen.
- **Clasificación de objetos:** El propósito del sistema es categorizar los elementos que se encuentran en la imagen, para lograr esto los algoritmos hacen uso de características del objeto para así determinar la clase o grupo al que pertenecen los elementos de la imagen.
- **Mediciones de objetos:** El sistema tienen como finalidad medir el objeto de estudio en la imagen adquirida.
- **Conteo de objetos:** El propósito de este tipo de reconocimiento es el contar los objetos de estudio en la imagen [23] [24].

### **1.7. Componentes de un sistema de visión artificial**

Tal como se puede ver en la Figura 1.6. un sistema de visión artificial cuenta con componentes estándar que ayudan en el análisis del proceso, estos son, sistema de procesamiento, sistema de iluminación, sistema para captar y adquirir las imágenes en el computador; a continuación se detalla cada uno de ellos [25] [26].

**Figura 1.6.**

*Componentes físicos de un sistema de visión artificial [25]*



### **1.7.1. Sistema de procesamiento**

El sistema de procesamiento es el encargado de recibir, almacenar y procesar las imágenes para extraer la información y tomar decisiones. La velocidad con la que se realiza estas tareas depende directamente de las características del sistema y del lenguaje en el que se realice el programa [27].

### **1.7.2. Sistema de iluminación**

Su función es la de ayudar en la etapa de adquisición a obtener imágenes con características relevantes ya que de esta manera se puede realizar un mejor procesamiento y por ende resultados más precisos.

#### **1.7.2.1. Tipos de iluminación**

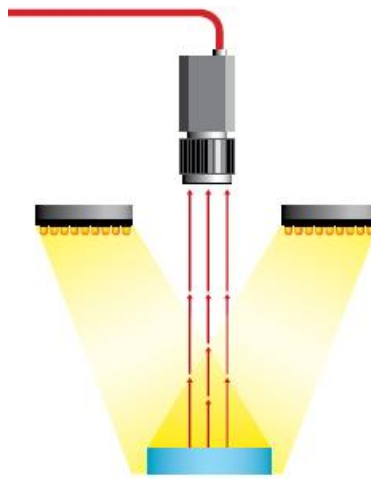
Existen varios tipos de iluminación que dependen del tipo de aplicación a desarrollar debido a que cada una resalta diferentes aspectos del objeto que se está analizando. A continuación, se explica los tipos de iluminación más utilizados en sistemas de visión artificial.

### a) Iluminación frontal

Consiste en posicionar la fuente de luz y la cámara en la misma dirección para que la luz incida frontalmente de forma directa sobre el objeto (Figura 1.7.). Esto permite visualizar las características externas del objeto como color, forma o superficie, logrando así realizar una mejor segmentación y reconocimiento de patrones [28].

### Figura 1.7.

*Iluminación frontal* [28]

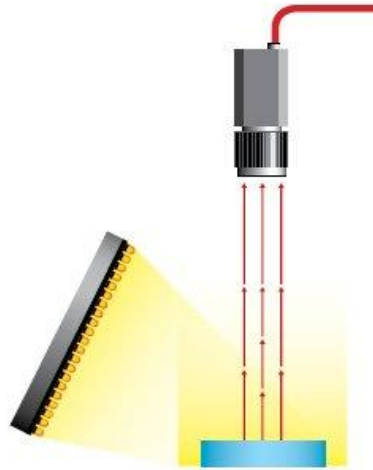


### b) Iluminación lateral

Consiste en ubicar la cámara mirando al objeto y la fuente de luz de manera lateral a él (Figura 1.8.), esta técnica se utiliza con la finalidad de mostrar ciertos detalles o aspectos que solo se pueden apreciar con este tipo de iluminación, el inconveniente es que este tipo de iluminación puede formar sombras indeseadas [28].

**Figura 1.8.**

*Iluminación lateral [28]*



### **1.7.3. Sistema de captación**

Las cámaras más comunes son las basadas en CCD (Charge Couple Device) (Figura 1.9.) ya que son de bajo costo, bajo consumo, poseen una buena resolución, duran más tiempo y son más fáciles de manejar. Así también, este tipo de cámaras poseen una baja deformación geométrica de la imagen, son sensibles a la luz, y poseen una buena resolución [29].

**Figura 1.9.**

*Cámara con sensor tipo CCD [30]*



## **1.8. Aportes previos de la visión artificial para la clasificación de huevos por peso a partir del procesamiento de imágenes**

Se han realizado diversos estudios con respecto al tema de clasificación de huevos, sin embargo, en su gran mayoría abordan el tema de control de la calidad para la detección de defectos, como es el caso de [31], [32] y [33], que realizaron sistemas de visión artificial para la identificación de huevos fértiles, controlar si el cascaron posee grietas y observar imperfecciones como manchas sobre el cascaron respectivamente.

En cuanto al tema de pesaje de huevos se han realizado estudios como [34] que propone el cálculo del peso a partir de 12 características del huevo y así proporcionar los datos obtenidos a una red de perceptrón multicapa (MLP) para predecir el peso del huevo. Otra investigación realizada [35] basa su estudio en el cálculo del peso a partir de parámetros geométrico como el ancho y alto del producto que se obtiene mediante el procesamiento de imágenes. Así también [36] utiliza de la misma manera los parámetros del ancho y la altura que se obtiene a través de procesamiento de imágenes para estimar el peso usando el Sistema de Inferencia Neuro-Fuzzy Adaptativo (ANFIS).

Todos los estudios mencionados utilizan distintos métodos para calcular el volumen y peso de los huevos, algunos son mucho más complicados que otros, pero todos sin excepción proporcionan una precisión sumamente alta. Desde este punto de vista se puede evidenciar que la parte primordial para obtener datos relevantes y que se acerquen a la realidad se basan fundamentalmente en la adquisición adecuada de imágenes para su posterior procesamiento debido a que como se mencionó anteriormente únicamente con los parámetros geométricos de ancho y alto se pueden obtener resultados aceptables.



## **1.9. Propuesta**

Los diferentes estudios de visión artificial con respecto al tema de clasificación de huevos se enfocan básicamente en controlar la calidad y detección de defectos. Desde este punto de vista y con la información antes mencionada se propone resolver el problema de clasificación de huevos por peso debido a que existen escasas investigaciones y estudios en el tema mencionado; en este trabajo el cálculo del peso se realizará tomando en cuenta factores geométricos como el ancho y alto del producto, debido a que los estudios realizados sobre el tema han abordado la resolución de este problema mediante dichos parámetros obteniendo resultados favorables.

## Capítulo 2

En esta sección se realiza la respectiva explicación del proceso que se siguió para llegar al objetivo planteado, es así como el usuario puede conocer todos los pasos que se usaron para resolver el problema.

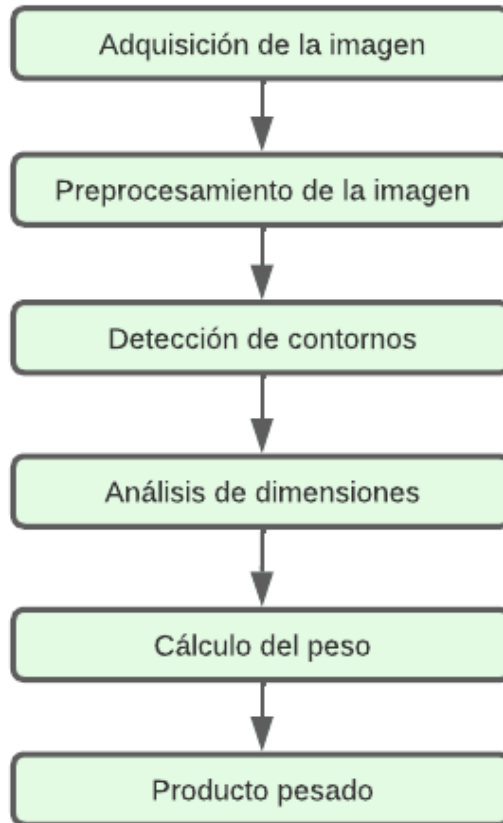
### Metodología

Para la elaboración del algoritmo de clasificación de huevos por peso se muestra el diagrama de flujo de la Figura 2.1.

El algoritmo inicia con la adquisición de la imagen, con la finalidad de mejorar la calidad a través del procesamiento de imágenes, para posteriormente encontrar los contornos y ordenarlos de izquierda a derecha. Con los contornos detectados se procede a dibujar el cuadro delimitador de cada uno de los objetos que posee la imagen. A partir de esto se procede a calcular la distancia euclidiana con la ayuda del cuadro delimitador y seguidamente se calcula el ancho y alto del objeto mediante un factor de conversión de pixeles a centímetros tomando como medida estándar el objeto de referencia. El peso se calcula a través de la fórmula de la densidad,  $\rho = \frac{masa}{volumen}$ .

**Figura 2.1.**

*Diagrama de flujo para la elaboración del algoritmo*



### **2.1. Adquisición de la imagen del huevo**

La escena donde se realizan las respectivas pruebas es de suma importancia, es así que de acuerdo con estudios previos [37] y [38], es necesario simular las condiciones óptimas donde el algoritmo desarrollado funcione de manera adecuada solucionando el problema planteado, es por ello que, se debe simular un ambiente controlado para lo que se debe tener en consideración aspectos tratados en la Sección 1.6. y 1.7. del Capítulo 1 donde se realiza el análisis de los elementos que ayudan a realizar una simulación óptima.

Así también, tomando en cuenta las especificaciones realizadas en [28] y [39] para obtener una imagen de calidad, se utiliza un sistema de iluminación de tipo frontal debido a que permite visualizar las características del objeto a estudiar, específicamente la forma y el contorno, con lo que se logra realizar una mejor identificación de los objetos en la imagen.

En cuanto, al sensor que ayuda a adquirir la imagen se hace uso de una webcam Genius FaceCam 1000X, su selección se basa de acuerdo a [40]. En la siguiente tabla se detalla las características de la cámara web [30].

**Tabla 2.1.**

*Características Genius FaceCam 1000X*

<b>FaceCam 1000x</b>	<b>Especificaciones</b>
Tipo de enfoque	Manual
Resolución	720P
Sensor de imagen	CMOS de píxeles de alta definición de 720p
Sistema operativo	Windows 8, 7, Vista o posterior

En la Figura 2.2. las luces y la webcam se montan en un escenario cerrado para evitar alteraciones en la iluminación, la cámara se ubica de forma central con respecto a la fuente de luz y de forma paralela con el fondo del escenario, cabe mencionar que la fuente de luz y la webcam se ubican a una altura de 26 cm con respecto a la base; así también, la escena cuenta con un fondo homogéneo que permite identificar con mayor facilidad el objeto.

**Figura 2.2.**

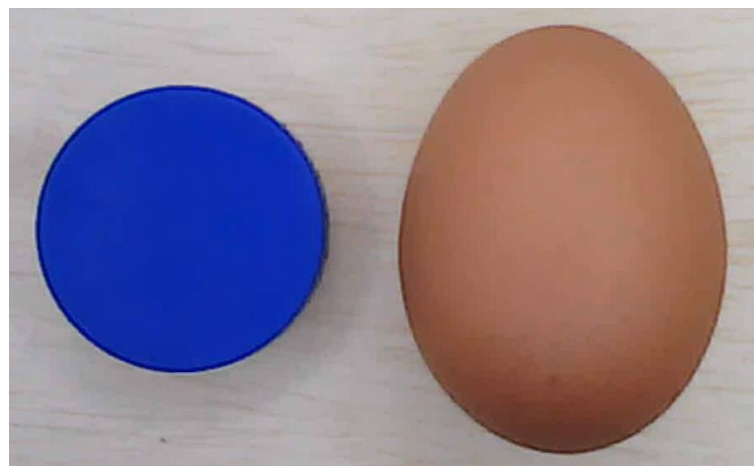
*Escenario para realizar pruebas*



La imagen digital del huevo de gallina en la Figura 2.3. se adquiere a una resolución de 720x640 píxeles. Tras su adquisición los objetos para analizar se tratan del huevo de gallina, color marrón, apariencia esferoidal, con una masa que varía entre 46 gramos hasta 76 gramos; así también, se analiza el objeto de referencia que es de tipo circular con un diámetro de 41mm.

**Figura 2.3.**

*Imagen digital antes de ser procesada*



## 2.2.Preprocesamiento de la imagen

El procesamiento de la imagen digital se basa en la obtención del contorno del objeto, por tal motivo tomando en cuenta la Sección 1.6.2.; para alcanzar el objetivo se realiza lo siguiente:

1. Adquirir la imagen en RGB.
2. Convertir la imagen a escala de grises.
3. Aplicar un filtro gaussiano.
4. Detectar los bordes de los objetos mediante Canny.
5. Aplicar un filtro de dilatación y erosión para que los bordes detectados sean más visibles.

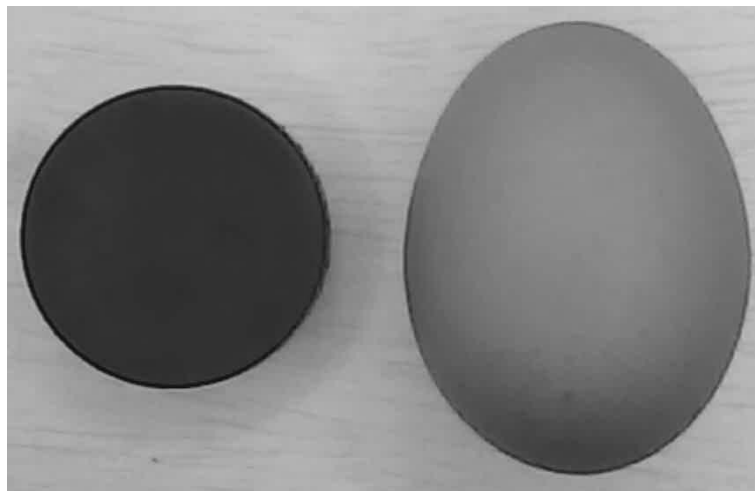
A continuación, se detalla lo antes mencionado.

### 1. Conversión a escala de grises

En OpenCv la función **cvtColor** permite convertir la imagen RGB a escala de grises (ver Figura 2.4.), con esto se obtiene una imagen unidimensional que disminuye la complejidad de procesamiento.

#### Figura 2.4.

*Imagen en escala de grises*

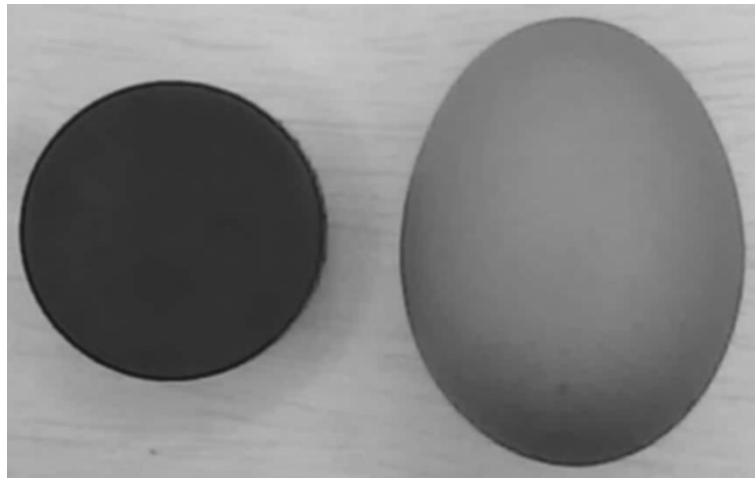


## 2. Filtro gaussiano

En OpenCV los filtros gaussianos permiten eliminar diferentes tipos de ruido que puede presentar la imagen después de la etapa de adquisición. Para el caso propuesto se usa la función **cv2.GaussianBlur ()** que sirve para realizar un suavizado a la imagen, con la finalidad de realizar un desenfoque que permite eliminar los pixeles atípicos que por lo general es ruido presente en la imagen (ver Figura 2.5.).

### Figura 2.5.

*Imagen con filtro gaussiano*

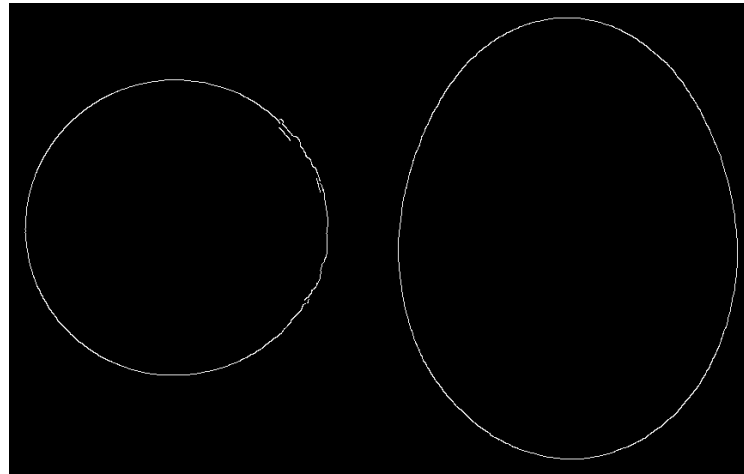


## 3. Detección de bordes de Canny

Después de convertir la imagen a escala de grises y eliminar el ruido con un filtro Gaussiano, se procede a detectar los distintos bordes (ver Figura 2.6.). En OpenCV la función **cv2.Canny()** sirve para detectar los bordes que se encuentran dentro de la imagen y así establecer los límites a los que van a estar sujetos los objetos dentro de la imagen [41].

**Figura 2.6.**

*Detección de bordes mediante Canny*

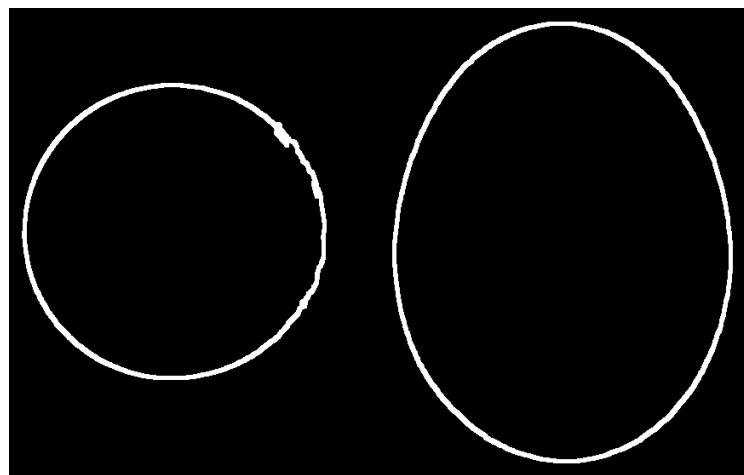


#### **4. Filtro de dilatación**

En OpenCV el filtro `cv2.dilate()` se utiliza para ampliar la imagen, es decir, sirve para aumentar el área del objeto y por lo general se utiliza para acentuar las características. Para el caso de estudio este filtro se utiliza para poder cerrar las posibles aperturas que pueden existir en los bordes que fueron detectados en la instancia anterior (ver Figura 2.7.)

**Figura 2.7.**

*Imagen con filtro de dilatación*



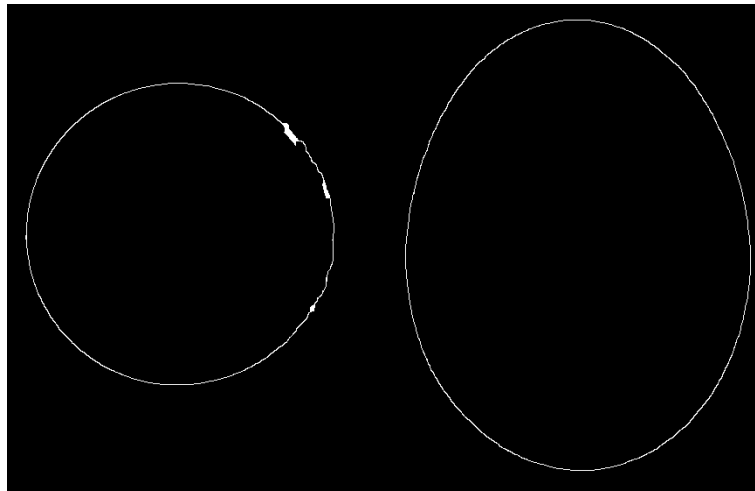


## 5. Filtro de erosión

En OpenCV el filtro **cv2.erode()** tiene como objetivo eliminar los ruidos que aun puedan existir, así también, sirve para separar dos objetos que pueden estar conectados después de haber aplicado el filtro de dilatación (ver Figura 2.8.) [42].

### Figura 2.8.

*Imagen con filtro de erosión*



### 2.3. Detección de contornos

Después de procesar la imagen y detectar los bordes de los objetos que se encuentran en la imagen se procede a detectar los contornos. Para ello simplemente se utiliza la función que facilita encontrar contornos tomando en cuenta lo siguiente:

**cv2.findContour()** posee tres argumentos:

- El primero hace referencia a la imagen de donde se pretende encontrar los distintos contornos.

- El segundo es la forma de recuperación de contorno, que para este caso se utilizará **cv2.RET\_EXTERNAL**, ya que únicamente se va a hacer uso de los contornos externos, es decir, obviando aquellos contornos que se encuentran dentro de los objetos.
- El tercero es la forma de aproximación de contorno, que en este caso será **cv2.CHAIN\_APPROX\_SIMPLE**, que comprime el contorno de manera que únicamente toma como referencia de contorno los puntos finales de los objetos.

### 2.3.1. Orden de los contornos

Una vez detectados los contornos se procede a ordenar los mismos de manera que se distribuyan de izquierda a derecha y de arriba hacia abajo, esto con la finalidad de establecer en el algoritmo que el objeto de referencia que ayudará a obtener el factor de conversión de pixeles a centímetros se encuentre siempre en la parte izquierda.

Esto se logra con la ayuda de la librería **imutils** que se explica en la Sección 2.6. de este Capítulo. Al hacer uso de esta librería de acuerdo con OpenCV únicamente se debe llamar y utilizar la función:

```
imutils.contours.sort_contours()
```

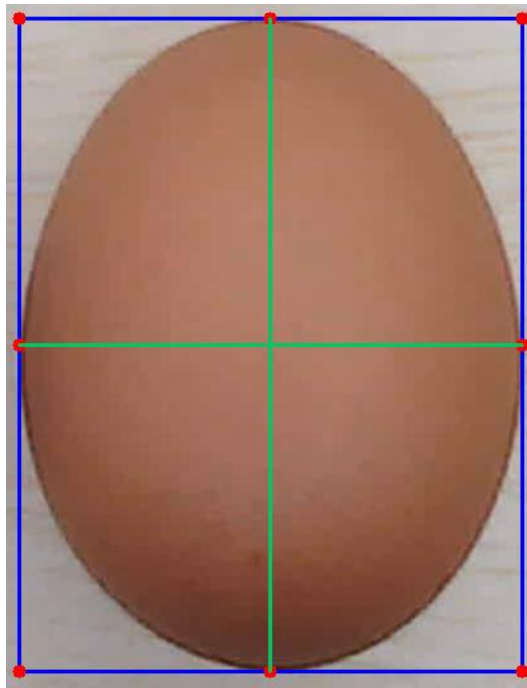
Esto genera que los contornos se ordenen de izquierda a derecha y de arriba hacia abajo; de esta manera se establece en el algoritmo que el objeto se encuentre en la parte izquierda siempre será el objeto de referencia.

## 2.4. Análisis de dimensiones

Una vez que se ha detectado los contornos que posee la imagen, se procede a dibujar un cuadro delimitador para que bordee los objetos que se encuentran en la imagen. Así también, se determinan y dibujan los puntos centrales de cada uno de los segmentos del cuadro, obteniendo como resultado la Figura 2.9.

### Figura 2.9.

*Cuadro que bordea al objeto con sus respectivos puntos medios*



Como se puede observar en la Figura 2.9. al dibujar el cuadro delimitador con sus respectivos puntos medios se puede apreciar que es posible calcular las variables de ancho y alto únicamente con el procesamiento que se ha realizado hasta el momento, sin embargo, al realizar dicho cálculo no se obtendría un resultado favorable debido a que se necesita un factor de conversión de píxeles a centímetros.

### 2.4.1. Análisis del tamaño en píxeles

Para conocer las dimensiones de las variables de ancho y alto, representadas en la Figura 2.9. con la línea vertical y horizontal respectivamente, se procede a determinar la distancia euclidiana que representa la distancia que existe entre los puntos de las variables; de forma matemática esto se obtiene al aplicar la siguiente formula:

$$d(ptoA,ptoB) = \sqrt{(ptoB - ptoA)^2} \quad (2.1)$$

*d*: Separación que tienen dos puntos en un espacio

*ptoA*: Punto inicial de la recta

*ptoB*: Punto final de la recta

En OpenCV esto se logra al usar la librería **SciPy** que se explica en la Sección 2.6. de este Capítulo, es así que mediante el comando **dist.euclidean()**, se calcula la distancia euclidiana de las variables del ancho y alto, aunque dichas dimensiones se proporcionan en píxeles.

Una vez calculada la distancia euclidiana de las variables buscadas, se procede a utilizar el objeto de referencia del que se tiene las dimensiones para poder obtener el factor de conversión, para ello se aplica la siguiente formula:

$$pixelPerMetric = \frac{dB}{ancho} \quad (2.2)$$

Donde:

*dB*: representa la distancia euclidiana del ancho del objeto de referencia en píxeles [*px*]

*ancho*: representa la dimensión del objeto de referencia [*cm*]

Con el factor de conversión de pixeles a centímetros establecido se procede a calcular las dimensiones mediante la siguiente formula:

$$Alto = \frac{dA}{pixelsPerMetric} \quad (2.3)$$

$$Ancho = \frac{dB}{pixelsPerMetric} \quad (2.4)$$

Donde:

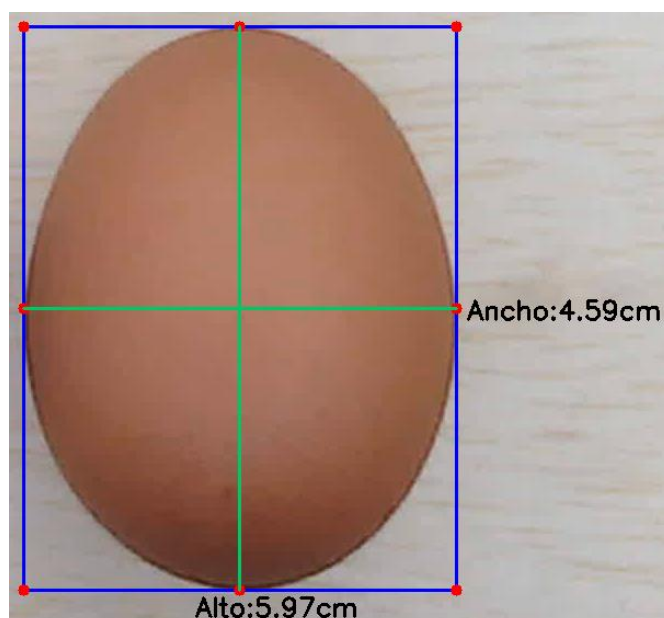
$dA$  y  $dB$ : representa a la distancia euclideana del alto y ancho respectivamente [px]

$pixelsPerMetric$ : factor de conversión  $\left[\frac{px}{cm}\right]$

Al aplicar los distintos cálculos se obtiene como resultado la Figura 2.10. que muestra las dimensiones del huevo de gallina.

**Figura 2.10.**

*Variables encontradas de ancho y alto*



## 2.5. Cálculo del peso

Una vez que se calcularon las variables geométricas de alto y ancho mediante el procesamiento de imagen, se hace uso de dichos parámetros para la obtención del peso del producto, para ello tomamos como referencia la fórmula proporcionada en [43], la cual ayuda a obtener el volumen del huevo.

$$Volumen = \pi * \frac{dimA * dimB^2}{6}$$

*dimA*: Representa el alto [*cm*]

*dimB*: Representa el ancho [*cm*]

Con el volumen establecido se procede a calcular la masa mediante la fórmula de la densidad, al realizar el respectivo despeje se obtiene la siguiente formula:

$$masa = v * \rho \tag{2.5}$$

Donde:

*v*: Volumen en [*cm*<sup>3</sup>]

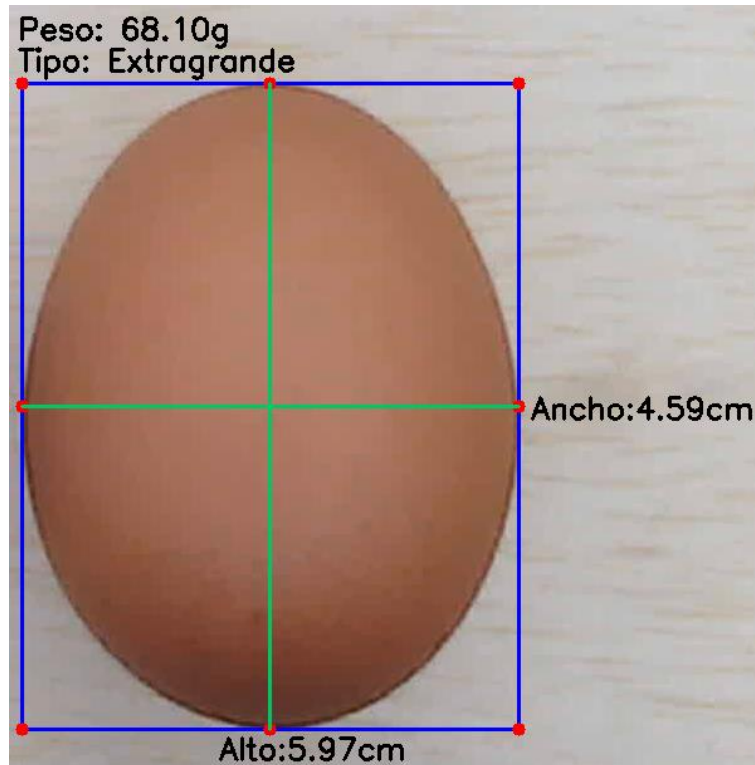
$\rho$ : Densidad en [ $\frac{gr}{cm^3}$ ]

Cabe mencionar que la densidad que se utiliza para el cálculo de la masa es la que se proporciona en [34], que es de 1.035 [ $\frac{gr}{cm^3}$ ].

En la Figura 2.11. se puede observar el resultado de implementar el proceso mencionado con anterioridad y la clasificación a la que pertenece tomando en cuenta la Tabla 1.1.

**Figura 2.11.**

*Peso y clasificación proporcionados por el algoritmo*



## **2.6. Librerías utilizadas**

En el desarrollo de cualquier sistema de visión artificial es de suma importancia el uso de librerías ya que con su ayuda se puede simplificar y facilitar la resolución del problema que se pretende solucionar.

A continuación, se detallan las diferentes librerías utilizadas durante el desarrollo del sistema de visión artificial.

### **2.6.1. NumPy**

NumPy es una librería de Python que proporciona la facilidad de realizar operaciones rápidas con matrices. Su uso se debe principalmente a que la imagen es una matriz N-dimensional que posee píxeles como datos y puede ser modificada con operaciones numéricas básicas.

Además, cuenta con rutinas que permiten realizar operaciones matemáticas, lógicas, multiplicación de vectores, normalización, selección, simulación aleatoria, manipulación de formas, que son necesarias para realizar tareas como alinear, deformar, clasificar y agrupar imágenes [44].

El módulo Numpy se importa como: `import numpy as np`

### **2.6.2. Imutils**

Imutils es una librería de Python que proporciona una serie de funciones convenientes para hacer que las funciones básicas de procesamiento de imágenes, como traducción, rotación, cambio de tamaño, esqueletización, visualización de imágenes de Matplotlib, clasificación de contornos, detección de bordes, entre otras, sean más fácil en OpenCV.

Así también cuenta con submódulos como `imutils.contours` que permite clasificar y etiquetar los contornos de la imagen, esta función acepta dos parámetros que son los contornos y los arreglos de clasificación.

El módulo Imutils se importa como: `import imutils`

El submódulo se importa como: `from imutils import contours`

### **2.6.3. SciPy**

SciPy es una librería matemática que se basa en NumPy, sin embargo, aunque pueda realizar rutinas relacionadas, esta librería en el procesamiento de imágenes permite realizar arreglos y modificaciones que trabajan en matrices Numpy de 'n' dimensiones con lo que se puede realizar filtrados, interpolaciones, morfologías y mediciones de objetos.



Así también, cuenta con submódulos que permiten realizar tareas específicas como el caso del submódulo `scipy.spatial.distance` que permite realizar el cálculo de la matriz de distancia a partir de vectores de observación que se encuentran almacenados en una matriz rectangular[45].

El módulo SciPy se importa como: `import scipy`

El submódulo se importa como: `from scipy.spatial import distance`

#### **2.6.4. Math**

Math es una librería de Python que se utiliza para agregar funciones y constantes matemáticas como funciones trigonométricas seno, coseno y tangente, así también sirve para hacer uso de constantes matemáticas.

El módulo Math se importa como: `import math`

## Capítulo 3

### Resultados

El algoritmo que se puede ver a detalle en el Anexo de este documento se desarrolló en la interfaz gráfica Anaconda Navigator con la ayuda del software Spyder.

El funcionamiento del algoritmo se comprobó al realizar el análisis de 21 huevos de gallina seleccionados al azar. Durante la prueba se tomó nota del peso y la clasificación que proporcionó el algoritmo para posteriormente verificar si la clasificación realizada fue la correcta tomando en cuenta la Tabla 1.1.

La Tabla 3.1. muestra el peso y la categoría respectiva a la que pertenecen los 21 huevos de gallina, el dato del peso se tomó con la ayuda de una balanza con precisión de 1g y para clasificar el producto en su respectiva categoría se utilizó la Norma Técnica Ecuatoriana NTE INEN 1973. Cabe mencionar que para realizar el pesaje no se utilizó ningún elemento estabilizador que ayude a mantener el producto en una posición fija.

**Tabla 3.1.**

*Peso y clasificación realizados de forma manual*

<b>N° de muestra</b>	<b>Peso de acuerdo con la balanza [g]</b>	<b>Categoría según la norma NTE INEN 1973</b>
1	56	Mediano
2	63	Grande
3	60	Grande
4	66	Extragrande
5	73	Gigante
6	67	Extragrande
7	63	Grande
8	64	Extragrande
9	76	Supergigante
10	84	Supergigante
11	69	Extragrande
12	72	Gigante
13	73	Gigante

14	67	Extragrande
15	57	Mediano
16	56	Mediano
17	77	Supergigante
18	57	Mediano
19	75	Gigante
20	77	Supergigante
21	62	Grande

La Tabla 3.2. muestra los datos que se obtuvieron al realizar el pesaje y clasificación del producto a través del algoritmo desarrollado. Cabe mencionar que durante la prueba del algoritmo se realizó la toma de 3 datos referentes al peso para observar cómo puede llegar a variar.

**Tabla 3.2.**

*Peso y clasificación proporcionados por el algoritmo*

<b>N° de muestra</b>	<b>Primer peso [g]</b>	<b>Segundo peso [g]</b>	<b>Tercer peso [g]</b>	<b>Promedio [g]</b>	<b>Categoría</b>
1	53,13	53,5	53,82	53,48	Mediano
2	60,91	60,59	60,91	60,80	Grande
3	57,75	57,82	57,54	57,70	Mediano
4	64,72	64,53	64,71	64,65	Extragrande
5	72,85	72,63	72,28	72,59	Gigante
6	65,21	65,07	65,3	65,19	Extragrande

7	60,38	60,69	60,89	60,65	Grande
8	60,92	61,05	60,97	60,98	Grande
9	77,01	76,94	76,86	76,94	Supergigante
10	84,53	84,44	84,29	84,42	Supergigante
11	68,66	67,9	68,17	68,24	Extragrande
12	71,65	72,06	71,86	71,86	Gigante
13	71,49	71,28	70,89	71,22	Gigante
14	66,23	66,56	66,9	66,56	Extragrande
15	53,56	53,67	53,44	53,56	Mediano
16	52,78	52,89	52,71	52,79	Mediano
17	76,73	76,64	76,78	76,72	Supergigante
18	53,7	53,81	53,64	53,72	Mediano
19	73,61	73,82	73,75	73,73	Gigante
20	76,59	76,51	76,73	76,61	Supergigante
21	59,09	59,14	59,21	59,15	Grande

---

### 3.1. Análisis de resultados

La eficiencia del algoritmo se obtiene al comparar los resultados obtenidos al realizar las pruebas del algoritmo con la norma técnica para ovoproductos.

Para conocer el porcentaje de error con respecto a la clasificación, se hace una comparación entre la Tabla 3.1. y Tabla 3.2. Es así que comparando los resultados obtenidos mediante el algoritmo con los datos obtenidos de la balanza en cuanto a clasificación existe un 90,47% de eficiencia, tal como se puede observar en la Tabla 3.3.

Por otra parte, con respecto a la estimación del peso del producto el algoritmo tiene un 97,32% de eficiencia, lo que da a entender que el algoritmo está cerca del valor real con un 2,68% de falla en cuanto a la estimación. Esto se puede apreciar en la Tabla 3.4. y Tabla 3.5. donde se analiza el porcentaje de error del pesaje realizado de manera general y por categoría respectivamente, con la finalidad de conocer que categoría presenta un mayor porcentaje de error al momento de estimar el peso.

**Tabla 3.3.**

*Eficiencia de clasificación con el algoritmo*

<b>Muestras</b>	21
<b>Coincidencias</b>	19
<b>Error</b>	2
<b>Eficiencia de clasificación con el algoritmo</b>	90,47%

**Tabla 3.4.***Porcentaje de error en la estimación del peso*

<b>Muestra</b>	<b>Balanza</b>	<b>Promedio</b>	<b>Error con un peso (Primero)</b>	<b>Error general (Promedio)</b>
1	56	53,48	5,13%	4,49%
2	63	60,80	3,32%	3,49%
3	60	57,70	3,75%	3,83%
4	66	64,65	1,94%	2,04%
5	73	72,59	0,21%	0,57%
6	67	65,19	2,67%	2,70%
7	63	60,65	4,16%	3,72%
8	64	60,98	4,81%	4,72%
9	76	76,94	1,33%	1,23%
10	84	84,42	0,63%	0,50%
11	69	68,24	0,49%	1,10%
12	72	71,86	0,49%	0,20%
13	73	71,22	2,07%	2,44%
14	67	66,56	1,15%	0,65%
15	57	53,56	6,04%	6,04%
16	56	52,79	5,75%	5,73%
17	77	76,72	0,35%	0,37%
18	57	53,72	5,79%	5,76%
19	75	73,73	1,85%	1,70%

20	77	76,61	0,53%	0,51%
21	62	59,15	4,69%	4,60%
<b>Porcentaje de error</b>			2,72%	2,68%

**Tabla 3.5.**

*Porcentaje de error por categoría*

<b>Categoría</b>	<b>Error al estimar el peso</b>
Mediano	5,5%
Grande	3,9%
Extragrande	2,24%
Gigante	1,2%
Supergigante	0,7%
Error total	2,68%



# Capítulo 4

## Conclusiones y trabajo futuro

### 4.1. Conclusiones

- Se determinaron los requisitos y requerimientos del sistema para que el algoritmo diseñado funcione de manera óptima y arroje los resultados más aproximados a la realidad. Para ello se estableció que la escena de pruebas sea en un lugar cerrado en la que se monten los equipos mencionados y así evitar alteraciones.
- Se determina que la iluminación para la adquisición de las imágenes debe ser de manera frontal ya que así se logra resaltar las características de forma y contorno del objeto procesado (huevo) reduciendo de esta manera la alteración en los resultados.
- El algoritmo diseñado toma en cuenta las variables geométricas de ancho y alto de las imágenes de los huevos en 2D, para ello se utilizó los siguientes pasos: Adquisición de la imagen, preprocesamiento, detección de contornos, análisis de dimensiones y cálculo del peso.
- La validación del algoritmo se realizó con el pesaje de 21 huevos de gallina seleccionados al azar. La clasificación se realizó de acuerdo con la norma técnica NTE

INEN 1973. Los resultados muestran que existe un 90,47% de eficiencia en la clasificación y un 97,32% de eficiencia en la estimación del peso.

## **4.2. Trabajos futuros**

Para ampliar y mejorar el estudio que se ha desarrollado a lo largo de este documento se propone como trabajo futuro lo siguiente:

- Realizar una aplicación para el celular que pueda ser utilizado para clasificar los huevos en cualquier ambiente.
- Simular una línea de producción en la que los huevos de gallina sean separados de acuerdo con su clasificación de manera automática.
- Desarrollar una interfaz de usuario que permita facilitar el uso del prototipo.
- Desarrollar un algoritmo que permita realizar un control de calidad de los huevos de gallina y clasificarlos según el grado de defecto que posee el cascaron.

## Referencias

- [1] R. Aguirre and M. J. Pizarro, “Panorama y mercado del huevo,” *Odepa*, pp. 1–12, 2018. [Online]. Available: <https://www.odepa.gob.cl/wp-content/uploads/2018/04/Huevos.pdf>
- [2] S. Avícolas, “La producción mundial de huevos: situación actual y previsiones,” *Selecciones Avícolas*. <https://seleccionesavicolas.com/avicultura/2016/01/la-produccion-mundial-de-huevos-situacion-actual-y-previsiones>
- [3] D. general de políticas Agrarias, “Panorama y mercado del huevo de gallina,” pp. 1–9, 2019. [Online]. Available: [https://cdn.www.gob.pe/uploads/document/file/438987/panorama\\_\\_mercado\\_huevo\\_de\\_gallina.pdf](https://cdn.www.gob.pe/uploads/document/file/438987/panorama__mercado_huevo_de_gallina.pdf)
- [4] M. y Soya, “Cada año en Ecuador aumenta consumo de pollo y huevos,” *Maiz y Soya*. <http://www.maizysoya.com/lector.php?id=20200548&tabla=articulos>
- [5] A. Sánchez, T. Vayas, F. Mayorga, and C. Freire, “Sector Avícola Ecuador,” *Inec - Espac*, pp. 1–4, 2019. [Online]. Available: <https://blogs.cedia.org.ec/obest/wp-content/uploads/sites/7/2020/09/Sector-avicola-Ecuador.pdf>
- [6] R. León, A. Boy, L. Peláez, and L. Tacanga, “Diseño y construcción de un prototipo de

- máquina selladora y clasificadora de huevos de gallina por peso, en la empresa peruana Aviporc S.A.C,” *S.A.C. Ing. Ind.*, no. 038, pp. 217–234, Jul. 2020, doi: 10.26439/ing.ind2020.n038.4822.
- [7] D. Velazco and M. Pinto, “Caracterización Del Huevo De Gallina Para El Diseño De Un Sistema Automático De Clasificación,” *Rev. Investig. Desarro. Innov*, vol. 3, pp. 33–43, 2012, [Online]. Available: [https://revistas.uptc.edu.co/index.php/investigacion\\_duitama/article/view/2136](https://revistas.uptc.edu.co/index.php/investigacion_duitama/article/view/2136)
- [8] P. Toalombo, A. Villafuerte, J. Benavidez, and E. Oleas, “Caracterización Del Mercado De Huevo Comercial (Gallina Lohmann Brown) Versus El Huevo Criollo (Gallina De Campo) En La Provincia De Tungurahua,” *Fac. Ciencias Pecu.*, vol. 3, no. 2, pp. 13–22, 2016.
- [9] INEN 1973, “Norma Técnica Ecuatoriana NTE INEN 1973. HUEVOS COMERCIALES Y OVOPRODUCTOS, REQUISITOS Segunda revisión,” *Instituto Ecuatoriano De Normalización*, p. 19, 2013. [Online]. Available: <https://www.normalizacion.gob.ec/buzon/normas/1973-2.pdf>
- [10] MOBA, “MOBANETTE3 | MOBA68/88.” [Online]. Available: [moba.net/page/es/Products/Detail/mobanette3/967](https://moba.net/page/es/Products/Detail/mobanette3/967)
- [11] “Características Técnicas de la Clasificadora de Huevos Yemita,” *Yemita clasificadora de huevos*. [https://www.clasificadorayemita.com/post/características-técnicas-de-la-clasificadora-de-huevos-yemita#:~:text=- Capacidad de clasificación%3A 3600 huevos,HP\\* Funciona con electricidad 220v.](https://www.clasificadorayemita.com/post/características-técnicas-de-la-clasificadora-de-huevos-yemita#:~:text=- Capacidad de clasificación%3A 3600 huevos,HP* Funciona con electricidad 220v.)
- [12] D.-W. Sun, “Computer vision—an objective, rapid and non-contact quality evaluation tool for the food industry,” *J. Food Eng.*, vol. 61, no. 1, pp. 1–2, Jan. 2004, doi: 10.1016/S0260-

8774(03)00182-1.

- [13] J. C. García Monsálvez, “Python como primer lenguaje de programación textual en la Enseñanza Secundaria,” *Educ. Knowl. Soc.*, vol. 18, no. 2, pp. 147–162, Aug. 2017, doi: 10.14201/eks2017182147162.
- [14] J. Troyano, F. Cruz, M. Gonzáles, C. Vallejo, and T. Miguel, “Introducción a la Programación con Python, Computación Interactiva y Aprendizaje Significativo,” *Dep. Lenguajes y Sist. Informáticos*, vol. 3, pp. 223–230, 2018, doi: 10.6035/Sapientia93.
- [15] I. Challenger Pérez, Y. Díaz Ricardo, and R. Becerra García, “El lenguaje de programación Python/The programming language Python,” *Rev. Ciencias Holguín*, vol. 20, pp. 1–13, 2014, doi: <http://www.redalyc.org/articulo.oa?id=181531232001>.
- [16] A. Zelinsky, *Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008)[On the Shelf]*, vol. 16, no. 3. 2009. doi: 10.1109/MRA.2009.933612.
- [17] S. Amaya-Zapata, D. Pulgarín-Velásquez, and Í. D. Torres-Pardo, “Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM),” *Lámpsakos*, vol. 15, no. 15, pp. 43–50, Mar. 2016, doi: 10.21501/21454086.1702.
- [18] C. V. Niño Rondón, S. A. Castro Casadiego, and B. Medina Delgado, “Caracterización para la ubicación en la captura de video aplicado a técnicas de visión artificial en la detección de personas,” *Rev. Colomb. Tecnol. Av.*, vol. 2, no. 36, pp. 83–88, Oct. 2020, doi: 10.24054/16927257.v36.n36.2020.24.
- [19] D. C. Ramírez González, G. Pulido Sarmiento, B. Gerardino Arévalo, J. M. Cruz Romero,

- E. Estupiñán Escalante, and S. Cancino Suárez, “Adquisición y Reconocimiento de Imágenes por Medio de Técnicas de Visión e Inteligencia Artificial,” *ITECKNE*, vol. 6, no. 1, pp. 5–13, Jun. 2009, doi: 10.15332/iteckne.v6i1.290.
- [20] J. Reyes-Reyes, “Construcción de una Interfaz Eléctrica de Salida para Sistemas de Visión Artificial Basados en MATLAB Construcción de una Interfaz Eléctrica de Salida para Sistemas de Visión Artificial Basados en MATLAB Introducción,” *3er Congr. Cómputo AGECOMP, UAEM, México*, no. August, pp. 46–53, 2015.
- [21] G. A. Figueredo-Ávila and J. A. Ballesteros-Ricaurte, “Identificación del estado de madurez de las frutas con redes neuronales artificiales, una revisión,” *Cienc. y Agric.*, vol. 13, no. 1, pp. 117–132, Jan. 2016, doi: 10.19053/01228420.4811.
- [22] V. Alvear-Puertas, P. Rosero-Montalvo, D. Peluffo-Ordóñez, and J. Pijal-Rojas, “Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura,” *Enfoque UTE*, vol. 8, no. 1, pp. 244–256, Feb. 2017, doi: 10.29019/enfoqueute.v8n1.121.
- [23] W. Donat, “Introducing Python,” in *Learn Raspberry Pi Programming with Python*, Berkeley, CA: Apress, 2018, pp. 55–89. doi: 10.1007/978-1-4842-3769-4\_3.
- [24] A. Marzal Varó, I. Gracia Luengo, and P. García Sevilla, *Introducción a la programación con Python 3*. Universitat Jaume I, 2014. doi: 10.6035/Sapientia93.
- [25] V. G. Narendra and K. S. Hareesh, “Prospects of Computer Vision Automated Grading and Sorting Systems in Agricultural and Food Products for Quality Evaluation,” *Int. J. Comput. Appl.*, vol. 1, no. 4, pp. 1–12, Feb. 2010, doi: 10.5120/111-226.
- [26] S. Janardhana, J. Jaya, K. J. Sabareesaan, and J. George, “Computer aided inspection system

- for food products using machine vision; A review,” in *2013 International Conference on Current Trends in Engineering and Technology (ICCTET)*, Jul. 2013, pp. 29–33. doi: 10.1109/ICCTET.2013.6675906.
- [27] M. J. Delgado-Gutiérrez, D. F. Herrera-Guillén, L. M. Medina-Barragán, and J. P. Corredor-Gómez, “Implementación de un sistema de procesamiento de imágenes integrado con Raspberry PI 2B para reconocimiento y recolección de fresas maduras,” *Rev. Politécnica*, vol. 13, no. 25, pp. 75–85, Jun. 2017, doi: 10.33571/rpolitec.v13n25a6.
- [28] A. Nogue and J. Antiga, “Aplicación práctica de la visión artificial en el control de procesos industriales Conocimientos básicos de visión artificial,” pp. 1–28, 2012. [Online]. Available: <http://visionartificial.fpcat.cat/wp-content/uploads/Conocimientos.pdf>
- [29] M. Lesser, “Charge coupled device (CCD) image sensors,” in *High Performance Silicon Imaging*, no. Ccd, Elsevier, 2014, pp. 78–97. doi: 10.1533/9780857097521.1.78.
- [30] C. Features, “FaceCam 1000X,” pp. 1–2, 2020. [Online]. Available: <https://us.geniusnet.com/product/facecam-1000x/>
- [31] D. Norman, P. Malewadkar, S. Naik, and F. Carvalho, “Eggs Defect Detection using Image Processing,” *Int. J. Eng. Res.*, vol. V6, no. 07, pp. 91–93, Jul. 2017, doi: 10.17577/IJERTV6IS070075.
- [32] R. Mota-Grajales *et al.*, “Defect detection in eggshell using a vision system to ensure the incubation in poultry production,” *Measurement*, vol. 135, pp. 39–46, Mar. 2019, doi: 10.1016/j.measurement.2018.09.059.
- [33] A. S. Alon, J. Marasigan, Rufo I., J. Nicolas- Mindoro, and C. D. Casuat, “An Image



- Processing Approach of Multiple Eggs' Quality Inspection," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 2794–2799, Dec. 2019, doi: 10.30534/ijtcse/2019/18862019.
- [34] M. Soltani, M. Omid, and R. Alimardani, "Egg volume prediction using machine vision technique based on pappus theorem and artificial neural network," *J. Food Sci. Technol.*, vol. 52, no. 5, pp. 3065–3071, May 2015, doi: 10.1007/s13197-014-1350-6.
- [35] J. Alikhanov, S. M. Penchev, T. D. Georgieva, A. Moldazhanov, Z. Shynybay, and P. I. Daskalov, "An indirect approach for egg weight sorting using image processing," *J. Food Meas. Charact.*, vol. 12, no. 1, pp. 87–93, Mar. 2018, doi: 10.1007/s11694-017-9619-x.
- [36] P. Javadikia, M. H. Dehrouyeh, L. Naderloo, H. Rabbani, and A. N. Lorestani, "Measuring the Weight of Egg with Image Processing and ANFIS Model," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7076 LNCS, no. PART 1, 2011, pp. 407–416. doi: 10.1007/978-3-642-27172-4\_50.
- [37] D. Escobar Figueroa and E. Roa Guerrero, "Sistema de visión artificial para la identificación del estado de madurez de frutas (granadilla)," *Redes Ing.*, vol. 7, no. 1, pp. 84–92, Jun. 2016, doi: 10.14483/udistrital.jour.redes.2016.1.a08.
- [38] I. Nyalala *et al.*, "Weight and volume estimation of single and occluded tomatoes using machine vision," *Int. J. Food Prop.*, vol. 24, no. 1, pp. 818–832, Jan. 2021, doi: 10.1080/10942912.2021.1933024.
- [39] A. Nogue and J. Antiga, "Aplicación práctica de la visión artificial en el control de procesos industriales," pp. 1–26, 2012. [Online]. Available: <http://www.adimenlehiakorra.eus/documents/29934/43025/Aplicación+práctica+de+la+vi>

sión+artificial+en+el+control+de+procesos+industriales.pdf/92a1a455-ffe2-4290-be95-f9ba33c02241#:~:text=Con la visión artificial se pueden%3A&text=Automatizar tareas repetit

- [40] Á. P. Fernández Soria, “Sistema Electrónico Para El Control De Calidad De Huevos De Gallina Mediante Procesamiento De Imágenes.,” Universidad Técnica de Ambato, 2018. [Online]. Available: <http://repositorio.uta.edu.ec/jspui/handle/123456789/28950>
- [41] J. Howse, “OpenCV Computer Vision with Python,” *Cs\_Python\_in*, p. 121, 2013, [Online]. Available: [www.it-ebooks.info](http://www.it-ebooks.info)
- [42] A. González Marcos *et al.*, *Técnicas y algoritmos básicos de visión artificial*. 2006. [Online]. Available: <https://publicaciones.unirioja.es/catalogo/online/VisionArtificial.pdf>
- [43] R. Majid, M. Mohammad, and G. Mohammad, “Egg Volume Determination by Spheroid Approximation and Image Processing,” *J. Food Sci. Technol.*, vol. 52, no. 5, pp. 590–596, 2015, doi: 10.1007/s13197-014-1350-6.
- [44] NumPy-community, *NumPy User Guide v1.3*. 2022. [Online]. Available: <https://numpy.org/doc/stable/numpy-user.pdf>
- [45] J. E. Solem, *Programing Computer Vision with Python*. O’Reilly Media, 2012. [Online]. Available: <https://www.oreilly.com/library/view/programming-computer-vision/9781449341916/>

# Anexo

## Código del programa

```
# Algoritmo para la estimacion del peso de huevos de
# gallina (clasificacion de acuerdo con su peso)

# Importacion librerias a utilizar
import cv2
import numpy as np
import imutils
from scipy.spatial import distance as dist
from imutils import perspective
from imutils import contours
import math

# Se obtiene la imagen en tiempo real mediante la webcam
cap = cv2.VideoCapture(2)
cap.set(10,160)
cap.set(3,1280)
cap.set(4,720)

# Valor de la densidad del huevo a utilizar
# y ancho del objeto de referencia
ancho = 41
density = 1.035

# Funcion para encontrar puntos medios
def midpoint(ptA, ptB):
    return ((ptA[0] + ptB[0]) * 0.5, (ptA[1] + ptB[1]) * 0.5)

# Inicio del programa
```

```

while True:
    # Filtros para mejorar la imagen que se obtiene, ademas se encuentra los bordes
    # y despues se encuentran los contornos para ordenarlos de izquierda a derecha
    # y de arriba hacia abajo
    success,img=cap.read()
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (7, 7), 0)
    imgCanny = cv2.Canny(imgBlur, 50,100)
    kernel = np.ones((5,5))
    imgDial = cv2.dilate(imgCanny, kernel, iterations=1)
    imgThr = cv2.erode(imgDial, kernel, iterations=1)
    cnts = cv2.findContours(imgThr.copy(), cv2.RETR_EXTERNAL,
                            cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    (cnts,_) = contours.sort_contours(cnts)
    pixelsPerMetric = None

    for c in cnts:
        if cv2.contourArea(c) <100:
            continue

        # Se dibuja el cuadro delimitador de los contornos detectados
        orig = img.copy()
        box = cv2.minAreaRect(c)
        box = cv2.cv.BoxPoints(box) if imutils.is_cv2() else cv2.boxPoints(box)
        box = np.array(box, dtype='int')
        box = perspective.order_points(box)
        cv2.drawContours(orig, [box.astype("int")], -1, (255, 0, 0), 2)
        for (x, y) in box:
            cv2.circle(orig, (int(x), int(y)), 5, (0, 0, 255), -1)

        # Se encuentra los puntos centrales de los segmentos del cuadro delimitador
        (tl, tr, br, bl) = box
        (tltrX, tltrY) = midpoint(tl, tr)
        (blbrX, blbrY) = midpoint(bl, br)
        (tlblX, tlblY) = midpoint(tl, bl)
        (trbrX, trbrY) = midpoint(tr, br)

        # Se dibuja los puntos medios y las lineas que pasan a travez de ellos
        cv2.circle(orig, (int(tltrX), int(tltrY)), 5, (0, 0, 255), -1)
        cv2.circle(orig, (int(blbrX), int(blbrY)), 5, (0, 0, 255), -1)
        cv2.circle(orig, (int(tlblX), int(tlblY)), 5, (0, 0, 255), -1)
        cv2.circle(orig, (int(trbrX), int(trbrY)), 5, (0, 0, 255), -1)

        cv2.line(orig, (int(tltrX), int(tltrY)), (int(blbrX), int(blbrY)),
                (100, 200, 0), 2)

```

```

cv2.line(orig, (int(tlbrX), int(tlbrY)), (int(trbrX), int(trbrY)),
         (100, 200, 0), 2)

# Calculo de la distancia euclidean
dA = dist.euclidean((tltrX, tltrY), (blbrX, blbrY))
dB = dist.euclidean((tlblX, tlblY), (trbrX, trbrY))

# Calculo del factor de conversion de pixeles a cm
if pixelsPerMetric is None:
    pixelsPerMetric = dB/ancho

# Calculo de las dimensiones con la constante encontrada
dimA = (dA / pixelsPerMetric)/10 #Alto
dimB = (dB / pixelsPerMetric)/10 #Ancho

# Calculo del volumen y de la masa
vol = (math.pi*(dimA*(dimB*dimB)))/6
#print(vol)
mass = vol*density
#print(density)

# Se dibuja las dimensiones y el peso encontrado
cv2.putText(orig, "Alto: {:.2f}cm".format(dimA),
            (int(blbrX - 40), int(blbrY + 24)), cv2.FONT_HERSHEY_SIMPLEX,
            0.8, (0, 0, 0), 2)

cv2.putText(orig, "Ancho: {:.2f}cm".format(dimB),
            (int(trbrX + 10), int(trbrY + 10)), cv2.FONT_HERSHEY_SIMPLEX,
            0.8, (0, 0, 0), 2)

if mass>45:
    cv2.putText(orig, "Peso: {:.2f}gr".format(mass),
                (int(tltrX - 200), int(tltrY - 35)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

# Se dibuja las categorias una vez que se ha encontrado el peso
if mass >= 50 and mass <58:
    cv2.putText(orig, "Tipo: Mediano",
                (int(tltrX - 200), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

if mass >= 58 and mass <64:
    cv2.putText(orig, "Tipo: Grande",
                (int(tltrX - 200), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

```

```

if mass >= 64 and mass <70:
    cv2.putText(orig, "Tipo: Extragrande",
                (int(tltrX - 200), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

if mass >= 70 and mass <76:
    cv2.putText(orig, "Tipo: Gigante",
                (int(tltrX - 200), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

if mass >= 76:
    cv2.putText(orig, "Tipo: SuperGigante",
                (int(tltrX - 200), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,
                0.8, (0, 0, 0), 2)

cv2.imshow('Img',orig)

key = cv2.waitKey(1)
if key == 27:
    break

cap.release()
cv2.destroyAllWindows()

```

