



# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACIÓN**

“Detección de galletas que no cumplen con el estándar de la línea de producción, utilizando reconocimiento de formas y análisis morfológico”

## **INFORME DE MATERIA DE GRADUACIÓN**

Previo a la obtención del Título de:

## **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

Cleotilde Albertina Naranjo Chano

María Fernanda Rivera Jara

GUAYAQUIL – ECUADOR

AÑO: 2009

## **AGRADECIMIENTO**

Agradecemos a Dios, nuestros padres, hermanos y amigos por su apoyo en esta importante etapa de nuestras vidas.

## **DEDICATORIA**

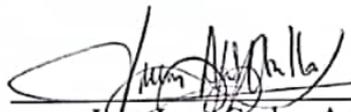
A nuestros padres y hermanos, por su apoyo y confianza incondicional.

# TRIBUNAL DE SUSTENTACION



---

MSc. Patricia Chavez  
PROFESOR DE LA MATERIA



---

Ing. Juan Carlos Aviles  
PROFESOR DELEGADO DEL  
DECANO

## DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(REGLAMENTO DE GRADUACIÓN DE LA ESPOL)

CLEOTILDE NARANJO Ch

Cleotilde Albertina Naranjo Chano

Ma. Fernanda Rivera

María Fernanda Rivera Jara

## RESUMEN

En este proyecto se presenta un algoritmo para analizar imágenes de galletas con el fin de determinar cuáles no cumplen con el estándar, por medio de reconocimiento de formas y morfología matemática.

El primer capítulo da a conocer conceptos respecto al procesamiento digital de las imágenes y las técnicas que se emplearán en el algoritmo.

El segundo capítulo explica mediante que método se realiza el reconocimiento de formas y como se calculan los parámetros que se analizan en el proyecto, además se define valores de los parámetros útiles para determinar si las galletas cumplen o no con el estándar.

En el último capítulo se detalla el desarrollo del proyecto, el proceso del análisis de las imágenes, los obstáculos que surgieron, sus soluciones y límites definidos de forma preventiva. También se desarrolló una interfaz gráfica en la que el usuario indica que imágenes desea analizar, realiza la comparación de formas, detecta si se cumple con el estándar, presenta visualmente diferenciando por colores el tipo de galleta al que corresponde y si cumple o no con el estándar, además genera gráficos por tipo de galletas en los que se observa que los parámetros analizados y resultados obtenidos.

# ÍNDICE GENERAL

RESUMEN.....	I
ÍNDICE GENERAL.....	II
ÍNDICE DE FIGURAS.....	III
ÍNDICE DE TABLAS.....	IV
INTRODUCCIÓN.....	V
1. GENERALIDADES.....	3
1.1 Reconocimiento de formas.....	2
1.1.1 Método geométrico.....	2
1.1.2 Método Estructural.....	3
1.2 Morfología matemática.....	3
2 ESPECIFICACIONES Y PARÁMETROS.....	5
2.1 Pre procesamiento de la imagen.....	6
2.2 Coeficiente de Correlación.....	6
2.3 Métrica.....	7
2.4 Margen de Tolerancia.....	9
3 DESARROLLO DEL PROYECTO.....	11
3.1 Preparación de las Imágenes.....	11
3.2 Manipulación de las Imágenes.....	14
3.3 Funcionalidad del Programa.....	16
CONCLUSIONES.....	18

RECOMENDACIONES.....	19
ANEXOS	
ANEXO A: Manual del usuario.....	20
ANEXO B: Código del programa.....	28
BIBLIOGRAFÍA.....	36

## ÍNDICE DE FIGURAS

Figura 3.1: Flujo de desarrollo del proyecto.....	12
Figura 3.2: Adquisición de imágenes.....	13
Figura 3.3 Basegalletas.....	14
Figura 3.4 Función recortar.....	15
Figura 3.5 Comparacion de resultados.....	17
Figura A-1.- Menu principal del programa.....	20
Figura A-2.- Configuracion de rango de imágenes a ser analizadas.....	21
Figura A-3.- Botón de análisis.....	21
Figura A-4.- Botón de reseteo de contadores.....	21
Figura A-5.- Botón que muestra resultados en forma gráfica.....	22
Figura A-6.- Gráfico de resultados de galletas triangulares.....	22
Figura A-7.- Gráfico de resultados de galletas redondas.....	22
Figura A-8.- Campo en el que se visualiza la imagen analizada.....	23
Figura A-9.- Campo en el que muestra el número de la figura analizada.....	23
Figura A-10.- Número de galletas analizadas en una iamgen.....	24
Figura A-11.- Número de galletas totales analizadas.....	25

## ÍNDICE DE TABLAS

Tabla 3.1: Resultados obtenidos.....	17
--------------------------------------	----

# INTRODUCCIÓN

En las industrias se realizan diversos controles de calidad para reducir cantidad de productos con defectos, aumentar eficiencia y productividad; a fin de obtener productos de primera calidad y ser más competitivos. En el área de control de una fábrica de galletas, existe personal que manualmente separa las galletas que no cumplan con el estándar, por lo que a menudo se cometen errores y es frecuente que consumidores encuentren galletas con fallas en los empaques.

El avance tecnológico y la automatización industrial se han constituido en herramientas indispensables para prevenir y reducir fallas en la producción, simplificar procesos e incrementar productividad y competitividad. Esto ha contribuido al desarrollo e implementación de controles automatizados, obteniendo así productos de mayor calidad.

En este proyecto se diseña un algoritmo para la detección galletas que no cumplen con el estándar la línea de producción de una fábrica de galletas, que permita identificar que productos difieren del estándar.

El algoritmo ha sido desarrollado en Matlab, mediante el procesamiento digital de imágenes, se realiza el reconocimiento de formas y el análisis de parámetros como: el tamaño de las galletas y métrica.

Su ejecución mediante una interfaz gráfica permite apreciar los resultados de forma visual e interactiva, se adjunta un manual de usuario con las indicaciones respectivas a fin de que cualquier persona pueda hacer uso del programa.

# CAPÍTULO 1

## 1. GENERALIDADES

El tratamiento de las imágenes se divide en tres etapas principales:

- Adquisición de imágenes.- Mediante la conversión analógica-digital, transductores procesan la luz emitida o reflejada por cuerpos, obteniendo la imagen de un objeto. Ejemplo: Escáner o cámara digital.
- Procesamiento de imágenes.- “Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información”.
- Presentación.- Es el método que se emplea para mostrar la imagen luego del procesamiento, este puede ser el monitor, televisor o impresora. Cabe recalcar que siempre hay que tomar en cuenta niveles de percepción humana.

## 1.1. Reconocimiento de formas

Reconocer un objeto es asociarlo o identificarlo con un significado. Un sistema de reconocimiento de formas esta constituido por: representación e interpretación.

La representación consiste en preprocesar la imagen (mediante filtrado o umbralización) y parametrizarla para suprimir información redundante o inútil. Los objetos se pueden representar de dos formas:

- No estructurada, el objeto se representa por un conjunto de sub-objetos sin relación entre ellos.
- Estructurada, el objeto se representa por un conjunto de sub-objetos más un conjunto de relación entre ellos.

La interpretación comprende la comparación entre el objeto y un conjunto de formas predefinidas. Puede realizarse según la representación elegida de los objetos, por el método geométrico o estructural.

### 1.1.1. Método Geométrico

Basado en una representación no estructurada, los objetos son considerados de una pieza, donde los objetos se representan por una matriz y la forma se representa mediante una medida de similitud entre ellos y los objetos a reconocer. Este método es muy útil para reconocer formas simples, cuando se tiene un

número pequeño de formas separables mediante funciones discriminantes no complejas.

### 1.1.2. Método Estructural

Se asume una representación estructurada de los objetos, por lo que el reconocimiento no solo debe basarse en el método de reconocimiento geométrico sino también en analizar la relación entre los objetos.

Este método se lo emplea en los casos que el método geométrico es insuficiente, ya que permite elaborar modelos más complejos de estructuras, y así obtener no solo la forma de cada objeto reconocido sino también una descripción de los sub-objetos de la estructura.

## 1.2. Morfología matemática

La morfología matemática, comenzó a finales de los sesenta, se basa en la geometría y forma, sus principales protagonistas son Matheron y Serra. Es el estudio de las formas de objetos en una imagen, se utiliza para la representación y descripción de formas de regiones de una imagen.

Las transformaciones morfológicas constituyen una fase intermedia en el procesamiento de las imágenes, en la que la imagen es segmentada (eliminando el fondo) a fin de obtener una imagen binaria en la que se puedan apreciar los objetos.

Las operaciones morfológicas simplifican datos de las imágenes y eliminan aspectos irrelevantes mediante el filtrado morfológico. Así como también se puede mantener las principales características de forma de los objetos por medio de: la extracción de bordes o del esqueleto, la envolvente convexa, la reducción y ampliación.

Los objetivos del uso de la morfología matemática son los siguientes: suprimir ruido y simplificar formas en el pre-procesamiento de las imágenes; extraer el esqueleto, detectar objetos, envolvente convexa, ampliación y reducción para destacar la estructura de los objetos; y, obtener una descripción morfológica cualitativa de los objetos de la imagen, tales como: área, perímetro.

Por ejemplo, el conjunto de todos los píxeles negros de una imagen binaria, es una descripción morfológica de la imagen.

La morfología es de suma importancia debido a que las formas de una imagen contienen mucha información, ejemplo de ello es la detección de fallas o defectos en líneas de producción.

# CAPÍTULO 2

## 2. ESPECIFICACIONES Y PARÁMETROS

Como se indicó en la introducción para el control de calidad de las galletas es necesario distinguir el tipo de galleta al que corresponde la imagen en estudio con respecto a las imágenes de los moldes y cuan uniforme geoméricamente es la imagen, además de un margen de tolerancia respecto al tamaño final de la galleta.

El reconocimiento de forma se lo emplea para determinar el modelo de galleta al que corresponde la imagen en estudio, este proceso se realizará en dos partes: pre-procesamiento de la imagen y su interpretación, mediante el coeficiente de correlación.

El análisis morfológico de la galleta será representado por una métrica, que indica cuán perfecta es la geometría de la galleta.

El margen de tolerancia lo definimos con el área máxima y mínima permitida para que la galleta cumpla con el estándar.

### **2.1. Pre procesamiento de la imagen**

En la etapa de adquisición de las imágenes puede haber manchas de grasa y/o migajas de galletas, que son consideradas como información no deseada, por lo que en esta etapa se procede a suprimirlas.

Debido al método empleado en la adquisición de las imágenes, que se explica en el tercer capítulo, aislamos el entorno y obtenemos imágenes sin ruido significativo, por lo que no se considera necesario el uso de filtros.

Mediante el comando `bwareaopen` de Matlab, el cual permite remover de una imagen binaria objetos que sean menor a  $P$  píxeles, se eliminan los objetos irrelevantes para el análisis.

### **2.2. Coeficiente de Correlación**

Coeficiente de correlación describe la intensidad de relación entre dos variables. El valor del coeficiente de correlación puede tomar valores desde menos uno hasta uno, indicando que mientras más cercano a uno sea el valor del coeficiente de correlación, en cualquier dirección, más fuerte será la asociación lineal entre las dos variables. Mientras más cercano a cero sea el coeficiente de correlación indicará que más débil es la asociación entre ambas variables. Si es igual a cero

se concluirá que no existe relación lineal alguna entre ambas variables.

Matlab permite calcular el coeficiente de correlación entre dos matrices o vectores del mismo tamaño, dando como resultado un valor que representa la igualdad entre dos matrices. corr2 realiza el coeficiente de correlación usando la expresión:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

Donde:  $\bar{A} = \text{mean2}(A)$ , and  $\bar{B} = \text{mean2}(B)$ .

En los que A y B representan las matrices a comparar que pueden tener valor numérico o lógico, y el resultado será un escalar doble.

### 2.3. Métrica

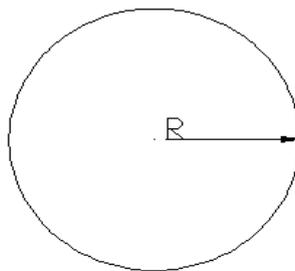
Es un indicador, su valor está comprendido entre cero y uno, mientras más cercano a uno sea su valor, más perfecta geoméricamente será la galleta.

Para realizar el cálculo de la métrica se requiere de la descripción morfológica de las regiones de la imagen. El comando regionprops de Matlab proporciona las medidas de las propiedades de las regiones de la imagen, mediante el obtenemos el área, la cual representa el número de píxeles efectivos en una región.

Perímetro es la distancia alrededor de la frontera de la región. Se lo obtiene por el cálculo de la distancia entre cada par adyacente de píxeles alrededor del borde de la región.

Según la forma de la galleta que se analice, se calcularán las métricas que a continuación se describen; cabe recalcar que la métrica definida para las formas triangulares solo es válida para moldes triangulares equiláteros.

- Formas redondas



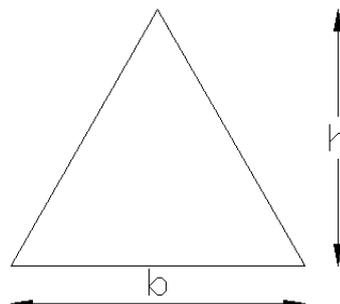
$$area = \pi r^2$$

$$perímetro = 2\pi r$$

Dando como resultado, la siguiente fórmula:

$$métrica = 4\pi * \frac{área}{perímetro^2}$$

- Formas triangulares



$$area = \frac{b * h}{2}$$

$$perímetro = 3b$$

Dando como resultado, la siguiente fórmula:

$$métrica = \frac{36}{\sqrt{3}} * \frac{área}{perímetro^2}$$

#### 2.4. Margen de tolerancia

Debido a que el tamaño de los objetos de nuestro estudio varía, se requirió determinar una media del área de las galletas, así como también tamaño máximo y mínimo permitido para verificar que la galleta cumpla con el estándar.

Como resulta complicado determinar cuánto va a aumentar el tamaño de la galleta al momento de hornearse, se decidió realizar un muestreo de 30 galletas redondas, el análisis se hizo respecto al área, perímetro y métrica.

Los valores que se obtuvieron para las galletas redondas son los siguientes:

$$\mu AR = 9522 \pm 10.85\% \text{ píxeles}$$

$$\mu AR_{min} = \mu AR * (1-0.1085) = 8485 \text{ píxeles}$$

$$\mu AR_{max} = \mu AR * (1+0.1085) = 10558 \text{ píxeles}$$

Respecto a los valores de métricas obtenidos, concluimos que debido a que las galletas no tienen una masa homogénea no son perfectas geométricamente, se decide tomar como valor referencial 0.9. Por lo que se puede decir que si la métrica de las galletas redondas es

mayor a 0.9 se consideran aceptables, caso contrario no cumplen con el estándar.

Como consecuencia de no tener un molde de galletas triangular, no se pudo realizar el muestreo respectivo para obtener valores de área, ni de métrica. Al analizar la figura que se utiliza como molde triangular se puede especificar como media de área, lo siguiente:

$$\mu_{AT} = 4000 \pm 5\% \text{ píxeles}$$

$$\mu_{ATmin} = \mu_{AT} * (1-0.05) = 3800 \text{ píxeles}$$

$$\mu_{ATmax} = \mu_{AT} * (1+0.05) = 4200 \text{ píxeles}$$

El valor de la métrica triangular definido es de 0.95, debido a que su forma será casi perfecta y el valor de las métricas que se obtengan serán cercanas a uno.

Considerando la existencia de figuras de galletas triangulares que no sean equiláteras, en las que el perímetro es mayor que el área, dando como resultado de la fórmula descrita en la sección anterior una cantidad mayor a uno, se ha establecido como métrica límite el valor de uno, el mismo que también se aplicara para las galletas redondas a fin de evitar incongruencias y manejar fórmulas estándar.

# CAPÍTULO 3

## 3. DESARROLLO DEL PROYECTO

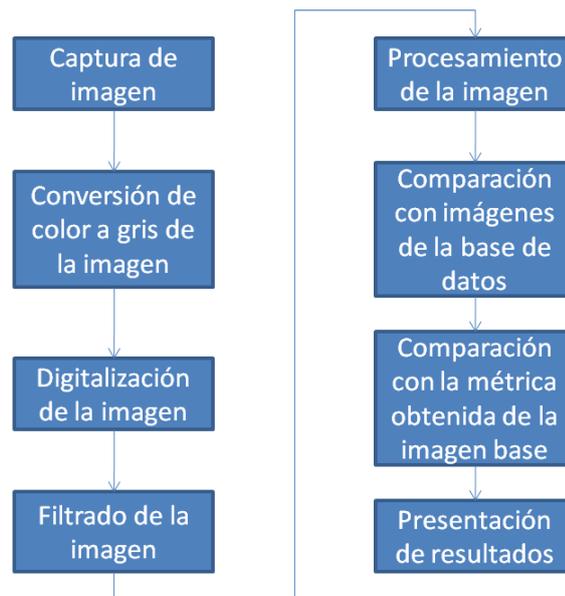
Durante el desarrollo del proyecto nos encontramos con varias dificultades para la adquisición de imágenes idóneas las cuales serian analizadas por el algoritmo desarrollado para la detección de galletas que no cumplen con el estándar, estos inconvenientes fueron resueltos luego de haber realizado varias pruebas en diferentes entornos.

### 3.1. Preparación de las imágenes

Este proyecto se basa en la adquisición y reconocimiento de imágenes de galletas, para este análisis se eligió galletas redondas y triangulares debido a que son más fáciles de comparar que si se utilizara galletas redondas y cuadradas. Las galletas triangulares

están representadas por moldes de cartulina ya que no se pudo contar con un molde triangular para elaborar las galletas.

La evaluación de las galletas con procesamiento digital de imágenes se realizó de la siguiente forma:



**Figura 3.1:** Flujo del desarrollo del proyecto

La primera etapa fue tener galletas muestra para tener un grupo de galletas con defecto y otro grupo de galletas sin defecto y tener un mejor panorama dadas las diferentes características de las fracturas y tomando en cuenta que entre una galleta y otra podría existir un margen de diferencia.

La segunda etapa fue la adquisición de la imagen con una cámara web para tener imágenes digitales, al momento de realizar la captura de las imágenes a ser analizadas se debió tener en cuenta varios detalles para obtener imágenes óptimas.

Los problemas que se nos presentaron fueron la ubicación de la fuente de luz, la intensidad de la fuente, la ubicación de la cámara web, la ubicación de las galletas, y el fondo sobre el que estarían las galletas ya que todos estos factores influyeron en que las imágenes de las galletas tengan sombra y así se tome esta sombra como un defecto en la galleta.

La forma de tomar las imágenes fue colocar la cámara web a 90cm de las galletas que se encontraban sobre una hoja de papel blanco con una iluminación blanca, esto resolvió el problema del fondo de las galletas, la fuente de luz y la ubicación de la cámara.



**Figura 3.2:** Adquisición de imágenes

Durante la captura de las imágenes tuvimos que probar varias ubicaciones de la fuente de luz para evitar que se produzca sombra o la luz se descomponga en sus diferentes colores en el papel blanco, la solución fue iluminar en posición vertical pero con un pequeño

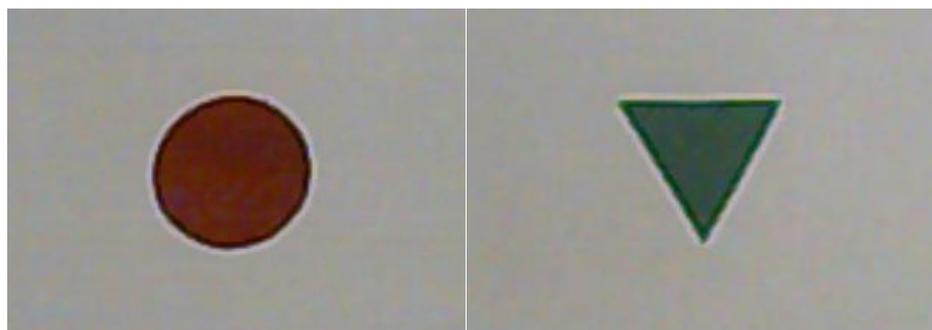
ángulo de inclinación para evitar iluminar la cámara y que no genere su sombra en el fondo blanco.

### 3.2. Manipulación de las imágenes

El desarrollo del algoritmo que se utilizó para la detección de fallas en las galletas dentro de una línea de producción, se lo hizo en MATLAB, herramienta que nos permitió el análisis de las imágenes obtenidas utilizando la conversión de imágenes a matrices, la representación de datos y funciones, para finalmente mostrar los resultados a través de una interfaz gráfica diseñada en MATLAB.

El proceso que se realizó para analizar las imágenes en MATLAB empieza con el cambio de color a escala de grises de la imagen para poder guardarla en un arreglo *figuras*, desde donde se la llamará para realizar el respectivo análisis.

Se tiene además el arreglo *basegalletas* en el que constan: la figura de los moldes de la galleta redonda y triangular. Estas figuras son nuestras plantillas con las que compararemos las galletas que vamos a verificar.

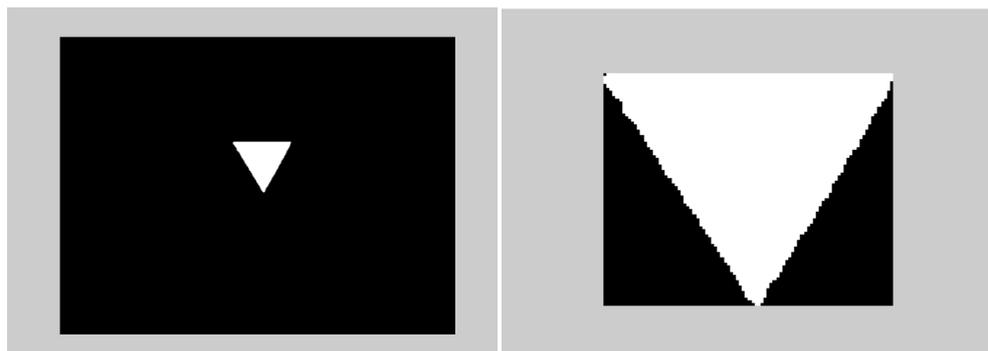


**Figura 3.3:** Basegalletas.

Las imágenes que tenemos en el arreglo *figuras* se las toma una a una, luego se llama a la función de preparar imagen (*prepareim*) que pasa a la misma por un filtro que elimina las partes irrelevantes para el análisis como las migajas de las galletas y las manchas de grasa que se pudieran encontrar sobre nuestra base de papel blanco.

Luego se obtiene las propiedades de la figura que se está analizando, estas propiedades son el área y perímetro de la galleta, como se explicó en el capítulo 2 sección 3.

Siguiendo el algoritmo, se llama a la función *cortar*, que toma la imagen y la busca sus valores máximos y mínimos para limitarla y proceder a eliminar la información que se encuentra fuera de estos límites, de tal forma que entrega una figura en la que se observa solo la galleta, a esta figura se la redimensiona por medio del comando *imresize*, para obtener una imagen de 42x42.



**Figura 3.4:** Función recortar.

Con la imagen delimitada procedemos a llamar a la función *comparar*, en la que se utilizan las figuras del arreglo *basegalletas* y la figura de

la imagen que se encuentra en análisis para calcular el coeficiente de correlación, determinando así el tipo de galleta al que corresponde.

Según el tipo de galleta, se calcula la métrica de la figura, y se compara con los parámetros definidos en el capítulo 2. Luego de realizar la comparación, dependiendo si la figura de la galleta en análisis está dentro de los parámetros especificados, se procede a presentar gráficamente: el contorno de la galleta (los colores mostrados en la interfaz grafica se detallan en el anexo A), se actualiza el valor del contador (se detallan los contadores en el anexo A), y se almacenan estos datos en los arreglos *ResultadoTRI* (*galletastriangulares*) o *ResultadoRED* (*galletas redondas*).

En los arreglos de resultado se introducen los valores de la correlación, el área, perímetro, métrica y test de la figura que está siendo analizada; test toma el valor de 0 cuando una galleta no pasa la prueba y 1 cuando una galleta pasa la prueba.

Los valores de los contadores se muestran gráficamente cuando se está analizando cada imagen y finalmente una vez que se analizan todas las imágenes deseadas muestra un resultado total que también puede ser apreciado por el usuario.

### **3.3. Funcionalidad del Programa**

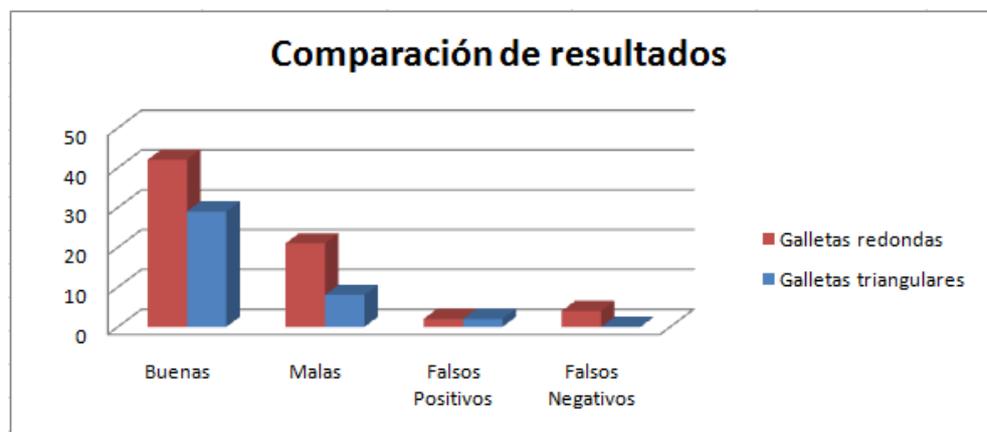
Para demostrar la eficiencia del algoritmo obtenido para la detección de galletas que no cumplen con el estándar, se realizó un análisis de falsos positivos y negativos.

En el análisis se considera como: “Falsos positivos” cuando un observador indica que la galleta en análisis cumple con el estándar, pero el algoritmo muestra lo contrario; y viceversa para “Falsos negativos”.

A continuación se presentan los resultados obtenidos y su gráfico correspondiente:

Analisis / Resultados	Galletas redondas	Galletas triangulares	Total
Buenas	42	29	71
Malas	21	8	29
Falsos Positivos	2	2	4
Falsos Negativos	4	0	4

**Tabla 1.1:** Resultados obtenidos.



**Figura 3.5:** Comparación de resultados.

## CONCLUSIONES

1. El algoritmo diseñado en este proyecto realiza de manera eficiente el reconocimiento de formas de las galletas, ya que no hubieron casos en los que indique que una galleta redonda es triangular o viceversa.
2. Los datos de las características cualitativas de las galletas obtenidos en la ejecución del algoritmo, son almacenados en registros y presentados gráficamente para que puedan ser analizados y contribuyan a tener un mejor control de los productos.
3. Al analizar la eficiencia del algoritmo se aprecia un 4% de falsos positivos, que puede ser minimizado con: la elaboración de las galletas triangulares y la obtención de valores reales de área y métrica; las galletas que son retiradas del proceso de producción se las puede emplear como materia prima para otros productos.
4. Se observa un 4% de falsos negativos correspondientes a las galletas redondas, esto se produce cuando la galleta tiene ligeramente achatada una parte del contorno y al realizar el análisis morfológico el valor de la métrica es mayor al definido en el estándar.
5. A futuro proyecto se deja la obtención de imágenes desde la banda transportadora de galletas mediante adquisición de video. Así como también la automatización de retirar las galletas que no cumplen con el estándar.

## RECOMENDACIONES

1. Aislar el entorno para la adquisición de las imágenes, de tal manera que se pueda evitar que en el fondo blanco se observe la descomposición de la luz blanca.
2. Agregar un filtro de acuerdo a características de la empresa en la que se vaya a implementar el algoritmo.
3. Realizar la adquisición de las imágenes a la misma altura para que la comparación concuerde con los parámetros especificados.

## ANEXO A

### MANUAL DEL USUARIO

El programa desarrollado esta creado para clasificar a las galletas analizadas por tipo y detectar las fallas que existen en las galletas.

Este análisis se lo realiza a un grupo de imágenes determinado por el usuario y que previamente se encuentran almacenados en una matriz.

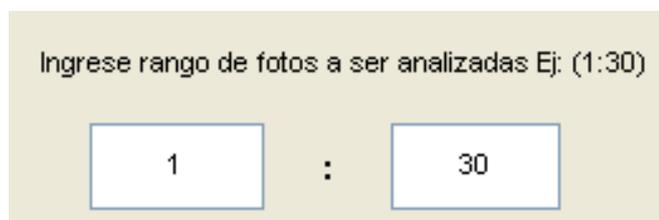


Figura A-1.- Menu principal del programa

#### Parámetro configurable:

**Rango de las imágenes:** Este campo determina las imágenes que serán analizadas por el algoritmo desarrollado. Por el momento este parámetro

puede tomar valores de 1 a 30 ya que en nuestra matriz tenemos 30 imágenes en escala de grises.



Ingrese rango de fotos a ser analizadas Ej: (1:30)

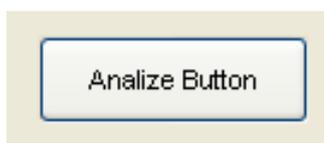
1 : 30

The image shows a light beige rectangular box containing the text 'Ingrese rango de fotos a ser analizadas Ej: (1:30)'. Below this text are two white rectangular input fields with blue borders. The first field contains the number '1', followed by a colon ':', and the second field contains the number '30'.

**Figura A-2.-** Configuración del rango de las imágenes a ser analizadas

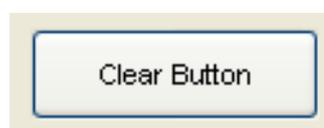
#### **Funcionalidad de los botones:**

**Analyze Button:** Con este botón se da inicio al análisis de las imágenes, el número de galletas que se analizarán va a depender de los valores ingresados en el rango.



**Figura A-3.-** Botón de análisis

**Clear Button:** Pone en cero los contadores parciales y totales para poder realizar un nuevo análisis y cambiar el rango.



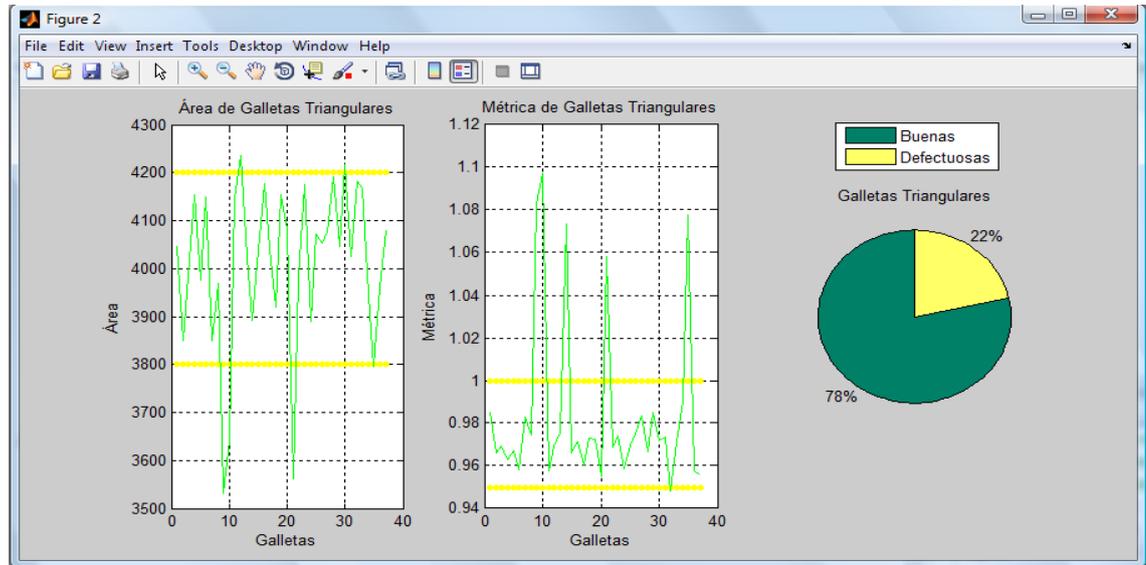
**Figura A-4.-** Botón de reseteo de contadores

**Graphic Button:** muestra gráficamente los resultados obtenidos del análisis, como el número de galletas con fallas y sin fallas, el área y la métrica de las mismas, ambos resultados se clasifican dependiendo del tipo de galleta.

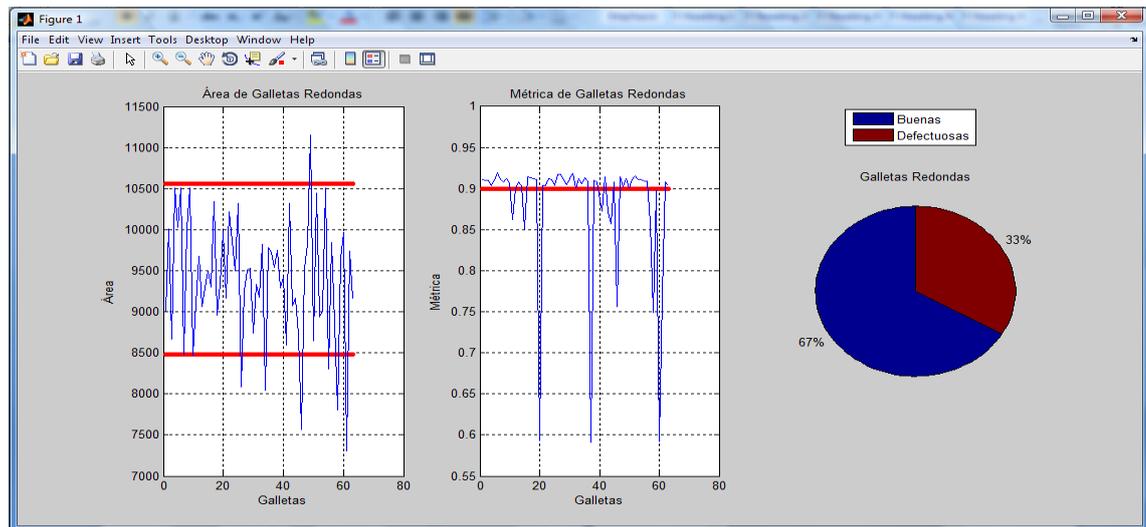


Graphic Button

**Figura A-5.-** Boton que muestra resultados de forma grafica



**Figura A-6.-** Gráfico de resultados de las galletas triangulares

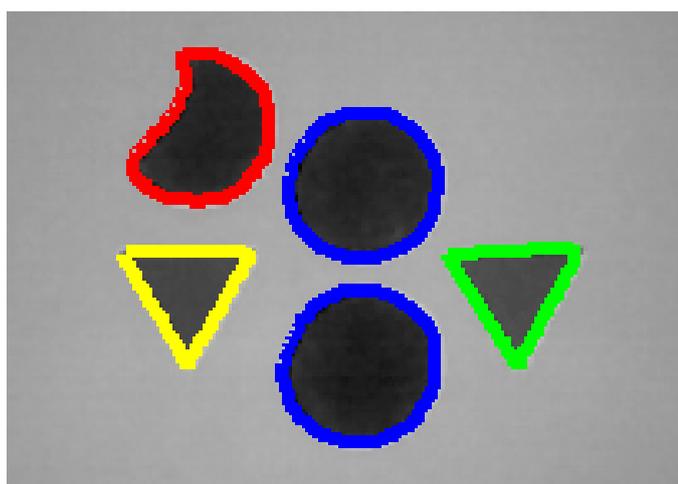


**Figura A-7.-** Gráfico de resultados de las galletas redondas

## Campos de resultados

**Campo visual:** Presenta la imagen que en ese momento el programa está analizando, muestra además bordes de colores que dependerán del tipo de galleta analizada y el estado de la misma, el código de colores utilizado en nuestro algoritmo es:

- **Azul:** Galleta redonda sin fallas
- **Rojo:** Galleta redonda con fallas
- **Verde:** Galleta triangular sin fallas
- **Amarillo:** Galleta triangular con fallas



**Figura A-8.-** Campo en el que se visualiza la imagen analizada

**Campo de imagen No:** Muestra el número de la imagen que en ese instante está siendo mostrada en el campo visual.

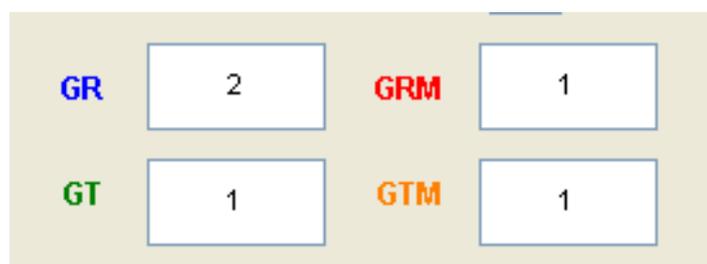


**Figura A-9.-** Campo en el que muestra el número de la figura analizada

### Campo de resultados parciales

Durante el análisis el programa ira analizando imagen por imagen y mostrando los resultados parciales, es decir, los resultados de la foto que en ese instante se está analizando. Para mostrar estos resultados parciales contamos con cuatro campos que son GR, GRM, GT y GTM, dichos campos significan:

- **GR:** Numero de galletas redondas en la imagen en buen estado
- **GRM:** Numero de galletas redondas en la imagen en mal estado
- **GT:** Numero de galletas triangulares en la imagen en buen estado
- **GTM:** Numero de galletas triangulares en la imagen en mal estado



**Figura A-10.-** Número de galletas analizadas en una iamgen

### Campo de resultados totales

Durante el análisis se muestra resultados en este campo a medida que se va analizando, este contador se va acumulando y luego de analizar las galletas que se encuentran dentro del rango determinado por el usuario, el programa muestra los resultados totales, para este efecto utiliza los campos GRT, GRMT, GTT y GTMT, dichos campos significan:

- **GRT:** Numero de galletas redondas totales en buen estado

- **GRMT:** Numero de galletas redondas totales en mal estado
- **GTT:** Numero de galletas triangulares totales en buen estado
- **GTMT:** Numero de galletas triangulares totales en mal estado

Resultados total			
<b>GRT</b>	42	<b>GRMT</b>	21
<b>GTT</b>	29	<b>GTMT</b>	8

**Figura A-11.-** Número total de galletas analizadas

## ANEXO B

### CÓDIGO DEL PROGRAMA

#### dsp2.m

```

function varargout = DSP2(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @DSP2_OpeningFcn, ...
                  'gui_OutputFcn',  @DSP2_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before DSP2 is made visible.
function DSP2_OpeningFcn(hObject, eventdata, handles,
varargin)
movegui(hObject,'center')
handles.output = hObject;
load figuras; %matriz que contiene las imagenes a ser
analizadas
load basegalletas; %matriz q contiene imagenes base para
comparacion de forma
background = imread('galletas\background.jpg'); %Leer imagen
%axes(handles.background); %Carga la imagen en background
%colormap('bone');
axis off;
imshow(background); %Presenta la imagen
axis equal;
axis tight;
axis off;
handles.output = hObject;
guidata(hObject, handles);

function varargout = DSP2_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

```

```

function edit1_Callback(hObject, eventdata, handles)
e1=get(hObject,'String');%Almacenar valor ingresado
edit1=str2double(e1);%Transformar a formato double
handles.edit1=edit1;
guidata(hObject, handles);

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
e2=get(hObject,'String');%Almacenar valor ingresado
edit2=str2double(e2);%Transformar a formato double
handles.edit2=edit2;
guidata(hObject, handles);

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%ANALIZAR IMAGENES
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
GR=0; GRM=0; GRT=0;GRMT=0; %contadores de galletas Redondas
GT=0; GTM=0; GTT=0;GTMT=0; %contadores de galletas
Triangulares
CGR=1; CGT=1;
resultadoRED=[]; %arreglo que almacenara los parametros de
estudio
resultadoTRI=[];
j=1;
m1=handles.edit1; %desde que imagen se realizara el analisis
m2=handles.edit2; % hasta que imagen se realizara el analisis
%-----
%-----PARAMETROS A COMPARAR (estos tambien deben estar
especificados en push_button1 MOSTRAR RESULTADOS)-----
%-----
uAR=9522; %media de area de galleta redonda
uARmin=uAR*(1-0.1089); %area minima permitida de las galletas
segun muestreo
uARmax=uAR*(1+0.1089); %area maxima permitida de las galletas
segun muestreo
metricaRED=0.9;
%-----
uAT=4000; %media de area de galleta triangular
uATmin=uAT*(1-0.05); %area triangular minima
uATmax=uAT*(1+0.05); %area triangular maxima
metricaTRI=0.95;
metricaLIM=1;
%%-----
%%-----

```

```

load figuras; %matriz de imagenes a analizar
for m=m1:1:m2
    set(handles.edit8,'String',m);
    im=figuras{1,m};
    imshow(im);
    axis equal;
    axis tight;
    axis off;
    imprueba=prepareim(im);
    guidata(hObject, handles);
    [B L]=bwboundaries(imprueba);
    for k=1:length(B)
        stats=regionprops(L, 'Area');
        area=stats(k).Area;
        boundary=B{k};
        delta_sq = diff(boundary).^2;
        perimeter = sum(sqrt(sum(delta_sq,2)));
        impruebal=cortar(L==k);%cortar imagen
        impruebal=imresize(impruebal,[42 42]);
        [forma tipo]=comparar(impruebal);
        if tipo==1
            metrica=4*pi*area/(perimeter)^2;
            if metrica>metricaRED & (metrica<metricaLIM |
metrica==metricaLIM) & (area>uARmin |area==uARmin) &
(area<uARmax | area==uARmax)
                hold on
                plot(boundary(:,2), boundary(:,1), 'b',
'LineWidth', 3)
                GR=GR+1;%GalletaRedonda que pasan la prueba
                resultadoRED(CGR,[1 2 3 4 5 6])=[CGR area
perimeter metrica tipo 1];
                CGR=CGR+1;
            else
                hold on
                plot(boundary(:,2), boundary(:,1), 'r',
'LineWidth', 3)
                GRM=GRM+1;%GalletaRedonda que no pasan la
prueba
                resultadoRED(CGR,[1 2 3 4 5 6])=[CGR area
perimeter metrica tipo 0];
                CGR=CGR+1;
            end
        elseif tipo==2
            metrica=((36/sqrt(3))*area)/(perimeter^2);
            if metrica>metricaTRI & (metrica<metricaLIM |
metrica ==metricaLIM) & (area>uATmin |area==uATmin) &
(area<uATmax | area==uATmax)
                hold on
                plot(boundary(:,2), boundary(:,1), 'g',
'LineWidth', 3)
                GT=GT+1;%GalletaTriangular que pasan la prueba
                resultadoTRI(CGT,[1 2 3 4 5 6])=[CGT area
perimeter metrica tipo 1];
                CGT=CGT+1;
            else
                hold on

```

```

        plot(boundary(:,2), boundary(:,1), 'y',
'LineWidth', 3)
        GTM=GTM+1;%GalletaTriangular que no pasan la
prueba
        resultadoTRI(CGT,[1 2 3 4 5 6])=[CGT area
perimeter metrica tipo 0];
        CGT=CGT+1;
        end
    end
    set(handles.edit3,'String',GR);%show values in each
image
    set(handles.edit4,'String',GRM);
    set(handles.edit11,'String',GT);
    set(handles.edit12,'String',GTM);
    pause(0.005)%ESTE VALOR PUEDE SER CAMBIADO, DA UNA
PAUSA PARA PODER OBSERVAR EL ANALISIS DE LA IMAGEN
    end
    GRT=GRT+GR;%refresh total values
    GRMT=GRMT+GRM;
    GTT=GTT+GT;
    GTMT=GTMT+GTM;
    set(handles.edit5,'String',GRT);%show total values
    set(handles.edit6,'String',GRMT);
    set(handles.edit13,'String',GTT);
    set(handles.edit14,'String',GTMT);
    GR=0;GRM=0;GT=0;GTM=0;%enceramos para saber cuantas
galletas hay en una imagen
    handles.GRT=GRT; handles.GRMT=GRMT;
    handles.GTT=GTT; handles.GTMT=GTMT;
end
save('resultadoRED','resultadoRED');
save('resultadoTRI','resultadoTRI');
guidata(hObject, handles);

%--MOSTRAR RESULTADOS
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
load resultadoRED;load resultadoTRI;
[z1 z2]=size(resultadoRED);
[z3 z4]=size(resultadoTRI);
xr=1:z1; xt=1:z3;
%-----
%-----PARAMETROS A COMPARAR (estos tambien deben estar
especificados en push_button1 MOSTRAR RESULTADOS)-----
%-----
uAR=9522; %media de area de galleta redonda
uARmin=uAR*(1-0.1089);
uARmax=uAR*(1+0.1089);
metricaRED=0.9;
%-----
uAT=4000; %media de area de galleta triangular
uATmin=uAT*(1-0.05); %area triangular minima
uATmax=uAT*(1+0.05); %area triangular maxima
metricaTRI=0.95;
metricaLIM=1;
%%-----

```

```

%%-----
GRT=handles.GRT;GRMT=handles.GRMT;
GTT=handles.GTT;GTMT=handles.GTMT;
REDONDAS=[GRT GRMT]; TRIANGULARES=[GTT GTMT];
%-----plot area de galletas Redondas
figure,subplot(1,3,1); plot(xr,uARmin,'-or',xr,uARmax,'-
or','MarkerSize',1.5,'MarkerEdgeColor','r','MarkerFaceColor','
red')
hold on;
subplot(1,3,1); plot (resultadoRED(1:z1,2:2),'-b');
figure(gcf)
xlabel('Galletas'); ylabel('Área'); title('Área de Galletas
Redondas'); grid on;
%-----plot metrica de galletas Redondas
subplot(1,3,2); plot (xr,metricaRED,'-or',xr,metricaLIM,'-
or','MarkerSize',1.5,'MarkerEdgeColor','r','MarkerFaceColor','
red');
hold on;
subplot(1,3,2); plot (resultadoRED(1:z1,4:4),'-b');
figure(gcf)
xlabel('Galletas'); ylabel('Métrica'); title('Métrica de
Galletas Redondas'); grid on;
%-----plot % buenas y defectuosas
subplot(1,3,3), pie(REDONDAS);title('Galletas
Redondas');colormap jet;
legend('Buenas','Defectuosas');

%-----plot area de galletas Triangulares
figure, subplot(1,3,1); plot(xt,uATmin,'-og',xt,uATmax,'-
og','MarkerSize',1.5,'MarkerEdgeColor','y','MarkerFaceColor','
yellow');
hold on;
subplot(1,3,1); plot (resultadoTRI(1:z3,2:2),'-g');
figure(gcf)
xlabel('Galletas'); ylabel('Área'); title('Área de Galletas
Triangulares'); grid on;
%-----plot metrica de galletas Triangulares
subplot(1,3,2); plot (xt,metricaTRI,'-og',xt,metricaLIM,'-
og','MarkerSize',1.5,'MarkerEdgeColor','y','MarkerFaceColor','
yellow');
hold on;
subplot(1,3,2); plot (resultadoTRI(1:z3,4:4),'-g');
figure(gcf)
xlabel('Galletas'); ylabel('Métrica'); title('Métrica de
Galletas Triangulares'); grid on;
%-----plot buenas y defectuosas
subplot(1,3,3), pie(TRIANGULARES);title('Galletas
Triangulares');colormap summer;
legend('Buenas','Defectuosas');

guidata(hObject, handles);

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
function edit8_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%-----INICIALIZAR TODO-----
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.edit3,'String','');
set(handles.edit4,'String','');
set(handles.edit5,'String','');
set(handles.edit6,'String','');
set(handles.edit8,'String','');
set(handles.edit11,'String','');
set(handles.edit12,'String','');
set(handles.edit13,'String','');
set(handles.edit14,'String','');
resultadoRED=[];resultadoTRI=[]; %inicializar arreglos
background = imread('galletas\background.jpg');
imshow(background);
axis equal;
axis tight;

```

```
axis off;
handles.output = hObject;
guidata(hObject, handles);
```

```
function edit9_Callback(hObject, eventdata, handles)
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit10_Callback(hObject, eventdata, handles)
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit11_Callback(hObject, eventdata, handles)
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit12_Callback(hObject, eventdata, handles)
function edit12_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit13_Callback(hObject, eventdata, handles)
function edit13_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
function edit14_Callback(hObject, eventdata, handles)
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

**plantillas.m**

```

% Script para crear las base de datos
% Leer imágenes de moldes de galletas
GRED=imread('galletas\redonda.jpg');
GTRI=imread('galletas\pic0139.jpg');

%convertir imagenes a escala de grises
GRED=rgb2gray(GRED);
GTRI=rgb2gray(GTRI);

% Preparar imágenes
GRED=prepareim(GRED);
GTRI=prepareim(GTRI);

% Cortar imágenes
GRED=cortar(GRED);
GTRI=cortar(GTRI);

% Redimensionar al tamaño de 42x42
GRED=imresize(GRED,[42 42]);
GTRI=imresize(GTRI,[42 42]);

% Almacenar en la base de datos
letter=[GRED GTRI];
character=letter;
basegalletas=mat2cell(character,42,[42 42]);
save ('basegalletas','basegalletas');

```

**figurasgalletas.m**

```

%Crear matriz que contenga todas las imagenes de las galletas
a analizar
im1=rgb2gray(imread('galletas\pic0163.jpg'));
im2=rgb2gray(imread('galletas\pic0147.jpg'));
im3=rgb2gray(imread('galletas\pic0148.jpg'));
im4=rgb2gray(imread('galletas\pic0149.jpg'));
im5=rgb2gray(imread('galletas\pic0150.jpg'));
im6=rgb2gray(imread('galletas\pic0151.jpg'));
im7=rgb2gray(imread('galletas\pic0164.jpg'));
im8=rgb2gray(imread('galletas\pic0048.jpg'));
im9=rgb2gray(imread('galletas\pic0143.jpg'));
im10=rgb2gray(imread('galletas\pic0131.jpg'));
im11=rgb2gray(imread('galletas\pic0152.jpg'));
im12=rgb2gray(imread('galletas\pic0102.jpg'));
im13=rgb2gray(imread('galletas\pic0153.jpg'));
im14=rgb2gray(imread('galletas\pic0154.jpg'));
im15=rgb2gray(imread('galletas\pic0155.jpg'));
im16=rgb2gray(imread('galletas\pic0156.jpg'));
im17=rgb2gray(imread('galletas\pic0157.jpg'));
im18=rgb2gray(imread('galletas\pic0158.jpg'));
im19=rgb2gray(imread('galletas\pic0159.jpg'));
im20=rgb2gray(imread('galletas\pic0160.jpg'));
im21=rgb2gray(imread('galletas\pic0139.jpg'));
im22=rgb2gray(imread('galletas\pic0161.jpg'));

```

```

im23=rgb2gray(imread('galletas\pic0162.jpg'));
im24=rgb2gray(imread('galletas\pic0163.jpg'));
im25=rgb2gray(imread('galletas\pic0116.jpg'));
im26=rgb2gray(imread('galletas\pic0140.jpg'));
im27=rgb2gray(imread('galletas\pic0134.jpg'));
im28=rgb2gray(imread('galletas\pic0128.jpg'));
im29=rgb2gray(imread('galletas\pic0165.jpg'));
im30=rgb2gray(imread('galletas\pic0146.jpg'));

galletas=[im1 im2 im3 im4 im5 im6 im7 im8 im9 im10 im11 im12
im13 im14 im15 im16 im17 im18 im19 im20 im21 im22 im23 im24
im25 im26 im27 im28 im29 im30];
figuras=mat2cell(galletas, 480, [640 640 640 640 640 640 640
640 640 640 640 640 640 640 640 640 640 640 640 640 640
640 640 640 640 640 640 640]);
save('figuras','figuras');

```

### prepareim.m

```

%Funcion para preparar imagen para analisis
function figura=prepareim(imagen)
imagen = medfilt2(imagen);
if ~islogical(imagen)
    imagen=im2bw(imagen,graythresh(imagen));
end
[f c]=size(imagen);
imagen (1,1)=1;
imagen (f,1)=1;
imagen (1,c)=1;
imagen (1,c-1)=1;
imagen (2,c)=1;
imagen (3,c)=1;
imagen (f,c)=1;
imagen=bwareaopen(~imagen,320);
figura=imagen;

```

### cortar.m

```

%Funcion para cortar imagen
function figura=cortar(imagen)
a=imagen;
[f c]=find(a);
lmaxc=max(c);lminc=min(c);
lmaxf=max(f);lminf=min(f);
figura=a(lminf:lmaxf,lminc:lmaxc);

```

comparar.m

```
function [forma tipo]=comparar(imagen)
% Cálculo de la correlación entre dos imágenes de 42*42 px
comp=[];
load basegalletas.mat
for n=1:2 %2->Número de formas en la base de datos
    sem=corr2(basegalletas{1,n},imagen);
    comp=[comp sem];
end

comp=abs(comp);

tipo=find(comp==max(comp));

if     tipo==1
    forma='REDONDA';
elseif tipo==2
    forma='TRIANGULAR';
end
```

## BIBLIOGRAFIA

1. González Rafael, y Woods Richard, Tratamiento Digital de Imágenes, (USA, Co-publicación de Addison-Wesley Iberoamericana, S.A. y Ediciones Díaz de Santos, S.A., 1996).
2. Hernández Tejera Francisco, y Lorenzo Navarro José, Reconocimiento de Formas: Clasificación y Aprendizaje, (España, Universidad de Las Palmas de Gran Canaria, Departamento de Informática y Sistemas, Monografía, 2002).
3. Serra Jean, Image Analysis and Mathematical Morphology, Academic Press, 1982.
4. Díaz-de-León Santiago, J.L. & Yáñez Márquez, C. (2003). "Introducción a la morfología matemática de conjuntos", Colección de Ciencia de la Computación, CIC-IPN-UNAM-FCE, México. ISBN: 970-36-0075-1. J.
5. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall International, 1989.
6. The Mathworks Inc., Regionprops, <http://www.mathworks.com/access/helpdesk/help/toolbox/images/regionprops.html>, 2009