

## REPRESENTACIÓN DEL CONOCIMIENTO EN UN SISTEMA DIFUSO PARA LA TOMA DE DECISIONES EN LA PRODUCCIÓN DE BALINES

Bustillo Díaz Mario Mauricio, García García Alma Delia, Hernández Moyotl Veronica Judith, Martínez Guzmán Gerardo, Rangel Huerta Alejandro, Ata Pérez Apolonio.

Benemérita Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación

Av. San Claudio y Blvd. 14 sur, col. San Manuel 72570. H. Puebla de Zaragoza, Pue.

Tel +52(222) 229 5500 Ext. 7202

bustillo@cs.buap.mx, deliakzy@gmail.com, judith\_h21@hotmail.com, gguzman@cs.buap.mx, arangel@cs.buap.mx, apolonio@cs.buap.mx

### RESUMEN

Los sistemas automáticos de inspección visual son aplicados para determinar la calidad de productos en diferentes áreas, en especial al área industrial. Estos sistemas han tenido grandes avances debido al progreso de las técnicas de procesamiento digital de imágenes. Este proyecto propone desarrollar un prototipo para el proceso de detección de defectos en la superficie de balines. Este prototipo está basado en técnicas de procesamiento digital y lógica difusa, que realice la actividad de inspección. El objetivo de este trabajo es que el sistema verifique la calidad del producto (balines) en la fábrica de Tzubaki de CA [1]. El resultado que proporcione el sistema permitirá tomar la decisión acerca del estado del balín bueno o defectuoso.

**Palabras Clave:** Balín, Inspección, Lógica difusa, Técnicas de procesamiento digital de imágenes, Calidad.

### ABSTRACT

Automatic inspection visual systems have been used to determine products quality in different areas. Development systems are due to the new progress of the digital image processing techniques. The main goal of this project is to develop a prototype for detection process defects surface of metal pellets. This prototype is based on digital processing techniques and fuzzy logic necessary to perform the inspection activity. This work is used to the system checks the quality of the product (pellets) in factory Tzubaki AC [1]. Results provided by the system will make the contribution to decision about the status of goodness or defective pellets.

**Keywords:** Pellet, Inspection, fuzzy logic, techniques digital image processing, quality.

### 1. INTRODUCCION

En la actualidad los sistemas visuales de inspección automática son aplicados en diferentes áreas para revisar la calidad de los productos, en especial al área industrial. Estos sistemas han tenido grandes avances debido al progreso de las técnicas del procesamiento digital de imágenes.

En este proyecto se propone implementar un sistema de inspección de control de calidad para la detección de defectos en la superficie de balines. Esta tarea, actualmente es supervisada por expertos en el área, donde como instrumento principal utilizan el sentido de la vista.

Para el diseño del sistema es necesario el conocimiento del experto, ya que permitirá un mayor conocimiento más de esa actividad. El experto es el operador encargado del área de inspección que determina la clasificación y caracterización de

los defectos que afectan los estándares de calidad que debe cumplir el producto.

El objetivo de este trabajo es desarrollar un prototipo que pueda implementarse para el mejoramiento del control de calidad del producto (balines). La ejecución de la inspección es realizada utilizando la vista. Las consecuencias de la ejecución de esta tarea es **posible** que se den problemas de salud, ya que es **posible** el cansancio visual y otras enfermedades afines.

Este sistema se implementa en base a las técnicas de procesamiento digital de imágenes y de lógica difusa. La lógica difusa se caracteriza por: (1) se puede modelar e implementar el conocimiento, y se expresa a través de un conjunto de reglas lingüísticas del tipo “si antecedente entonces consecuente” en lugar de sofisticadas ecuaciones diferenciales [2]; (2) es comprensible aún y cuando no se tengan antecedentes del proceso a controlar y (3) tiene la capacidad de realizar inferencias a partir de una información incompleta dado que maneja ambigüedad, imprecisión e incertidumbre [3].

Un sistema difuso se puede ver como un mapeo no lineal de una o más entradas y salidas. Como se visualiza en la figura 1.

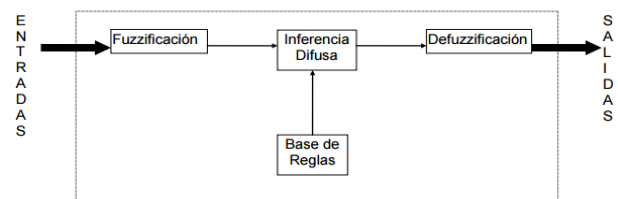


Figura 1. Mapeo del sistema difuso

Es importante comprender, que las entradas y salidas son números concretos y no conjuntos abstractos que resultan de la inferencia difusa. Internamente, nuestro sistema tipifica la entrada de acuerdo a su “membrecía”, con uno o más de los conjuntos de entrada. Este proceso se llama “fuzzificación” y su actividad es observar la entrada buscando si determinada característica de la misma coincide con la de algún conjunto de entrada y en qué medida.

Las reglas son operaciones sencillas de inferencia lógica con las que se describe el comportamiento del sistema. Estas reglas tienen una estructura tipo si-entonces. Si (se cumplen determinadas causas) entonces (ocurre tal consecuencia) [4].

Es manera que los sistemas de lógica difusa, pueden incorporar conocimientos de tipo cualitativo/subjetivo (además de los tradicionales). Estas reglas incorporan el conocimiento del experto al sistema en su forma nativa. Cada regla evaluada da por resultado un subconjunto difuso de su consecuente, dado que cada regla está vinculada con las restantes mediante una relación inclusiva. El conjunto difuso representante de la inferencia es la composición inclusiva de todos los subconjuntos difusos obtenidos como resultado inferencial en cada regla.

El proceso de defusificación se encarga de obtener un único valor numérico representativo del resultado de la inferencia.

Los problemas que se desean solucionar con este sistema son los siguientes: la determinación de los píxeles relacionados con los defectos del balón que se basa en una imagen digital y la toma de decisión acerca del estado del balón (bueno ó defectuoso).

Este artículo está organizado en secciones de la siguiente manera: en la Sección 2, el desarrollo del sistema. La Sección 3, el método de solución, en la sección 4, pruebas y resultados experimentales, en la Sección 5, se mostrará un análisis de eficiencia del algoritmo y finalmente las conclusiones y referencias bibliográficas.

## 2. DESARROLLO DEL SISTEMA

El principio de operación del sistema está basado en imágenes previamente capturadas bajo ciertas condiciones como: Luz, distancia y movimiento. Se sabe que una imagen está representada por una matriz de  $n \times m$  píxeles. Para poder realizar una clasificación y determinar que píxeles corresponden al defecto es necesaria la imagen en escala de grises, debido a que esto permite que sea más fácil clasificar en regiones la superficie del balón. En la figura 2 se muestran algunas imágenes de balines que fueron utilizadas para las pruebas.

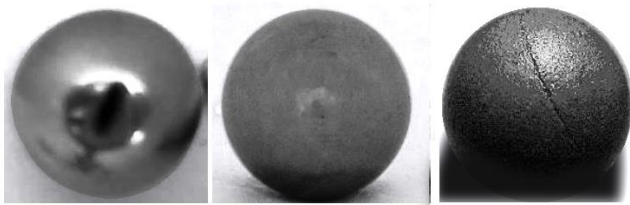


Figura 2. Imágenes de balines

Tomando en cuenta la información que se obtiene de la imagen se realiza un análisis cuantitativo y cualitativo con respecto a los valores de los píxeles permitiendo así, la obtención de 7 conjuntos.

Durante el análisis también se realizaron algunos cálculos [5] para caracterizar **que los conjuntos difusos fueran deseables para la representación** del conocimiento.

Los cálculos aplicados fueron:

*Especificidad:* Permitió medir la dificultad para elegir el punto representante de un conjunto, debido a que si existe mayor

especificidad en los conjuntos, la dificultad de elegir uno de estos puntos será menor.

*Posibilidad:* Permitió medir cuando dos conjuntos se superponen, es decir que dado un evento se determina si es posible **o no en algún conjunto**, basando la operación en la ecuación 2.1.

$$Poss(A, B) = Sup_{x \in X} \{ \min(A(x), B(x)) \} \quad (2.1)$$

*Necesidad:* Permitió medir con qué grado cierto conjunto está incluido en otro, es decir, que dado un evento se determina si es necesario o no que exista en algún conjunto, basando la operación en la ecuación siguiente:

$$Nec(A, B) = inf_{x \in X} \{ Máx(A(x), 1 - B(x)) \} \quad (2.2)$$

*Compatibilidad:* Dio como resultado que tan compatibles son los valores que relacionan a un conjunto con respecto a otro y se aplicó la ecuación 2.3.

$$Comp(B, A)(u) = sup_{u=A(x)} \{ B(x) \}, \quad u \in [0, 1] \quad (2.3)$$

*Entropía:* Facilitó obtener el grado de incertidumbre en la información, el cálculo se realizó bajo la ecuación 2.4.

$$H(A) = 2 \sum_{i=1}^n |A(x_i) - A_{acorte}(x_i)| \quad (2.4)$$

*La concentración:* Brinda una idea más clara con respecto al tipo de función de pertenencia que se debe utilizar, tomando en cuenta la cantidad de píxeles en una imagen que corresponde a un mismo valor.

El resultado de las operaciones y considerando como parámetros la concentración y la tonalidad de grises con respecto al universo de discurso (0-255), se obtuvieron los siguientes conjuntos y sus rangos:

Defecto sin brillo	0-50
Defecto con brillo	30-80
Área buena sin brillo tono 1	60-110
Área buena sin brillo tono 2	85-150
Área buena sin brillo tono 3	120-190
Área buena con brillo tono 1	160-230
Área buena con brillo tono 2	205-255

Las funciones de pertenencia usadas para este proyecto fueron la función característica de hombro izquierdo para el conjunto de *Defecto sin brillo*. Figura 3, la función característica de hombro derecho para el conjunto *área buena con brillo Tono 2*. Figura 4.

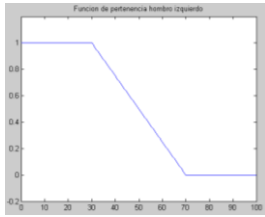


Figura 3. Función de pertenencia hombro izquierdo

$$f(x) = \begin{cases} x \leq \alpha \rightarrow & 1 \\ \alpha \leq x \leq \beta \rightarrow & \frac{\alpha - x}{\beta - \alpha} \\ x \leq \beta \rightarrow & 0 \end{cases}$$

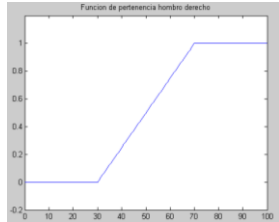


Figura 4. Función de pertenencia hombro derecho

$$f(x) = \begin{cases} x \leq \alpha \rightarrow & 0 \\ \alpha \leq x \leq \beta \rightarrow & \frac{x - \alpha}{\beta - \alpha} \\ x \leq \beta \rightarrow & 1 \end{cases}$$

Para los conjuntos de *defecto con brillo*, *área buena sin brillo tono 1*, *área buena sin brillo tono 2*, *área buena sin brillo tono 3* y *área con brillo tono 1* son de la forma trapezoidal definida por sus límites inferior, superior y los límites de su soporte, inferior y superior respectivamente. Se muestra en la figura 5.

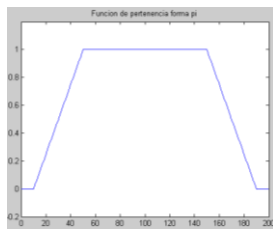


Figura 5. Función de pertenencia trapezoidal

$$f(x) = \begin{cases} \alpha \leq x \leq \beta \rightarrow & \frac{\alpha - x}{\alpha - \beta} \\ \beta \leq x \leq \varphi \rightarrow & 1 \\ \varphi \leq x \leq \delta \rightarrow & \frac{\delta - x}{\delta - \varphi} \\ 0 \text{ de otra manera} \end{cases}$$

Al aplicar estas funciones de pertenencia y relacionándolas con los conjuntos fue posible obtener la representación gráfica del conocimiento de la imagen, que se visualiza en la figura 6.

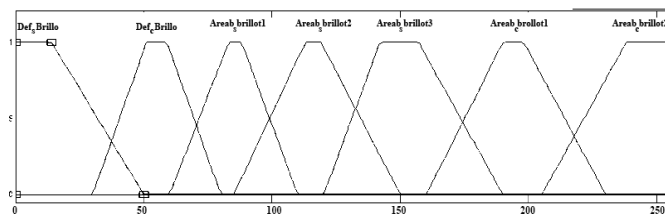


Figura 6. Representación del conocimiento

En la figura 6, se observan los 7 conjuntos obtenidos con sus respectivas funciones de pertenencia, que interpreta la representación del conocimiento.

De aquí en adelante se tratará el diseño de la metodología para la toma de decisiones acerca del estado del balón.

### 3. MÉTODO DE SOLUCIÓN

El método consiste:

1. Homogeneizar los datos de entrada
2. Fuzzificar (codificar) los datos x de entrada por bloques de 27 x 27
3. Evaluar cada uno de los valores de entrada de los bloques con respecto a cada una de las reglas mediante el producto de los grados de pertenencia  $A_{ij}(x)$  de los antecedentes que componen a las reglas difusas.
4. Defuzzificar (decodificar) los datos de salida por bloques promediando los valores de sus grados de pertenencia en cada regla.

Para tener una idea más clara sobre el funcionamiento del sistema en la figura 7 se mapea la arquitectura del sistema.

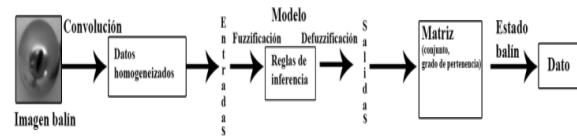


Figura 7. Arquitectura del sistema

#### 3.1. Fuzzificación

Se diseñó un algoritmo que permitiera el recorrido de la imagen mediante bloques de 27 x 27 debido a que el contenido de pixeles de este bloque puede determinar si existe o no un defecto en la imagen.

Posterior a esto fue necesario determinar la conectividad entre los pixeles. Se aplicó un filtro máscara que utiliza el concepto de vecindad es decir que cambia el contenido de pixeles en el punto medio de la matriz máscara [6], la cual tiene dimensiones n x m y ocho elementos en ella que permitieron homogeneizar los datos en cada uno de los bloques de la matriz y sirvieron para la entrada al sistema. Esto se puede ejemplificar en la figura 8.

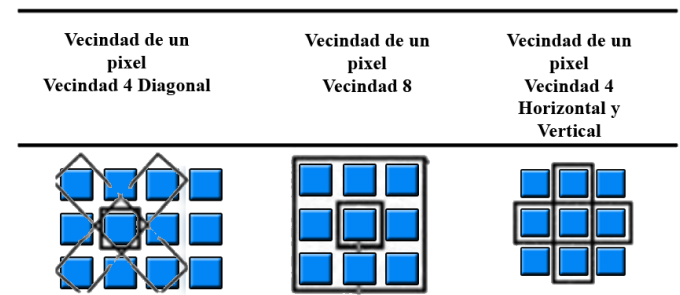


Figura 8. Representación de vecindades en una matriz

La vecindad de 8 es la que fue utilizada para el sistema debido a que esta tiene la propiedad de usar los elementos alrededor

del pixel central y multiplicar ocho elementos de la imagen, con ocho elementos de la máscara que se sobreponen en la imagen, esto se ejemplifica en la Figura 9.

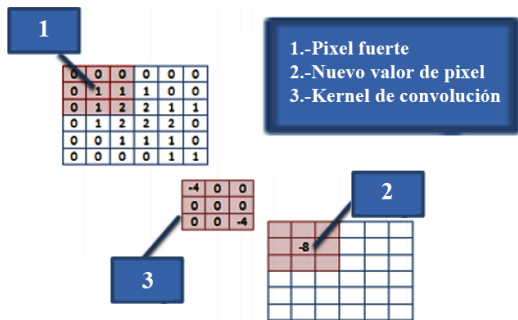


Figura 9. Operación aplicando vecindad de 8

#### Algoritmo 1: Datos de entrada homogeneizados

Leer bloque de imagen  
 Declarar kernel

for y igual a uno hasta m de la longitud de un bloque  
 for x igual a uno hasta m de la longitud de un bloque  
     Posición en x va hacer igual a x más la resta de  
     jr\_m menos valor medio de x  
     Posición en y va hacer igual a y más la resta de  
     ir\_n menos valor medio de y  
     dato homogeneizado va ser igual a posición en  
     x y y  
 fin del for x  
 fin del for y

#### 3.2 Evaluación Difusa

Para que la evaluación de los datos sea posible es necesario el diseño de las reglas base en el motor de inferencia.

Las reglas de inferencia: Consiste en una colección de reglas de tipo si-entonces expresadas de la siguiente forma:

Si X es C1 entonces Y

Este módulo de inferencia es un sistema que mapea un conjunto de entradas en un conjunto de salidas como se muestra en la figura 1. En otras palabras el motor de inferencia difusa es el encargado de calcular el valor global de la variable de salida, basado en contribuciones individuales de cada regla. Las reglas fueron formuladas por el conocimiento del especialista, en este caso el operador encargado de inspeccionar la calidad del balón. Para el sistema se utilizó una entrada con 7 conjuntos y una salida con el mismo número de conjuntos, estos son identificados propiamente, debido a que la regla base es derivada desde fuentes disponibles que contienen un conjunto de reglas relacionando la entrada con la salida usando niveles lingüísticos predefinidos. En este proyecto se diseñaron las reglas de inferencia por cada dato de entrada (*superficie*), a continuación se muestran algunas de las reglas utilizadas en el sistema.

Regla 1: Si *superficie* es defecto sin brillo entonces estado es balón malo

Regla 2: Si *superficie* es defecto con brillo entonces estado es balón malo

Regla 3: Si *superficie* es área buena tono 1 entonces estado es balón bueno

En la figura 10 se muestra la matriz de reglas que se generan a partir de las reglas.

Regla1 = [1 1 1 1 1 1 1]  
 Regla2 = [2 1 1 1 1 1 1]  
 Regla3 = [3 2 1 1 1 1 1]

Figura 10. Matriz de reglas

Cuando se evalúa cada uno de los valores del bloque, se obtiene una matriz con los datos defuzzificados (decodificados).

#### 3.3 Defuzzificación

Después de la evaluación el sistema se obtiene una matriz que representa a los conjuntos y los grados de pertenencia a los que cada uno de los datos de entrada corresponde. Al obtener esta matriz con los datos previamente defuzzificados fue necesario tomar en cuenta lo siguiente.

1. Si después de la evaluación de algún bloque se obtiene como resultado *bloque defectuoso* el algoritmo termina dando como respuesta *balón defectuoso*
2. Cuando se han terminado de evaluar todos los bloques de la imagen y el estado de todos los bloques haya sido *bloque bueno* el método termina obteniendo como resultado *balón bueno*
3. Mientras no se haya obtenido un bloque defectuoso es decir, que el estado de algún bloque sea *bloque bueno* y no se hayan terminado de analizar todos los bloques de la imagen se sigue ejecutando el proceso.

#### 4. PRUEBAS Y RESULTADOS EXPERIMENTALES

Las pruebas se realizaron sobre imágenes de balines que produce la empresa Tzubaki de CA. Los balines contienen defectos tales como: abolladuras, fisuras, tajadas entre otras en diferentes tamaños y formas. En la figura 11 se puede apreciar algunas imágenes de balines con y sin defectos para el experimento.



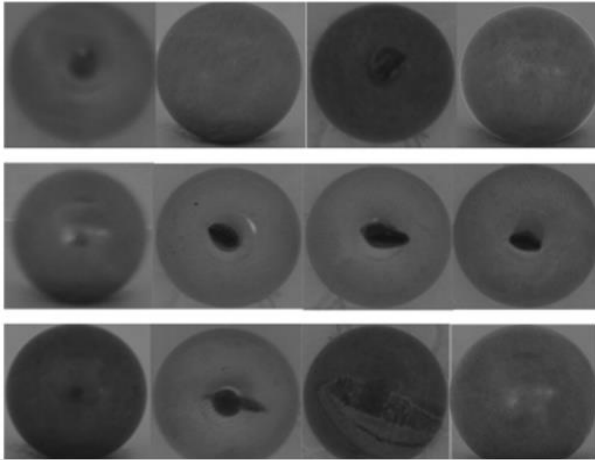


Figura 11. Imágenes de balines para el experimento

Se modificaron las condiciones del entorno, para la hacer la adquisición de imágenes. Se reducen los problemas de reflejo en el balín, originadas por las características de la superficie metálica [8]. Las modificaciones que se hicieron son: establecer una iluminación homogénea en la zona de inspección, utilizar el color blanco anti reflejante como fondo en todo el prototipo, construir una cubierta para el prototipo de color blanco y determinar la posición de la cámara, para la adquisición de las imágenes. La operación y evaluación del sistema se realizó utilizando las funciones de sistemas difusos del toolbox de Fuzzy logic de MATLAB. En la figura 12 se aprecia el entorno del toolbox de Fuzzy logic.

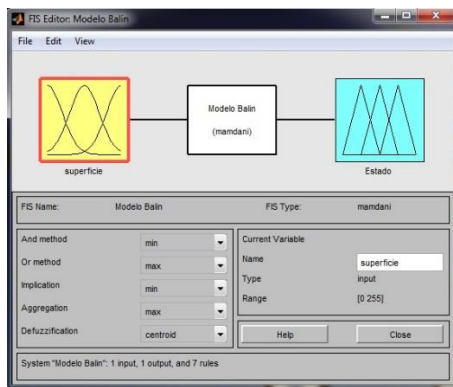


Figura 12. Toolbox de fuzzy logic de MATLAB

En la figura 13 se muestran los balines con los que se ilustrarán algunas pruebas realizadas en el sistema.



Figura 13. Balines que se utilizaron para las pruebas

Al decodificar cada valor de la imagen se obtiene un dato crisp como se visualiza gráficamente en la figura 14.

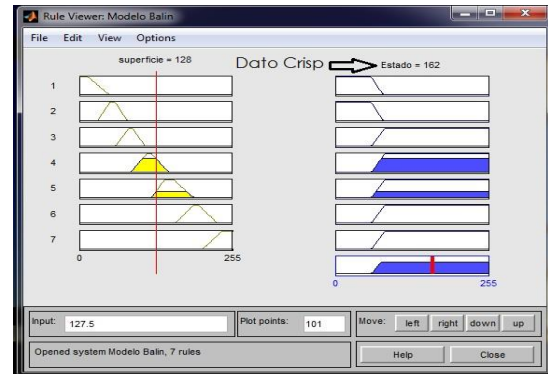


Figura 14. Visualizador de reglas

Después de obtener los datos decodificados se forma una matriz con datos crisp de 27x27 para cada uno de los bloques de la imagen, esto se muestra en la figura 15.

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
26	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
27	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figura 15. Matriz de datos crisp

Para validar si el bloque contiene un defecto o no, se contabilizan el número de datos de la matriz mostrada en la figura 14, que pertenezcan al conjunto del defecto. Con ello es posible que el sistema tome una decisión acerca del estado del balín.

En la tabla 1. Se muestran algunos resultados de procesar las imágenes en el sistema.

Tabla 1. Resultados de las pruebas

Numero de Balín	Resultado	Observaciones
Balín 1	Defectuoso	El defecto se encontró en el bloque 56
Balín 2	Defectuoso	El defecto se encontró en el bloque 45
Balín 3	Bueno	Se recorrió toda la matriz

## 5. ANALISIS DE EFICIENCIA DEL ALGORITMO

Cuando se dispone de un algoritmo que funciona correctamente, es necesario definir criterios para medir su rendimiento o comportamiento. Estos criterios se centran

principalmente en su simplicidad y en el uso eficiente de los recursos [9].

Para optimizar el tiempo de ejecución de un algoritmo se consideró para las pruebas un procesador Intel® Core™ i7-3770 CPU con una velocidad de procesamiento 3.40GHz, permitiendo obtener un tiempo de 0.2 segundos por imagen, alcanzando un tiempo total de 5.6 horas para validar la calidad del 20% de la producción total de balines.

Para este proyecto se hizo un análisis previo con respecto a la eficiencia de los recursos que se miden en función de dos parámetros: el espacio de memoria que utiliza, y el tiempo, que tarda en ejecutarse. Ambos parámetros representan los costes que supone encontrar la solución al problema planteado mediante un algoritmo.

El orden de complejidad de este proyecto es polinomial  $O(n^a)$ , donde  $n$  es el número de entradas y  $a$  representa el número de conjuntos, siendo tratable para las clases de orden polinomial.

## 6. CONCLUSIONES

En este artículo se desarrolló un método para detectar defectos en balines de la empresa Tsubaki CA, usando técnicas de procesamiento digital de imágenes creadas mediante un sistema de lógica difusa.

El tiempo obtenido para cada bloque de  $27 \times 27$  de la imagen es de  $5.06 \times 10^{-04}$  segundos. Analizando 100 bloques por cada imagen y considerando 8 frames para en el análisis de cada balín. Este tiempo permitió optimizar el proceso de inspección de calidad de balines utilizado por el operador de la empresa.

El proyecto aporta una metodología para la aplicación de sistemas difusos en el procesamiento digital de imágenes, con esto se pretende profundizar en la aplicación de nuevas tecnologías inteligentes.

## 7.- AGRADECIMIENTOS

Agradecemos a la Secretaria de Investigación y Estudios de Posgrado de la Facultad de Ciencias de la Computación de la BUAP por el apoyo para la difusión de los resultados de este proyecto.

## 8. REFERENCIAS

- [1] Tsubaki CA <http://www.tsubaki.ca/>
- [2] Luime Martínez Corral, Martín Vázquez Rueda, Francisco Flores García, Sistema de visión con lógica difusa para la identificación de regiones de interés en plantas, Instituto Tecnológico de la Laguna, Torreón Coahuila México, 29 de setiembre de 2006.
- [3] González C. Rafael, Woods E. Richard; Tratamiento digital de imágenes; Editorial Addison-Wesley/Díaz De Santos, Versión en español, Wilmington, Delaware, E.U.A 1996
- [4] Sr. Arístides Bouza, Ing. Omar Alimenti, Microcontroladores Estándar en el Desarrollo de Sistemas de Lógica Difusa, Instituto de Ingeniería y Ciencias en Computación, 1997
- [5] Galindo Gómez, Lógica Difusa y Aplicaciones, Departamento de lenguajes y Ciencias de la Computación, Universidad de Malaga <http://www.lcc.uma.es/~ppgg/FSS/FSS5.pdf>
- [6] Víctor Daniel. Sánchez N, Análisis a-priori del algoritmo de convolución bidimensional aplicado a imágenes biomédicas, Maestría en Ciencias área Cibernética de la Facultad de Ingeniería

[7] Mario Mauricio Bustillo Díaz, Leopoldo Bustillo Eguiluz, Bárbara Sánchez Rinza, Alejandro Rangel Huerta, Automatización de la etapa de inspección en la producción de balines, Facultad de Ciencias de la computación BUAP, Puebla México 2008

[8] Chang Jiang LI, Zhong ZHANG, Tetsuo MIYAKI, Takashi IMAMURA and Hisanaga FUJIWA. Processing Specular Reflection Components of Chrome-plated Surface by Multi-Image Reconstruction Method. Ed. International Journal of Computer Information Systems and Industrial Management Applications. ISSN: 2150-7988 Vol.1 (2009), pp.303-311.

[9] Rosa Guerequeta y Antonio Vallecillo, Técnicas y diseño de algoritmos, Universidad de Málaga ISBN-84-7496-666-3 Segunda edición, 1998.