

**CORPORACIÓN MEXICANA DE INVESTIGACIÓN EN MATERIALES**

**DIVISIÓN DE ESTUDIOS DE POSGRADO**



**“Integración de la red neuronal convolucional con el algoritmo de función de frontera de objeto para reconocimiento de piezas y detección de defectos.”**

**TESIS**

**Para obtener el grado académico de**

**Maestro en Ciencia y Tecnología en Manufactura Avanzada**

**POR**

**Ing. Yanier Basulto Rodríguez**

**SALTILLO, COAHUILA.**

**12 de diciembre de 2018**

“Integración de la red neuronal convolucional con el algoritmo de función de frontera de objeto para reconocimiento de piezas y detección de defectos.”

Por

Yanier Basulto Rodríguez

TESIS

Presentada al programa Interinstitucional en Ciencia y Tecnología

Sede

**Corporación Mexicana de Investigación en Materiales S.A. de C.V.**

Como requisito parcial para obtener el grado académico de

**Maestro en Ciencia y Tecnología en Manufactura Avanzada**

**Programa Interinstitucional en Ciencia y Tecnología**

**COMIMSA**

Saltillo Coahuila, 12 de diciembre de 2018

“Integración de la red neuronal convolucional con el algoritmo de función de frontera de objeto para reconocimiento de piezas y detección de defectos.”

Por

Yanier Basulto Rodríguez

Comité Revisor

**Director de Tesis / Tutor Académico:**

Dra.

Pamela Chiñas Sánchez

Instituto Tecnológico de Saltillo

**Revisor No. 1:**

Dra.

Alejandra Hernández Rodríguez

Corporación Mexicana de Investigación en Materiales S.A. de C.V.  
(COMIMSA)

**Revisor No.2:**

Dr.

José Luis Navarro González

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico  
Nacional Unidad Saltillo (CINVESTAV)

**Tutor de Planta:**

Dr.

Pedro Pérez Villanueva

Corporación Mexicana de Investigación en Materiales S.A. de C.V.  
(COMIMSA)

**Corporación Mexicana de Investigación en Materiales S.A de C.V**

**Gerencia de Desarrollo Humano**

**División de Estudios de Posgrado**

Los abajo firmantes, miembros del Comité Tutorial recomendamos que la Tesis **“Integración de la Red Neuronal Convolutiva con el algoritmo de Función de Frontera de Objeto para reconocimiento de piezas y detección de defectos.”**, realizada por el alumno **Yanier Basulto Rodríguez** matrícula **1706MA1129** sea aceptada para su defensa como Maestro en Ciencia y Tecnología en Manufactura Avanzada.

**El Comité Tutorial**

---

**Dra. Pamela Chiñas Sánchez**

Directora de Tesis

Tutor Académico

---

**M.C. José Alejandro**

**Almeda Rivas**

Asesor

---

**Dr. Pedro Pérez**

**Villanueva**

Tutor de planta

---

Vo. Bo

**Dr. Pedro Pérez Villanueva**

Coordinación General de Estudios de Posgrado

COMIMSA

**Corporación Mexicana de Investigación en Materiales S.A de C.V**

**Gerencia de Desarrollo Humano**

**División de Estudios de Posgrado**

Los abajo firmantes, miembros del Jurado del Examen de Grado del alumno, **Yanier Basulto Rodríguez** una vez leída y revisada la tesis titulada “**Integración de la red neuronal convolucional con el algoritmo de función de frontera de objeto para reconocimiento de piezas y detección de defectos.**”, aceptamos que la referida tesis revisada y corregida sea presentada por el alumno para aspirar al grado de Maestro en Ciencia y Tecnología en Manufactura Avanzada.

Y para que así conste firmamos la presente a los 12 días del mes de diciembre del año 2018.

---

Dr. José Luis Navarro González  
Presidente

---

Dra. Alejandra Hernández Rodríguez.  
Secretaria

---

Dra. Pamela Chiñas Sánchez  
Vocal

## Agradecimientos

1. CONACYT por las becas otorgadas y la oportunidad de superarme profesionalmente.
2. COMIMSA por las becas otorgadas, los profesores que contribuyeron en mi formación y el apoyo que brindaron en el desarrollo de la investigación.
3. Mi tutora, Dra. Pamela Chiñas Sánchez por el apoyo brindado en la investigación.
4. A mi esposa Susana por su apoyo incondicional.
5. A mis padres y hermana por darme la confianza de seguir adelante en mi desarrollo profesional.
6. A familiares y amigos que han compartido conmigo a lo largo de la maestría sus experiencias y conocimientos.

## ÍNDICE

<b>1. INTRODUCCIÓN</b> .....	2
<b>2. PLANTEAMIENTO DEL PROBLEMA</b> .....	5
2.1. Descripción del problema. ....	5
2.2. Hipótesis.....	9
2.3. Objetivos. ....	9
2.3.1. Objetivo General. ....	9
2.3.2. Objetivos Específicos. ....	9
2.4. Preguntas de Investigación. ....	10
2.5. Justificación. ....	10
2.6. Alcance y delimitaciones. ....	11
<b>3. CAPÍTULO ESTADO DEL ARTE</b> .....	14
<b>4. CAPÍTULO MARCO TEÓRICO</b> .....	18
4.1. Pre-procesamiento de la imagen a través del BOF. ....	20
4.1.1. Características paramétricas 2D ..... 20	20
4.1.2. Función de frontera BOF ..... 21	21
4.2. Redes neuronales artificiales ..... 22	22
4.2.1. Backpropagation. .... 24	24
4.2.1.1. Regla de aprendizaje ..... 25	25
4.2.2. Función de base radial..... 27	27
4.3. Máquinas de Soporte Vectorial..... 29	29
4.4. Redes Neuronales Convolucionales ..... 32	32
<b>5. METODOLOGÍA</b> . .... 37	37
5.1. Selección del equipamiento para el sistema físico. .... 37	37
5.2. Configuración adecuada para los elementos del sistema. (Estación de trabajo) ..... 38	38
5.3. Generar algoritmo para adquisición de imágenes mediante el software LABVIEW. .... 38	38
5.4. Adquisición de imágenes. .... 39	39
5.5. Diseño del algoritmo BOF. .... 39	39
5.6. Diseño de red neuronal CNN. .... 40	40
5.7. Diseño de la arquitectura del sistema de inspección. .... 40	40
5.8. Validación. .... 41	41
<b>6. EXPERIMENTACIÓN Y RESULTADOS</b> ..... 42	42
6.1. Adquisición de imágenes..... 42	42

6.2.	<i>Etapa de identificación del tipo de pieza.</i>	45
6.2.1.	<i>Implementación del algoritmo BOF.</i>	46
6.2.2.	<i>Implementación de la CNN para la identificación de piezas.</i>	52
6.2.3.	<i>Clasificación de los vectores descriptores obtenidos con el algoritmo BOF y la CNN.</i>	56
6.2.3.1.	<i>Clasificación de los vectores descriptores del BOF con la red Backpropagation.</i>	56
6.2.3.2.	<i>Etapa de clasificación de la CNN: Patrones generados por la extracción de características de CNN y analizados con Backpropagation.</i>	58
6.2.3.3.	<i>Etapa de clasificación de la CNN: Patrones generados por la extracción de características de CNN y analizados con RBF.</i>	60
6.3.	<i>Etapa de detección de defectos de las piezas.</i>	61
6.3.2.	<i>Implementación de la CNN para la detección de defectos.</i>	61
6.3.3.	<i>Implementación de SVM en la arquitectura de la CNN para la detección de defectos.</i>	62
6.4.	<i>Arquitectura del sistema de inspección propuesto.</i>	65
6.5.	<i>Validación del sistema.</i>	65
7.	<b>CONCLUSIONES Y TRABAJO FUTURO.</b>	67
8.	<b>BIBLIOGRAFÍA.</b>	70

### **ÍNDICE DE TABLAS**

<b>Tabla 6. 1</b>	<b>equivalencia entre las categorías de las etiquetas y las piezas a analizar.</b>	<b>43</b>
<b>Tabla 6. 2</b>	<b>equivalencia entre las categorías de las etiquetas y los defectos de las piezas a analizar.</b>	<b>44</b>
<b>Tabla 6. 3</b>	<b>clasificación de la red backpropagation en la detección de piezas con los vectores del bof.</b>	<b>58</b>
<b>Tabla 6. 4</b>	<b>entrenamiento de la backpropagation en la red cnn.</b>	<b>59</b>
<b>Tabla 6. 5</b>	<b>validación de las técnicas propuestas.</b>	<b>66</b>

### **ÍNDICE DE FIGURAS**

<b>Figura 2. 1</b>	<b>Imagen original y resultado de detección de la CNN. (Weimer et al., 2016).</b>	<b>7</b>
<b>Figura 4. 1</b>	<b>Representación de dos puntos en el plano de la imagen. Elaboración propia.</b>	<b>21</b>
<b>Figura 4. 2</b>	<b>Función de frontera de un cuadrado y un círculo. Comparación del BOF de dos figuras regulares de dimensiones aproximadas. Elaboración propia.</b>	<b>22</b>
<b>Figura 4. 3</b>	<b>Arquitectura general de una RNA (Demuth et al., 2014).</b>	<b>23</b>
<b>Figura 4. 4</b>	<b>Arquitectura de una red RBF. Elaboración propia.</b>	<b>27</b>
<b>Figura 4. 5</b>	<b>Clasificación de datos mediante el hiperplano óptimo (Morales, G., &amp; Gómez, 2005).</b>	<b>29</b>



<b>Figura 4. 6 Proyección de un espacio de entradas a uno de mayor dimensión. Elaboración propia. ....</b>	<b>31</b>
<b>Figura 4. 7 Arquitectura de una CNN (Diego Calvo, 2017). ....</b>	<b>35</b>
<b>Figura 5. 1 Metodología. Elaboración propia. ....</b>	<b>37</b>
<b>Figura 5. 2 Infraestructura de la estación de captura y procesamiento de la imagen. Elaboración propia. ....</b>	<b>38</b>
<b>Figura 5. 3 Arquitectura de la CNN. Elaboración propia. ....</b>	<b>40</b>
<b>Figura 6. 1 Piezas de entrenamiento. Bisagras y Soportes de lámparas. ....</b>	<b>43</b>
<b>Figura 6. 2 Defectos etiquetados. ....</b>	<b>45</b>
<b>Figura 6. 3 Pasos lógicos para la identificación del tipo de piezas. ....</b>	<b>45</b>
<b>Figura 6. 4 Comparación entre las técnicas de umbral y algoritmo de Canny aplicadas. ....</b>	<b>47</b>
<b>Figura 6. 5 Resultado del procedimiento para determinar el área. ....</b>	<b>48</b>
<b>Figura 6. 6 Puntos del perímetro del objeto. ....</b>	<b>50</b>
<b>Figura 6. 7 Patrones descriptores de la bisagra y el Soporte de lámparas. ....</b>	<b>50</b>
<b>Figura 6. 8 Comparación entre la operación max-pooling y la operación mean-pooling. ....</b>	<b>51</b>
<b>Figura 6. 9 Ejemplos de procesamiento de imágenes aplicando diferentes filtros en la etapa de convolución. ....</b>	<b>54</b>
<b>Figura 6. 10 Gráfico que representa un vector de las bisagras y un vector de los soportes de lámparas. ....</b>	<b>55</b>
<b>Figura 6. 11 Esquema de la red Backpropagation utilizada para la identificación de las piezas. ....</b>	<b>57</b>
<b>Figura 6. 12 Gráfico que representa dos vectores de defectos de los soportes de lámparas. ....</b>	<b>62</b>
<b>Figura 6. 13 Reducción de la imagen al área de interés. ....</b>	<b>63</b>
<b>Figura 6. 14 Clasificación con el Toolbox del Matlab Classification Learner. ....</b>	<b>64</b>
<b>Figura 6. 15 Arquitectura del sistema de inspección propuesto. ....</b>	<b>65</b>

**RESUMEN:**

Uno de los principales retos en la industria son los sistemas de inspección que aseguran la calidad del producto en el menor tiempo posible. La investigación propone un nuevo sistema de reconocimiento de defectos centrado en la integración de técnicas relevantes en su aplicación; la Función de Frontera de Objetos (BOF) por su rapidez en la segmentación de imágenes, la Red Neuronal Convolutiva (CNN) la cual logra una mayor extracción de características en las imágenes y las Máquinas de Soporte Vectorial (SVM) con la capacidad de clasificar datos linealmente no separables. Debido a que la extracción profunda de características (CNN) utiliza un mayor procesamiento, se implementa el BOF para reconocer las diferentes piezas elaboradas por una manufactura flexible y la CNN+SVM para el reconocimiento de sus defectos. Se logró un sistema más robusto de detección de defectos en comparación con las técnicas aplicadas por separado en cuanto a la detección de defectos en piezas a través de las imágenes reduciendo el tiempo de procesamiento.

Palabras Claves: Red Neuronal Convolutiva, BOF, SVM, detección de defectos

Campo específico: Redes Neuronales.

## 1. INTRODUCCIÓN

Actualmente para las empresas es necesario manufacturar productos con altos estándares de calidad considerando grandes volúmenes de producción, por lo tanto, es preciso desarrollar nuevas tecnologías que permitan aumentar la producción y al mismo tiempo aseguren la calidad del producto (López et al., 2005). El control de calidad es un factor importante en la industria para garantizar que los productos cumplan con los estándares establecidos por los clientes. Algunas empresas necesitan procesos complejos de calidad que demandan una gran cantidad de recursos humanos y económicos; éstos representan un costo significativo para las empresas. Por esta razón, se han desarrollado sistemas de visión artificial para automatizar los procesos de inspección de calidad con el propósito de detectar los defectos y reducir los costos (Yu et al., 2013), (Cai et al., 2017).

Tradicionalmente los sistemas de inspección visual son llevados a cabo por el personal de la empresa que trabaja como máquinas altamente especializadas, donde la calidad de los productos de la empresa está en manos del ser humano y su percepción visual (López et al., 2005). Adicionalmente, la importancia de los sistemas de visión artificial ha crecido desde los años 70's (Mauricio et al., 2007) y se han aplicado en diferentes campos de la ingeniería tales como celdas de ensamblaje robotizada (Lopez et al., 2010), en estudios biométricos (Galdámez et al., 2017), en la medicina (Kawahara et al., 2017), para el reconocimiento de escritura a mano (Elleuch et al., 2016), en la inspección de soldaduras (Nian Cai et al., 2016), en la inspección de fusibles térmicos (Sun et al., 2016) y la inspección de micro brocas en la producción de PCB (Printed Circuit Board, por sus siglas en inglés),(Duan et al., 2012).

El enfoque basado en la visión artificial ha traído grandes resultados en la búsqueda de la solución a los problemas de inspección visual en las industrias. Bajo este contexto ha sido útil el empleo de diferentes técnicas de inteligencia artificial, donde se destacan tanto la red FuzzyARTMAP como los algoritmos basados en máquinas de soporte vectorial (SVM, por sus siglas en inglés), demostrando mejores resultados por su precisión y tiempo de respuesta con respecto a otras redes utilizadas anteriormente como la Backpropagation y la Perceptron (Lopez et al., 2010), (Navarro et al., 2015), (Lopez-Juarez, 2012), (Rios-Cabrera et al., 2012), (Cai et al., 2017).

Las empresas manufactureras se enfocan cada vez más hacia la búsqueda de la excelencia, lo que requiere de procesos con mayor velocidad de procesamiento, mayor calidad de sus productos y la reducción de su costo-beneficio con la capacidad de localizar las fuentes de defectos en la producción, incluso con el proceso en curso (Castelán et al., 2013). En este sentido surge la necesidad de aplicar técnicas que ayuden a mejorar los sistemas convencionales de inspección.

Los avances recientes en el desarrollo de las redes neuronales artificiales muestran el uso del aprendizaje profundo, en particular, los métodos que utilizan un modelo de Red Neuronal Convolutiva (CNN, por sus siglas en inglés); los cuales han obtenido resultados satisfactorios en tareas de visión por computadoras y un rendimiento adecuado en la clasificación de la imagen, por su capacidad de extraer una gran cantidad de información de las características del objeto usando en su configuración un mayor número de descriptores en comparación con otras redes neuronales (Gando et al., 2016).

En este trabajo se propone el diseño de un sistema de inspección basado en las ventajas de la red CNN en el procesamiento de imágenes, debido a que muestran una mejora

significativa en la precisión de clasificación y en el manejo de mayor volumen de información en imágenes en comparación con las técnicas usadas hasta el momento como FuzzyARTMAP y SVM (D. C. Cirezan et al., 2011), (D. Cirezan et al., 2012). La combinación de esta red con métodos de extracción de características ha demostrado un mayor rendimiento en el reconocimiento de patrones (Smirnov et al., 2014), (Liu et al., 2015), (Elleuch et al., 2016), (Guo et al., 2016), (Kamnitsas et al., 2017). Uno de los métodos de extracción de características que puede ser usado en un proceso de reconocimiento de imágenes junto a la red CNN es el algoritmo de función de fronteras de objetos (BOF por sus siglas en inglés), debido a su capacidad de reconocimiento invariante del contorno de objetos y su capacidad de proveer una segmentación apropiada entre el objeto y el fondo. El algoritmo de función de fronteras de objetos se ha utilizado con diferentes redes neuronales y clasificadores de imágenes, proporcionando un método robusto para el aprendizaje en los sistemas de visión artificial (Peña-Cabrera et al., 2004), (Lopez et al., 2010), (Castro et al., 2010), (Lopez et al., 2010), (Cabrera et al., 2010), (Lopez-Juarez, 2012), (I López-Juárez et al., 2013), (Iorio et al., 2014), (Navarro-Gonzalez et al., 2015), por lo que se propone el uso del BOF como vector descriptor construido de patrones recurrentes que permite al clasificador determinar las diferentes formas 2D (contorno) de los objetos.

El propósito de esta investigación es diseñar un método novedoso que considere el diseño de un sistema de visión robusto, flexible y con factibilidad para su implementación y operación, que represente un avance hacia las aplicaciones en tareas de manufactura a través del reconocimiento de imágenes con técnicas conocidas como aprendizaje profundo, para alcanzar mayor precisión en la determinación de la calidad de los productos.

## **2. PLANTEAMIENTO DEL PROBLEMA**

### *2.1. Descripción del problema.*

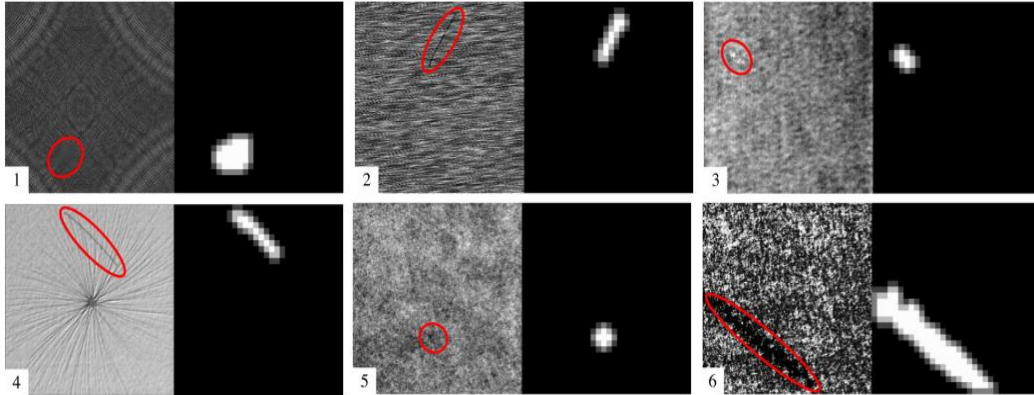
El control de la calidad es esencial en todos los procesos productivos, de ahí que, principales organizaciones internacionales de estándares como: la ISO (Organización Internacional de Estándares) y la IEC (Comisión Electrotécnica Internacional) establecen conjuntos de normas que regulan los diferentes campos de producción (López et al., 2005).

Para asegurar la calidad en los productos, es de vital importancia identificar los defectos generados durante el proceso manufactura (Grágeda et al., 2007). La inspección visual realizada por el ser humano puede tener efectos negativos en la toma de decisiones, ya que es una tarea tediosa y subjetiva. La percepción humana depende mucho de las condiciones y tiempo de trabajo (Guillermo Manuel et al., 2011).

El reconocimiento de patrones en datos correspondiente a imágenes de productos manufacturados, permite encontrar características descriptivas de éstos y la clasificación de los mismos en diferentes categorías, ejemplo de ello serían el tipo de pieza manufacturada en un sistema de producción flexible y los diferentes defectos de fabricación que podrían presentar dichas piezas (Camila, 2017). Con la manufactura flexible surge la necesidad de adaptación de algoritmos para la variabilidad que se presenta tanto en los productos como en las celdas de producción, lo que representa uno de los problemas fundamentales en los sistemas de inspección por computadora.

Existen diversas técnicas de visión artificial en el área de inspección de procesos, que tienen por objetivo reconocer y localizar objetos mediante el procesamiento de imágenes, teniendo como finalidad extraer características para su descripción e

interpretación (Baptista et al., 2013), (Guo et al., 2016). En los sistemas de inspección visual artificial se han empleado diferentes arquitecturas para el reconocimiento de imágenes, desde las más tradicionales como las redes Perceptron y Backpropagation, hasta técnicas más modernas como la red FuzzyARTMAP por su precisión y tiempo de procesamiento (Lopez, 2012) y SVM que es un clasificador muy eficiente. Estas técnicas necesitan para su reconocimiento y clasificación, patrones con características diferenciables. Los patrones con alto grado de similitud tienen una mayor probabilidad de errores en su reconocimiento. Además, los defectos superficiales pueden ocurrir en tamaños arbitrarios, forma u orientación, por lo que los descriptores de características de este tipo a menudo conducen a resultados de clasificación insuficientes (Weimer et al., 2016). Con el surgimiento del aprendizaje profundo aparecen métodos como las redes neuronales convolucionales (CNN) que alcanzaron resultados con mayor grado de precisión respecto a los métodos mencionados anteriormente (Ciresan et al., 2011), (Ciresan et al., 2012); además de contar con la capacidad de detectar diferentes defectos en imágenes más complejas en cuanto a textura, color, forma, resolución de píxeles, entre otras características. En la figura 2.1 se muestra la imagen original y el resultado de detección con la red CNN, aplicada a imágenes de 512 x 512 píxeles, donde cada índice en la imagen representa defectos de diferentes tipos sobre las distintas superficies.



*Figura 2. 1 Imagen original y resultado de detección de la CNN. (Weimer et al., 2016).*

La red CNN se utiliza para abordar con éxito tareas de reconocimiento de imagen altamente complejas con diversas clases de objetos, basadas en la aplicación de funciones que permiten resaltar determinados elementos de la imagen como contornos, mejora de enfoques o reducir el ruido de fondo. Estas funciones son denominadas filtros y se basan en operaciones de convolución, lo que simplifica el cálculo matemático y los tiempos de procesamiento. Al aplicar un filtro de convolución a la imagen de entrada se tiene como resultado una nueva imagen con los puntos más significativos para su reconocimiento y clasificación sin la necesidad de realizar un pre procesamiento riguroso en las imágenes. El rendimiento de CNN se puede incrementar al modificar la red CNN usando algunas técnicas de extracción de características y otros clasificadores diferentes a los definidos en su arquitectura original. (Santoni et al., 2015).

Existen diferentes soluciones de sistemas de inspección a través de la visión artificial, dependiendo de las necesidades del cliente se define el costo y las técnicas a aplicar para el diseño del sistema de inspección. En México existen varias empresas que brindan estos servicios, pero en su mayoría son empresas extranjeras que implican un costo más elevado de sus servicios, principalmente para las pequeñas y medianas empresas. Gran



parte de estos sistemas actualmente son sistemas estructurados, enfocados en exigencias muy particulares del cliente, donde se diseña un algoritmo para el reconocimiento de objetos y dependiendo de la pieza se tienen diseñados los algoritmos que detectan los defectos señalados por el mismo. En estos casos, los sistemas de inspección por visión artificial no son suficientemente robustos en la detección de un nuevo defecto no programado, clasificando una pieza defectuosa como una en calidad aceptable.

En este estudio se presenta como propuesta en el área de inspección visual, la implementación de una estación de control de calidad orientada a la inspección y verificación de especificaciones de fabricación de piezas, basada en visión artificial, capaz de identificar y distinguir entre los diferentes productos procesados en un sistema flexible de manufactura y asegurar una mayor precisión en la calidad de producción.

Ante las problemáticas mencionadas anteriormente con los sistemas de inspección, se pretende emplear técnicas que posean un gran potencial en la extracción profunda de características y en el reconocimiento de objetos, con el objetivo de aportar nuevas investigaciones que logren el diseño de sistemas de inspección con visión artificial más robustos en cuanto al análisis de imagen ante variaciones en las intensidades de los píxeles provocadas por cambios de luz y la detección de defectos no previstos por en el sistema, accesibles a todas las empresas mexicanas a un menor costo.

Por tanto, se pretende integrar la red CNN para la detección de defectos en productos manufacturados debido su capacidad de extracción de características en imágenes y el algoritmo de obtención de la función frontera de objeto en la identificación del tipo de piezas, por su capacidad en determinar contornos. El vector descriptor del BOF se determina definiendo la distancia desde el centroide del objeto hasta los puntos de su

perímetro (Castro et al., 2010), lo que proporciona un menor tiempo de procesamiento en el reconocimiento de patrones.

## *2.2. Hipótesis.*

Mediante la integración del algoritmo de función de frontera de objetos (BOF) y la red neuronal convolucional (CNN) será posible diseñar un sistema de reconocimiento de patrones para detección de defectos; con el propósito de emplear menor tiempo de procesamiento basado en el algoritmo BOF como técnica de segmentación y la extracción profunda de características relevantes al aplicar la red CNN.

## *2.3. Objetivos.*

### *2.3.1. Objetivo General.*

Integrar el algoritmo de función de frontera de objeto a la red neuronal convolucional, para el diseño de un sistema reconocimiento de patrones capaz de detectar diferentes tipos de piezas y sus defectos en un proceso de manufactura.

### *2.3.2. Objetivos Específicos.*

- Determinar los tipos y cantidad máscaras convolucionales necesarias para resaltar las características relevantes de las imágenes.
- Determinar las potencialidades del BOF y la CNN para el reconocimiento de piezas referente a tiempo de procesamiento y tamaño del vector descriptor.
- Sintetizar el tamaño de la imagen al área significativa de análisis, sin reducir las dimensiones de captura de imagen.
- Determinar el método de clasificación adecuado que garantice la identificación de piezas y sus defectos con una precisión mayor al 80%.

- Establecer las pruebas de validación del sistema de inspección.

#### *2.4. Preguntas de Investigación.*

- ¿Cómo se ha utilizado la red CNN en los sistemas de visión artificial?
- ¿Cómo se ha aplicado el algoritmo BOF en los sistemas de visión artificial?
- ¿Cuál es la arquitectura adecuada de la CNN para el análisis de defectos en la imagen?
- ¿Cuántas capas ocultas serán necesarias para realizar la clasificación esperada?
- ¿Cómo modificar la red CNN con el algoritmo BOF para obtener un mejor rendimiento de la red?
- ¿Qué tipo de máscaras de convolución se utilizarán para resaltar las características deseadas en la imagen?
- ¿El algoritmo BOF será capaz de mejorar el tiempo de procesamiento de la red?
- ¿Cómo validar la metodología aplicada?

#### *2.5. Justificación.*

Los sistemas de inspección industrial son de gran utilidad en las empresas manufactureras, ellos forman parte del sistema de control de la calidad y la estabilidad de los procesos de producción.

Existen diversas técnicas para el reconocimiento de imágenes aplicadas a los sistemas de inspección. Con la metodología propuesta en la tesis se pretende dar solución a uno de los factores que influye en gran medida sobre los sistemas de inspección actual, como es la precisión en el reconocimiento de objetos y de sus características principales que determinan su calidad. La nueva arquitectura que se propone (CNN + BOF) permitirá generar conocimiento a través de la integración de dos técnicas que presentan grandes potencialidades en el procesamiento de imagen: la primera, la red neuronal convolucional

para propósito de aprendizaje en el reconocimiento de imágenes con patrones similares, integrando al sistema de inspección el algoritmo de función de frontera de objeto (BOF), que posee la capacidad del reconocimiento invariante de objetos mediante el uso de características de contorno, forma y profundidad, las cuales son concatenadas para formar un vector descriptivo que le proporcionará los patrones de entrada al método utilizado para su clasificación , evitando un mayor tiempo de procesamiento de la misma (Lopez, 2012).

El desarrollo y el perfeccionamiento de la nueva arquitectura basada en la integración de la red CNN con el BOF pretende dar paso a diseños de sistema de inspección que presenten como ventajas potenciales, la eliminación de los errores cometidos por la inspección visual humana; el aumento de la velocidad de inspección para estar acorde con las velocidades de producción (López et al., 2010); la reducción de pérdidas en producción; la detección en tiempo de defectos en los productos, evitando el consumo de materia prima en las próximas etapas del proceso y garantizando un mejoramiento de la relación costo – beneficio.

En la actualidad las empresas manufactureras necesitan del empleo de sistemas de inspección con las características antes mencionadas, es importante sin duda alguna impulsar mediante el desarrollo científico y tecnológico, el crecimiento de las industrias y garantizar un aumento en el nivel de competitividad en el mercado nacional e internacional.

#### *2.6. Alcance y delimitaciones.*

Alcance: Integración de la arquitectura de la red neuronal convolucional (CNN) con el algoritmo de obtención de fronteras de objetos (BOF) para el reconocimiento de piezas y la detección de defectos.

Delimitaciones:

- Uso del algoritmo BOF y la red CNN como técnicas bases de la investigación.
- El sistema de iluminación, el modelo de cámaras y el tipo de piezas para el entrenamiento y validación de la red serán las proporcionadas por la corporación COMIMSA.
- Esta investigación se enfocará en la parte superior de las piezas.
- El procesamiento y la detección de defectos en las piezas no se realizará en tiempo real de producción.
- La investigación está proyectada entre el nivel TRL 2 y 3 de las etapas de maduración tecnológicas. Se ha identificado el potencial de los materiales y cuestiones de ampliación de escala. Se están validando los componentes de la tecnología, aunque todavía no se plantea integrar los componentes en un sistema completo.
- No se cuenta con un sistema comercial para realizar pruebas de factibilidad con respecto al sistema propuesto.

### *2.7. Aportación científica.*

Desarrollo de un sistema de reconocimiento de patrones de defectos basado en el algoritmo BOF y la red CNN.

### *2.8. Aportación tecnológica.*

Una estación de inspección para realizar estudios en sistemas de visión artificial.

### *2.9. Resultados y entregables esperados.*

- Resultados: La arquitectura con la integración del algoritmo BOF con la red CNN para la detección de defectos e identificación de piezas.

- Entregables: Código de programación de la (CNN+BOF) y una estación de trabajo para posteriores investigaciones.

### 3. *CAPÍTULO ESTADO DEL ARTE*

En este capítulo se hace una descripción del desarrollo de los sistemas de visión artificial para el reconocimiento de imágenes. Se expone la evolución de las redes neuronales respecto a tiempo de procesamiento y precisión.

El reconocimiento de objetos a través de los sistemas de visión artificial es uno de los problemas más desafiantes en visión por computadora, especialmente a gran escala y en ajustes realistas, con imágenes de alta resolución y numerosas categorías de objetos (Smirnov et al., 2014). En un sistema de visión se utilizan modelos de predicción basados en datos de entrenamiento, lo que permite al sistema incrementar su conocimiento, obteniendo así un proceso eficiente en el reconocimiento de imágenes (Peña et al., 2004). El mayor número de referencias encontradas muestran que las técnicas de aprendizaje más utilizadas en los sistemas de inspección son las Máquinas de Soporte Vectorial y las Redes Neuronales Artificiales, estas son empleadas para el entrenamiento y clasificación de datos (El-Naqa et al., 2002).

La visión artificial a través de las redes neuronales presenta un gran avance en la década de 1940 con el trabajo de Warren McCulloch y Walter Pitts (McCulloch et al., 1943), con el que demostraron que las redes neuronales artificiales podrían calcular función aritmética o lógica. La primera aplicación práctica de redes neuronales artificiales se produjo a finales de la década de 1950, con el diseño de la red Perceptron y la regla de aprendizaje asociada por Frank Rosenblatt (Rosenblatt, 1958), en la cual se manifestó su capacidad para realizar el reconocimiento de patrones, pero la red Perceptron básica sólo podría resolver una clase limitada de problemas.

En los años ochenta, se desarrolló el algoritmo de Backpropagation para el entrenamiento de redes Perceptron Multicapa, este avance se llevó a cabo de manera independientemente por varios investigadores simultáneamente. La publicación más influyente del algoritmo Backpropagation fue por David Rumelhart y James McClelland (Rumelhart, 1986).

A partir de los noventa, se implementa el algoritmo FuzzyARTMAP que incorpora lógica difusa para mejorar el rendimiento, especialmente con ruidos en los patrones de entrada (Carpenter et al., 1992). Este enfoque demuestra su idoneidad para implementarse en operaciones de tiempo real en los sistemas de inspección por sus resultados experimentales obtenidos al compararlo con otros modelos como la red Backpropagation y la Perceptron, mostrando mayor porcentaje en la tasa de reconocimiento de eficiencia global y la razón de reconocimiento en menor tiempo (Lopez et al., 2012), (Lopez, 2012).

Yann Lecun y Yoshua Bengio introducen la Red Neural Convolutiva (CNN) inspirados en el funcionamiento de los nervios en la corteza visual de los gatos, que es la parte del cerebro responsable del procesamiento de la imagen. La red CNN fue diseñada para reconocer patrones con diversas variaciones, además de presentar robustez en la clasificación ante la distorsión geométrica y la transformación simple (Santoni et al., 2015). Los investigadores han comenzado a emplear este método en la búsqueda de soluciones más precisas en el reconocimiento de patrones (Stutz, 2014). Lucas Bondi y colaboradores, abordan el problema de la optimización de las características de precisión de la tasa de energía de una arquitectura convolutiva para el reconocimiento facial (Bondi et al., 2016). Los autores Pratt y colaboradores proponen una red CNN para diagnosticar la retinopatía diabética a partir de imágenes de fondo digital, por su complejidad en la precisión de la



clasificación, tales como micro-aneurisma, exudados y hemorragias en la retina (Pratt et al., 2016). Sung Wook Baik y otros autores, proponen un mecanismo de selección de características orientadas a objetos para características convolucionales profundas de un CNN pre-entrenado. Los mapas de características convolucionales de una capa profunda se seleccionan en función del análisis de sus respuestas ante los diferentes filtros o kernels. Las características seleccionadas sirven para representar los patrones que definen a los objetos o rasgos del mismo como contorno, textura, color entre otros parámetros que los identifican (Ahmad et al., 2018). Se ha empleado la CNN como herramienta computacional para el aprendizaje automático, particularmente en el área de la Visión por Computadora, específicamente en la tarea de predicción del sentimiento a través de las imágenes. Se logra a través de experimentos de ajuste usando una CNN y mediante rigurosos análisis de arquitectura, presentando varias modificaciones que conducen a mejoras de precisión en un conjunto de datos de imágenes de una plataforma de redes sociales. Además, se presenta visualizaciones de patrones locales que la red aprendió a asociar con el sentimiento representado en la imagen para obtener una idea de cómo el modelo percibe la positividad visual (Campos et al., 2017).

Con la evolución de la investigación sobre la arquitectura de la red CNN se demuestra que el rendimiento de la misma es apto a mejora con la integración de otros métodos existentes de extracción de características de imagen. En uno de los artículos revisados se presenta un sistema de identificación de ganado bovino a partir de sus imágenes, donde la unión de la red CNN con la Matriz de Co-ocurrencia de Nivel Gris (GLCM, por sus siglas en inglés) muestra una mayor precisión de clasificación con un menor número de iteraciones de aprendizaje que la red CNN original (Santoni et al., 2015).

Mohamed Elleuch, Rania Maalej y Monji Kherallah hacen un reconocimiento de manuscritos árabes usando la red CNN para la extracción de características de la imagen y como clasificador a la salida de la red un SVM por su robustez en el aprendizaje automático (Elleuch et al., 2016). En el trabajo realizado por Kamnitsas y otro grupo de investigadores se utiliza la combinación de la red convolucional con una técnica ampliamente usada en la segmentación de imagen, Campos Condicionales Aleatorios (CRF, por sus siglas en inglés) para el análisis de lesiones cerebrales (Kamnitsas et al., 2017).

Como se ha mostrado en trabajos anteriores, la combinación de la red CNN con técnicas de extracción de características presenta ventajas en el desempeño en tareas como el reconocimiento y clasificación de imágenes, por tanto, en este trabajo se pretende el diseño de un sistema de inspección visual para reconocer patrones con diversas variaciones y transformaciones geométricas, con un alto grado de precisión y con el menor tiempo posible de procesamiento de datos. Para lograr las especificaciones mencionadas anteriormente, se propone en este trabajo de tesis la integración de la red CNN como técnica del aprendizaje profundo para la clasificación de imágenes y el algoritmo de Función de Frontera de Objeto (BOF) a la entrada de la primera capa de la red como vector descriptor de la imagen que refleja la forma 2D (contorno) del objeto.

El algoritmo BOF por su capacidad de reconocimiento invariante de objetos y la información que proporciona con la segmentación entre el objeto y el fondo, se ha empleado integrado a diferentes clasificadores de imágenes para procesos robotizados de reconocimiento de objeto y ensamblaje de piezas en tiempo real (Cabrera et al., 2010), (Navarro et al., 2015).

#### **4. *CAPÍTULO MARCO TEÓRICO.***

En el presente capítulo se mencionan las etapas que conforman el sistema de inspección, así como, el soporte teórico que sustenta el desarrollo de los procesos de reconocimiento y procesamiento de imágenes. La ciencia de la información ha alcanzado en la actualidad un desarrollo en la creación de computadoras más veloces y con mayor capacidad de procesamiento para el perfeccionamiento de los sistemas de inspección basados en computadoras, logrando que sean cada vez más autónomos. En 1959, Arthur Samuel definió el aprendizaje de máquina como “el campo de estudio, que da a las computadoras la capacidad de aprender sin necesidad de ser programada de forma explícita” (BELL, 2015), (Camila, 2017).

Proveer a las computadoras de capacidades sensoriales es una tarea muy complicada. La habilidad del ser humano referente a la visión ha sido un interés en los temas de investigación para dotar a los sistemas computacionales de este sentido del hombre y lograr sistemas capaces de tomar decisiones a partir del reconocimiento de imágenes (Lopez et al., 2010).

La visión artificial representa a partir de una imagen bidimensional la estructura y propiedades del mundo tridimensional. A partir de imágenes digitales obtenidas por diferentes dispositivos de captura de imagen, la visión artificial permite el procesamiento y análisis de la información para su interpretación, lo que contribuye la toma de decisiones de determinado proceso (Ratha et al., 1999), (Tim, 2008).

Los diseñadores de sistemas de visión utilizan diferentes métodos con el objetivo de obtener descripciones significativas de la escena de visión, donde se determinan las propiedades locales de la imagen (niveles, color y textura), los modelos geométricos (frontera, región, superficies y otros) y la descripción semántica de la escena con el objetivo de dividirla en objetos específicos para ser caracterizados y clasificados (Cabrera et al., 2010), (Peña-Cabrera et al., 2004).

Los sistemas de visión con aprendizaje a partir de un vector de entrada buscan reconocer el comportamiento de una variable de salida, la clasificación de estos datos depende de la técnica de entrenamiento utilizada ya sea un aprendizaje no supervisado, semi-supervisado o supervisado. El aprendizaje supervisado es uno de los más usados debido a que se conocen los valores de entradas y sus salidas correspondientes, lo que permite en la etapa de entrenamiento realizar ajustes en los parámetros basándose en los errores de clasificación en esta etapa. Para determinar si el sistema de entrenamiento obtuvo los resultados esperados, se ingresan nuevos datos y se analiza que tenga la capacidad de predecir la salida correspondiente con los nuevos datos de entrada. Algunas de las técnicas de aprendizaje de máquina más usadas son los métodos de Kernel (como las máquinas de soporte vectorial, SVM), las redes neuronales artificiales y el aprendizaje profundo (deep learning) (Wernick et al., 2010). En el desarrollo de los sistemas de aprendizaje de máquinas en ocasiones se incluye una etapa de pre-procesamiento de datos que permita encontrar los patrones descriptores con los que serán entrenadas las máquinas para una mejor clasificación.

#### 4.1. Pre-procesamiento de la imagen a través del BOF.

Para la digitalización de imágenes se utilizan dispositivos de captura como cámaras fotográficas o de videos con el objetivo de reconocer las formas de imágenes 2D, a partir de objetos en 3D, llevando de una escena tridimensional del mundo real a una bidimensional.

Con la imagen digitalizada, se busca como siguiente resultado una imagen binarizada, esto se consigue eliminando el ruido con algún tipo de filtro y una comparación de umbrales. Este procedimiento permite segmentarla en dos regiones: figura y fondo (Ismael López-Juárez et al., 2012).

##### 4.1.1. Características paramétricas 2D

Un objeto con determinada forma geométrica, se puede caracterizar a partir de sus propiedades métricas, creando un vector descriptivo que la identifique. Las características métricas se basan en una medida de la distancia  $d(V1, V2)$  entre dos puntos  $V1$  y  $V2$  del plano de la imagen como se muestra en la figura 4.1. Del teorema de Pitágoras se deduce la distancia Euclidiana:

$$dE(V1, V2) = \sqrt{(V2x - V1x)^2 + (V2y - V1y)^2} \quad (4.1)$$

Dónde:  $(V1x, V1y)$  son las coordenadas del punto  $V1$  y  $(V2x, V2y)$  son las coordenadas del punto  $V2$ , dentro de un mismo plano (Ayala, 2014).

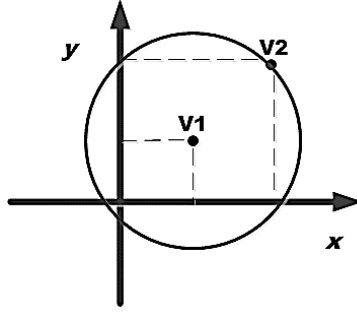


Figura 4. 1 Representación de dos puntos en el plano de la imagen. Elaboración propia.

Entre las propiedades métricas se encuentra el área y el perímetro, donde el área es la suma de la cantidad de píxeles que pertenecen a la forma y el perímetro es la cantidad de píxeles que pertenecen al contorno (Ayala, 2014).

$$A = \sum_i \sum_j \text{Píxeles}(i, j) \in \text{forma} \quad (4.2)$$

$$P = \sum_i \sum_j \text{Píxeles}(i, j) \in \text{contorno} \quad (4.3)$$

Otra de las características es el centroide, punto que define el centro geométrico de un objeto:

$$X_c = \frac{\sum_{x,y} j}{A}, Y_c = \frac{\sum_{x,y} i}{A} \quad (4.4)$$

Donde  $A$  es el área del objeto y  $(X_c, Y_c)$  son las coordenadas del centroide.

#### 4.1.2. Función de frontera BOF

El BOF está formado por un vector que contiene las diferentes distancias entre el centroide y cada uno de los puntos tomados en el contorno de objeto. Para cada uno de los  $n$  puntos que conforman el contorno  $(P_x, P_y)$ , se calcula la distancia de cada punto al centroide  $(C_x, C_y)$ :

$$BOF(n) = \sqrt{(P_x - C_x)^2 + (P_y - C_y)^2} \quad (4.5)$$

La función de frontera es un parámetro que permite poder diferenciar las formas de las figuras resultantes de objetos 3D capturados en imagen 2D. En la figura 4.2 se observa las diferencias entre los patrones de distancias obtenidos con el BOF, de un cuadrado y un círculo al ser graficados en un eje de coordenadas. Los puntos del contorno determinados se representan con la letra *P*, el centroide se representa con la letra *C* y la letra *D* son las distancias euclidianas calculadas que representan al vector BOF.

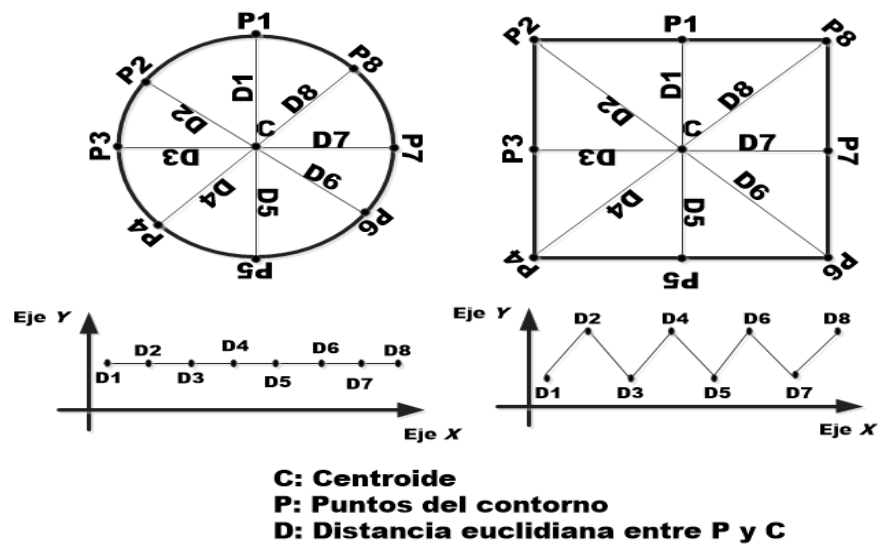


Figura 4. 2 Función de frontera de un cuadrado y un círculo. Comparación del BOF de dos figuras regulares de dimensiones aproximadas. Elaboración propia

La función de frontera permite obtener un vector numérico que conlleva a patrones característicos que diferencian las formas entre sí. El vector característico definido por el algoritmo BOF que representa el contorno de un objeto, determinará la entrada a un sistema de reconocimiento de patrones basado en RNA.

#### 4.2. Redes neuronales artificiales

Las RNA (Redes Neuronales Artificiales) son sistemas que imitan el esquema funcional de las neuronas del cerebro humano y tratan de emular su comportamiento,

intentando reproducir el aprendizaje de una red neuronal para realizar una determinada tarea, a través de representaciones matemáticas.

La arquitectura general de estos sistemas está compuesta por una capa de entrada que recibe toda la información descriptiva de los parámetros a clasificar; las capas utilizadas entre la capa de entrada y la de salida se les denomina capas ocultas y realizan el procesamiento de la información de la RNA; la capa de salida del sistema es donde las neuronas procesan la clasificación deseada. El trabajo fundamental consiste en determinar la topología adecuada que permita el correcto funcionamiento de la RNA (Hernández et al., 2017). Las conexiones entre las unidades de procesamiento (neuronas), poseen pesos numéricos que pueden ser modificados según las entradas que reciba la red, permitiéndole adaptarse y aprender. Las capas ocultas, contienen funciones de activación que determinan estado activo o no activo de la neurona. Las funciones de activación de las neuronas de la capa de salida obtienen un valor que identifica al vector de entradas con las posibles clases etiquetadas.

La figura 4.3 muestra la arquitectura general de una red neuronal artificial.

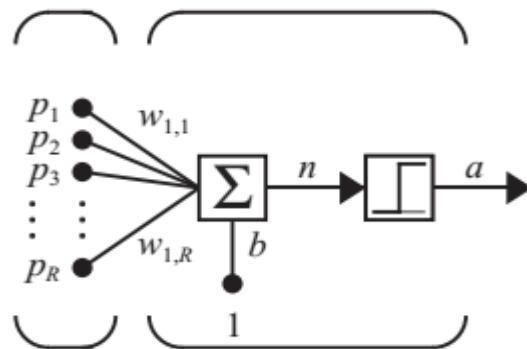


Figura 4. 3 Arquitectura general de una RNA (Demuth et al., 2014).



Donde el vector  $p$  representa las entradas,  $w$  los pesos sinápticos,  $b$  es el valor del bias,  $n$  es el valor neto que integra la información proveniente de las distintas neuronas y  $a$  es la salida de la función de activación. La ecuación general de este esquema está dada por:

$$\mathbf{a} = \mathbf{f}(\mathbf{n}) = \mathbf{f}(\mathbf{w}^T \times \mathbf{p} + \mathbf{b}) \quad (4.6)$$

Existen varias funciones de activación ( $a$ ), entre las que se destacan las funciones continuas como:

Lineal

$$a_i = f_i(n_i) = n_i \quad (4.7)$$

Sigmoidal

$$a_i = f_i(n_i) = \frac{1}{1+e^{-n_i}} \quad (4.8)$$

Tangente hiperbólica

$$a_i = f_i(n_i) = \frac{e^{n_i} - e^{-n_i}}{e^{n_i} + e^{-n_i}} \quad (4.9)$$

Gaussiana

$$a_i = f_i(n_i) = e^{-\frac{(n_i - \mu)^2}{2\sigma_i^2}} \quad (4.10)$$

Existen diferentes arquitecturas de RNA, en este trabajo se hará referencia a algunas de las más empleadas en los sistemas de visión encontradas en la literatura.

#### 4.2.1. *Backpropagation.*

El algoritmo de retropropagación (Backpropagation) es un entrenamiento supervisado que se emplea para redes multicapa, donde se ajusta el valor de los pesos en

función del error generado en la salida de la red, lo que permite que los pesos sinápticos de cada neurona de la capa oculta, se actualicen respecto a su inferencia en el error calculado, durante el entrenamiento. La variación en la neurona de salida al modificarse el valor de los pesos se le llama sensibilidad. Esta técnica es muy empleada ya que permite tener un método de optimización, que minimiza el gradiente del error con respecto a los parámetros de la red neuronal (Olabe, 1998),(Cruz, 2010).

#### 4.2.1.1. Regla de aprendizaje

El aprendizaje de la red de retropropagación utiliza un ciclo de propagación-adaptación al conjunto de entradas-salidas. Inicialmente el vector de entradas se propaga a través de todas las capas de la red hasta encontrar la salida del sistema, ecuación (4.11) (Demuth, Beale, Jess, & Hagan, 2014):

$$\mathbf{a}^M = \mathbf{f}^M(\mathbf{W}^M \mathbf{f}^{M-1}(\mathbf{W}^{M-1} \dots \mathbf{f}^2(\mathbf{W}^2 \mathbf{f}^1(\mathbf{W}^1 \mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) \dots + \mathbf{b}^{M-1}) + \mathbf{b}^M) \quad (4.11)$$

Donde  $M$  es el número total de capas que conforman la red.

Ecuación recursiva

$$\mathbf{a}^m = \mathbf{f}^m(\mathbf{W}^m \mathbf{a}^{m-1} + \mathbf{b}^m) \quad (4.12)$$

$$\mathbf{a}^0 = \mathbf{p}, \quad m \in [1, \dots M] \quad (4.13)$$

Donde  $m$  es el valor específico de la capa que se analiza.

El error generado entre la salida de la red y el valor deseado se determina:

$$e_q = \mathbf{t}_q - \mathbf{f}^M(\mathbf{n}^M) \quad (4.14)$$

$\mathbf{t}_q$  es el valor deseado de salida.

Con el error calculado se determina la sensibilidad de la capa de salida:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^m)e_q \quad (4.15)$$

Donde  $\dot{\mathbf{F}}^m(n^m)$  representa el gradiente del error para cada neurona que pertenece a una capa de procesamiento y se calcula aplicando la derivada a cada función de activación con respecto a la entrada neta ( $n$ ).

$$\begin{aligned} \dot{\mathbf{F}}^m(n^m) &= \frac{df^m}{dn^m} = \begin{bmatrix} \frac{df_1^m}{dn^m} & \dots & \frac{df_{s^m}^m}{dn^m} \end{bmatrix} \\ &= \begin{bmatrix} \frac{df_1^m}{dn_1^m} & 0 & \dots & 0 \\ 0 & \frac{df_2^m}{dn_2^m} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \frac{df_{s^m}^m}{dn_{s^m}^m} \end{bmatrix} = \text{diag} \left[ \frac{df_i^m}{dn_i^m} \right] \end{aligned} \quad (4.16)$$

Con el valor de sensibilidad de la capa de salida, que representa el error correspondiente a cada neurona, se realiza una propagación del error hacia atrás con el objetivo de que las neuronas intermedias reciban un porcentaje de error aproximado a su intervención en la salida original. Los pesos sinápticos y el bias se actualizan en función de la sensibilidad obtenida en cada capa de procesamiento.

$$\mathbf{W}^m = \mathbf{W}^m - \alpha s^m (\mathbf{a}^{m-1})^T \quad (4.17)$$

$$\mathbf{b}^m = \mathbf{b}^m - \alpha s^m, \forall m \in [1 \dots M] \quad (4.18)$$

Donde  $\alpha$  es el factor de aprendizaje ( $0 < \alpha < 1$ ) y  $s$  es la de sensibilidad de retropropagación.

Otra de las redes empleadas en la búsqueda de una solución en el trabajo realizado fue la función de base radial (RBF, por sus siglas en inglés).

#### 4.2.2. Función de base radial

La red RBF presenta una arquitectura compuesta por una capa de entrada, una capa oculta y una capa de salida. La capa oculta está completamente interconectada entre todos sus nodos con la capa de entrada, las neuronas de esta capa son activadas dependiendo de una función radial (gaussiana) y la capa de salida completamente interconectada a la capa oculta es activada por una función lineal continua (Demuth et al., 2014). La figura 4.4 muestra la arquitectura de la RBF donde  $x$  son las entradas,  $C$  son los centros de la función,  $W$  son los pesos sinápticos,  $a$  es la salida de la función de la capa oculta y  $y$  es la salida de la red.

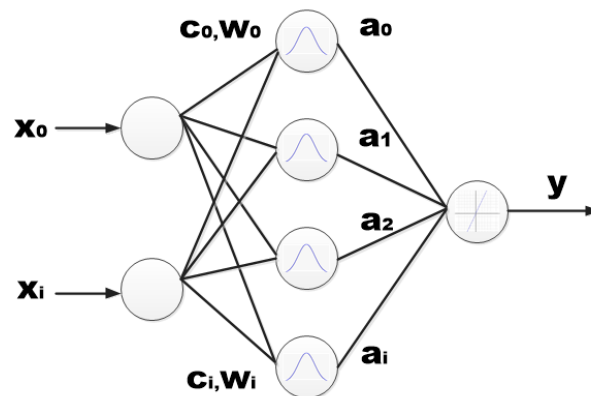


Figura 4. 4 Arquitectura de una red RBF. Elaboración propia.

Las funciones de base radial se rigen por los parámetros centro ( $c_i$ ) y ancho ( $d_i$ ), el primero es donde la función alcanza un nivel cercano al valor máximo de activación. El segundo es el término que define la amplitud originada por la campana de gauss de la función (Segal et al., 2000).

Inicialmente en la capa oculta se calcula la distancia radial (distancia euclidiana) ( $d_i$ ), entre el vector de entrada  $x$  con  $n$  observaciones y el centro de gravedad ( $c_i$ ) de cada neurona.

$$d = \|x - c\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2} \quad (4.19)$$

Entre la capa oculta y la capa de salida se derivan un conjunto de pesos  $w$  que se verían afectados de acuerdo al algoritmo de aprendizaje. En este caso particular, sería la combinación lineal entre los pesos y la resultante de cada función radial para determinar el valor de salida de la red ( $y$ ).

$$y = \sum w_i G(\bullet) \quad (4.20)$$

$$G(\bullet) = e^{\frac{-(\|x_j - c_{ij}\|)^2}{2d_i^2}} \quad (4.21)$$

Donde  $G(\bullet)$  es la matriz que contiene a las funciones gaussianas de la capa oculta.

En la capa oculta, en la medida que los valores de entrada se acerquen más a un centro su distancia tenderá a cero y de este modo la función gaussiana se dispararía a las vecindades de uno. Por otro lado, en la medida que los valores de entrada no se asemejen a su centro, la distancia será mayor y la función radial tendería a cero. Este proceso es una clasificación no lineal de las entradas.

Uno de los algoritmos de aprendizaje de la RBF utilizados es la pseudo-inversa en el cual se calculan los pesos  $w$  a partir de la siguiente expresión:

$$w = G^{-1}S \quad (4.22)$$

Donde  $S$  es el vector de las salidas deseadas y  $G^{-1}$  es la inversa de la matriz de funciones gaussianas de la capa oculta.

Las RNA son muy usadas en el reconocimiento de patrones, no obstante, existen otras técnicas que han demostrado ser un gran clasificador como las máquinas de soporte vectorial.

### 4.3. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial (SVM) están basadas en la búsqueda de hiperplanos que separen un conjunto de datos pertenecientes a diferentes clases, existe un único hiperplano de separación óptimo (HSO) donde la distancia entre el hiperplano y el patrón más cercano a él es máxima. En la figura 4.5 (a) se representa la separación de los datos a través de un hiperplano y en la (b) se muestra la separación de los datos utilizando el hiperplano óptimo. La figura muestra dos hiperplanos  $P_{-1}$  y  $P_{+1}$  que determinan el tipo de clases a la que pertenece una entrada,  $P_0$  es el hiperplano óptimo donde el margen es la distancia máxima entre HSO y los hiperplanos  $P_{-1}$  y  $P_{+1}$ ,  $w$  son los pesos de la función y  $b$  es el bias (Morales et al., 2005), (Carrillo, 2007).

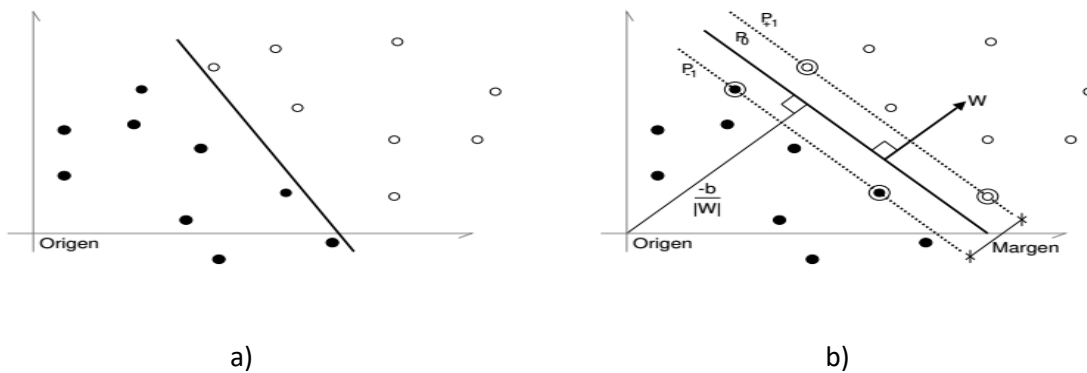


Figura 4. 5 Clasificación de datos mediante el hiperplano óptimo (Morales, G., & Gómez, 2005).

La función de clasificación para un caso linealmente separable, con un vector  $x$  que representa las entradas y las etiquetas de salida deseada a clasificar  $y \in (-1, +1)$  se expresa como:

$$(\vec{w} * \vec{x}_i) + b \geq +1 \quad y_i = +1 \quad (4.23)$$

$$(\vec{w} * \vec{x}_i) + b \leq -1 \quad y_i = -1 \quad (4.24)$$

La función de decisión que determina la separación de los datos en las diferentes clases, se define como:

$$y_i = \text{sign}((\vec{w} * \vec{x}_i) + b) \quad (4.25)$$

Para encontrar el hiperplano óptimo se debe maximizar el margen, teniendo en cuenta la restricción que define que los datos son linealmente separables, se expresa como:

$$\min \frac{1}{2} (\vec{w} * \vec{w}) \quad (4.26)$$

$$\text{Sujeto a: } y_i (\vec{w} * \vec{x}_i) + b \geq 1, \forall i \quad (4.27)$$

Si existen datos erróneos, ruido o alto solapamiento de clases en los datos de entrenamiento, puede afectar el hiperplano clasificador óptimo. Por esta razón se busca el mejor hiperplano clasificador que pueda tolerar el ruido en los datos de entrenamiento. En este caso para buscar la linealidad de los datos se hace un mapeo de los datos de entrada al espacio características de una dimensión mayor utilizando una función kernel, figura 4.6. Entre las funciones kernel más utilizadas se encuentran la polinomial, la de base radial y la sigmoidea.

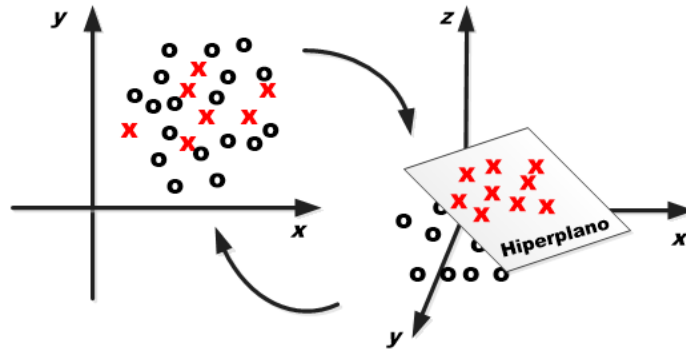


Figura 4. 6 Proyección de un espacio de entradas a uno de mayor dimensión. Elaboración propia.

En los casos donde los datos no son linealmente separables el hiperplano óptimo se determina por la siguiente expresión:

$$\min \frac{1}{2} (\vec{w} * \vec{w}) + C \sum_{i=1}^n \varepsilon_i \quad (4.28)$$

$$\text{Sujeto a: } y_i(\vec{w} * \vec{x}_i) + b \geq 1 - \varepsilon_i \quad \varepsilon_i \geq 0, \forall i \quad (4.29)$$

Donde  $\varepsilon_i \geq 0$  es el margen de error entre el punto  $i$  y los límites de separación. El parámetro  $C$  es elegido a priori por el usuario de tal manera que un valor grande es una alta penalización a los errores.

Los problemas con múltiples clases se manejan con diversos tipos de técnicas que llevan el procedimiento de una clasificación múltiple a una clasificación de dos clases.

Cuando se usan funciones de decisión continuas en lugar de discretas, como es el caso de la investigación propuesta, se resuelven las regiones no clasificadas con la técnica de clasificación múltiple “uno contra todos”. En el uno contra todos, un problema de  $n$ -clases se convierte en  $n$  problemas de dos-clases y para el  $i$ -ésimo problema de dos clases, la clase  $i$  es separada del resto de las clases, es decir, los patrones pertenecientes a la clase



$i$  toman valor de salida 1 y el resto de los patrones del resto de las clases toman valor de salida -1 formando la otra clase del proceso binario.

#### 4.4. Redes Neuronales Convolucionales

Otras de las arquitecturas de las RNA son las Redes Neuronales Convolucionales (CNN), las cuales son ampliamente aplicadas al reconocimiento de imágenes. Esta red presenta como ventaja con respecto a las redes neuronales totalmente conectadas tradicionales, un pre-procesamiento de datos a través de filtros de convolución, donde las características descriptivas contenidas en un vector de entradas a la red de gran dimensión, se mantengan en un vector de entradas a la red de menor tamaño sin perder los descriptores relevantes de la imagen.

Existen diferentes filtros para la detección de bordes, en el proceso de experimentación se utilizaron diferentes filtros para determinar una mejor definición de contornos en las piezas a analizar. Entre las máscaras usadas se encuentran los Laplacianos, gradiente de Prewitt y Sobel (Aldalur, 2002).

El filtro Laplaciano es una operación que resalta todos los bordes en una imagen, independientemente de sus orientaciones, donde se destacan las regiones donde hay cambios bruscos de intensidad. Las máscaras Laplaciana más comunes se muestran en los siguientes vectores:

$$[0 \ 1 \ 0; 1 \ -4 \ 1; 0 \ 1 \ 0]$$

$$[1 \ 1 \ 1; 1 \ -8 \ 1; 1 \ 1 \ 1]$$

$$[0 \ -1 \ 0; -1 \ 4 \ -1; 0 \ -1 \ 0]$$

$$[-1 \ -1 \ -1; -1 \ 8 \ -1; -1 \ -1 \ -1]$$

La operación del gradiente de Prewitt realiza un realce de borde direccional. Con una máscara de tamaño 3x3 aplicada a una imagen, pueden generarse imágenes de gradientes donde se resaltan los bordes orientados en diferentes direcciones, un ejemplo de las máscaras de convolución correspondientes se representan en los siguientes vectores:

$$[-1 \ -1 \ -1; 0 \ 0 \ 0; 1 \ 1 \ 1]$$

$$[-1 \ 0 \ 1; -1 \ 0 \ 1; -1 \ 0 \ 1]$$

La suma de los coeficientes que componen el filtro es igual a cero, lo que provoca que a medida que la máscara pasa sobre regiones de la imagen con valores muy cercanos entre sí, el resultado es cero (color negro) al igual que aquellos valores que se generen con números negativos. Aquellos bordes con la orientación direccional correcta en la imagen original, aparecen como blancos en la imagen de gradiente.

El filtro de Sobel detecta los bordes horizontales y verticales separadamente sobre una imagen en escala de grises. Al aplicar el operador Sobel sobre una región con intensidad de imagen constante, el resultado es un vector cero y el resultado de aplicarlo en un punto sobre un borde es un vector cuyo sentido es de los puntos más oscuros a los más claros. Los vectores que representan las máscaras de Sobel se muestran a continuación.

$$[-1 \ 0 \ 1; -2 \ 0 \ 2; -1 \ 0 \ 1]$$

$$[1 \ 2 \ 1; 0 \ 0 \ 0; -1 \ -2 \ -1]$$

Un filtro paso bajo tiene por efecto mantener los componentes de baja frecuencia espacial de una imagen. Los componentes de alta frecuencia están atenuados en la imagen

de salida. Si se aplica en una región donde los valores de intensidad de los píxeles cambian de 255 a 0 o viceversa, el resultado será un valor medio de gris entre 0 y 255. En las regiones donde los valores de intensidad son similares el resultado es el mismo valor de intensidad. El filtro de paso alto tiene un efecto opuesto al filtro paso bajos, este tipo de filtros pretende aislar los componentes de alta frecuencia en una imagen. Filtros paso bajo y paso alto.

$$[-1 \ -1 \ -1; \ -1 \ 9 \ -1; \ -1 \ -1 \ -1]$$

$$\left[\frac{1}{9} \ \frac{1}{9} \ \frac{1}{9}; \ \frac{1}{9} \ \frac{1}{9} \ \frac{1}{9}; \ \frac{1}{9} \ \frac{1}{9} \ \frac{1}{9}\right]$$

Otro de los filtros utilizados para la atenuación de la imagen es el de gauss, el filtro gaussiano se usa para eliminar el ruido que provoca algunos píxeles en la definición de la imagen. Un ejemplo de filtro gaussiano se representa en el siguiente vector.

$$\left[\frac{1}{16} \ \frac{1}{8} \ \frac{1}{16}; \ \frac{1}{8} \ \frac{1}{4} \ \frac{1}{8}; \ \frac{1}{16} \ \frac{1}{8} \ \frac{1}{16}\right]$$

Los filtros mencionados anteriormente son aplicados en la capa de convolución de la CNN, la cual posee en su arquitectura tres tipos básicos de capas: convolucional, pooling o sub-muestreo y la capa completamente conectada. Las dos primeras capas se encargan de la extracción de características de la imagen, las cuales forman el vector de entrada a la etapa de clasificación que corresponde a la capa completamente conectada tal y como se muestra en la figura 4.7 (Wagner et al., 2013).

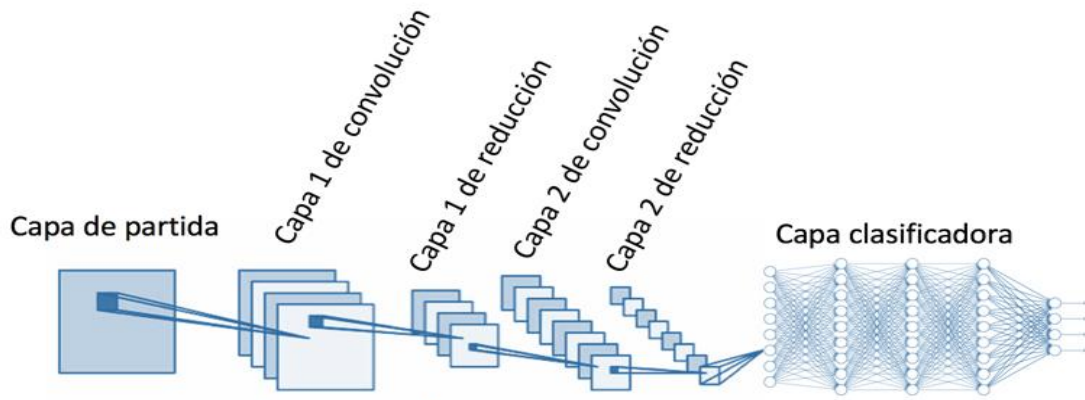


Figura 4. 7 Arquitectura de una CNN (Diego Calvo, 2017).

La capa convolucional realiza una transformación de la imagen original aplicando  $k$  filtros o kernels, llevando a cabo un proceso de convolución a través de ella y resaltando características como: bordes, texturas, cambios de luz, nitidez, las cuales van caracterizando lo imagen. La salida de este proceso está dada por  $k$  mapas de características que es el resultado de la aplicación de las máscaras de convolución (filtros). Estos mapas de características pasan por una función de activación llamada ReLu (del inglés, rectified linear unit) la cual está definida por (Manzano, 2015):

$$f(x) = \max(0, x) \quad (4.30)$$

Esta función toma del mapa de características los valores mayores que cero y los menores los convierte en cero. Con los valores obtenidos en esta etapa los datos pasan por un proceso de sub-muestreo a lo largo de la dimensión espacial de la imagen, reduciendo el tamaño de la imagen. Las operaciones que se realizan son de max-pooling (valor máximo del sub-muestreo) o mean-pooling (valor promedio del sub-muestreo), utilizando una ventana de tamaño  $p \times p$ , donde generalmente  $p = 2$  para imágenes pequeñas y no mayor a 5 para imágenes de mayor tamaño, el resultado para la primera operación sería una imagen reducida con los valores máximos mapeados por la ventana en el área de características de

la imagen original y para la segunda operación el promedio de los valores mapeados de igual manera. Para la etapa de clasificación generalmente se usan redes neuronales como la Backpropagation, la RBF y otros clasificadores como las SVM por su gran desempeño en los procesos de clasificación (Liberda et al., 2016).

Durante el entrenamiento de una red neuronal supervisada, se dividen los datos en dos grupos: entrenamiento y validación. Los datos con los cuales se entrena la red son analizados  $n$  veces por el algoritmo de la arquitectura diseñada de clasificación, hasta que alcance el desempeño deseado ante el conjunto de datos de validación. El desempeño de la red se determina hallando el porcentaje de los datos que logró clasificar correctamente con respecto a todo el conjunto de datos.

Para alcanzar una mayor precisión en la clasificación, la CNN requiere una gran cantidad de ejemplos de imágenes, si el número de imágenes es limitado se recomienda realizar técnicas que ayuden a construir datos sintéticos mediante transformaciones como obtener una imagen nueva que represente un “espejo” en diferentes posiciones (vertical y horizontales) de la imagen original, rotarla, variaciones de color y del nivel de zoom, entre otras que permitan al modelo aprender de un mayor número de variaciones dentro de las clases (Weimer et al., 2016).

## 5. METODOLOGÍA.

Para el desarrollo de la investigación es preciso diseñar una metodología que permita a partir de una serie de pasos lógicos determinar los procedimientos necesarios para recolectar, analizar y clasificar los datos, con el objetivo de alcanzar los resultados esperados en la hipótesis del método propuesto de inspección. La figura 5.1 muestra el esquema propuesto para la metodología.

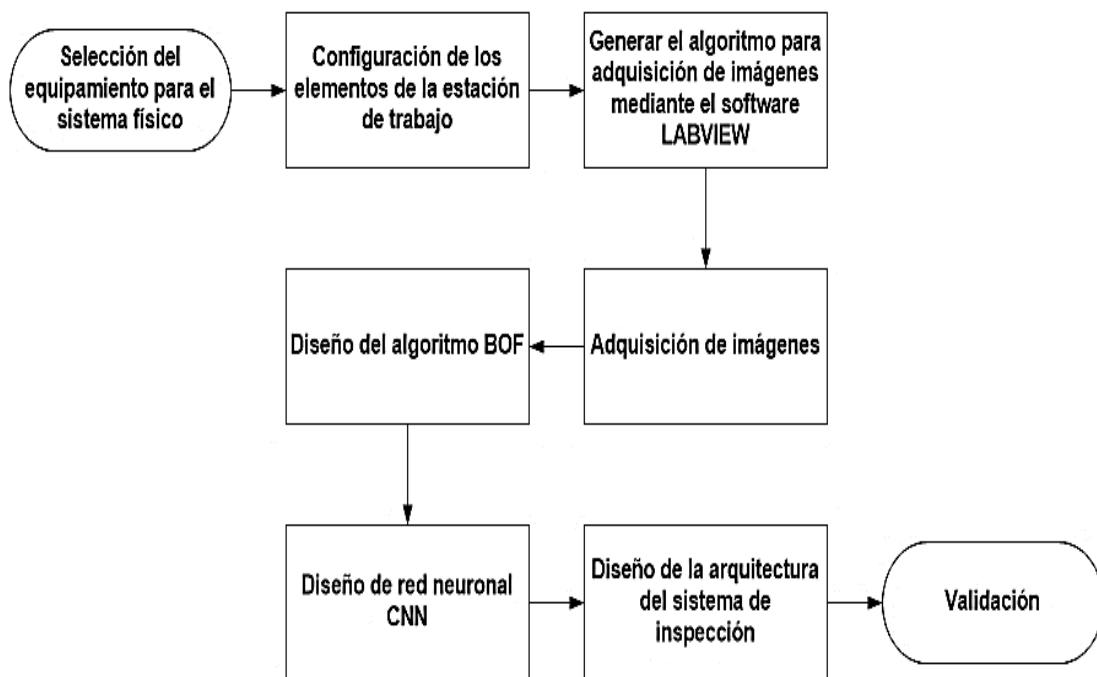


Figura 5. 1 Metodología. Elaboración propia.

### 5.1. Selección del equipamiento para el sistema físico.

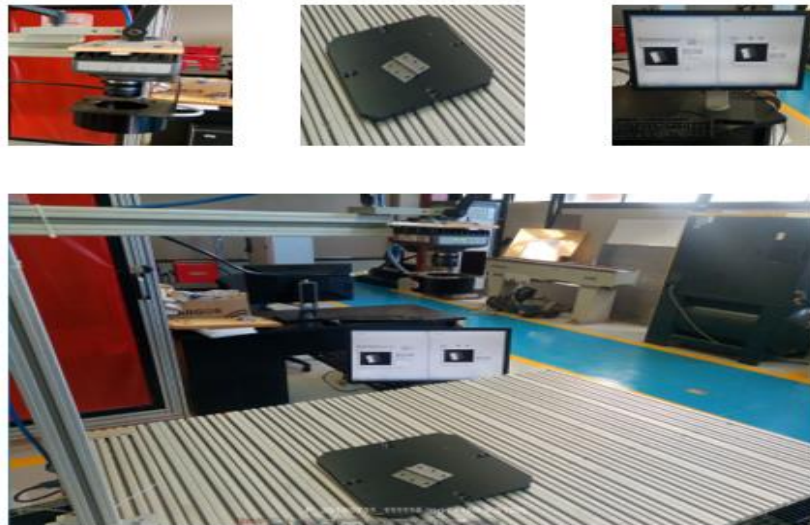
Para llevar a cabo el proyecto se emplearon una serie de equipos y software que permitieron realizar los experimentos necesarios para el desarrollo del proyecto de inspección de piezas manufacturadas. La toma de las imágenes se realiza con una cámara NI 1742 Smart Camera de la National Instrument, con sensor de imagen monocromático

640 x 480. Conectada a una PC con procesador Xeon (R) a 2.39Gz, memoria RAM (6Gb), tarjeta gráfica (NVIDIA Quadro FX 580) y sistema operativo Windows 7 Profesional, el protocolo de comunicación usado fue Ethernet/IP. El software de LabView se empleó para la interfaz entre cámara-PC y con el software de Matlab se realizó la programación de procesamiento de la imagen.

### *5.2. Configuración adecuada para los elementos del sistema. (Estación de trabajo)*

Se determinará la configuración adecuada de cada elemento del sistema: posición y ángulo de la cámara, la ubicación del sistema de luz y la computadora empleada, cables de conexión, estructura necesaria de la mesa para la colocación de los elementos como se muestra en la figura 5.2.

#### **Infraestructura**



*Figura 5. 2 Infraestructura de la estación de captura y procesamiento de la imagen. Elaboración propia.*

### *5.3. Generar algoritmo para adquisición de imágenes mediante el software LABVIEW.*

Desarrollar la programación necesaria para poder adquirir las imágenes a través del software del LabView y enviar los datos al software del Matlab para el procesamiento de la imagen utilizando la arquitectura diseñada.

#### *5.4. Adquisición de imágenes.*

En el proceso de adquisición de imágenes es necesario obtener una mayor variabilidad en las posiciones de captura de la imagen del producto, debido a que las piezas que llegan a la estación de inspección a través de las bandas transportadoras, llegan en posiciones aleatorias. Además, analizar cómo afecta la luz en los diferentes materiales del producto para lograr una base de datos completa que permita un sistema robusto ante cambios en el medio de producción.

#### *5.5. Diseño del algoritmo BOF.*

Para identificar el tipo de piezas es necesario un método de procesamiento de imágenes que sea capaz de reconocer un objeto a partir de su contorno o forma. Uno de los algoritmos que se caracteriza por su método simple en la obtención de vectores descriptores de contornos es el algoritmo BOF, el cual se implementa en la investigación propuesta para determinar los vectores que describen las piezas utilizadas. El vector BOF está compuesto por las diferentes distancias euclidianas que existen entre el centroide del objeto representado en una imagen 2D y diferentes puntos determinados del contorno. Para obtener el BOF se realiza un preprocesamiento que permita determinar el área que ocupa la pieza en la imagen capturada y con esta información se determinan los diferentes puntos del contorno. Con el área se calcula el centroide del objeto y se determinan las distancias entre el centroide y los puntos del contorno para conformar el vector.



### 5.6. Diseño de red neuronal CNN.

La red neuronal convolucional es un método de extracción de características que ha obtenido resultados favorables para el reconocimiento de imágenes. En la investigación se utiliza para identificar el tipo de pieza y comparar su desempeño con el algoritmo BOF con respecto al tamaño del vector descriptor y el tiempo de procesamiento logrado con cada método para la clasificación de las piezas por su contorno. En la etapa de detección se utiliza otra CNN para identificar los diferentes defectos de las piezas. Para conformar la arquitectura de la CNN es necesario determinar los filtros de convolución adecuados para resaltar las características deseadas del objeto, determinar la etapa de sub-muestreo que no permita la pérdida de la información extraída en la operación de convolución y el método de clasificación capaz de identificar los patrones obtenidos en la etapa de extracción de características. La figura 5.7 muestra la arquitectura de la CNN.

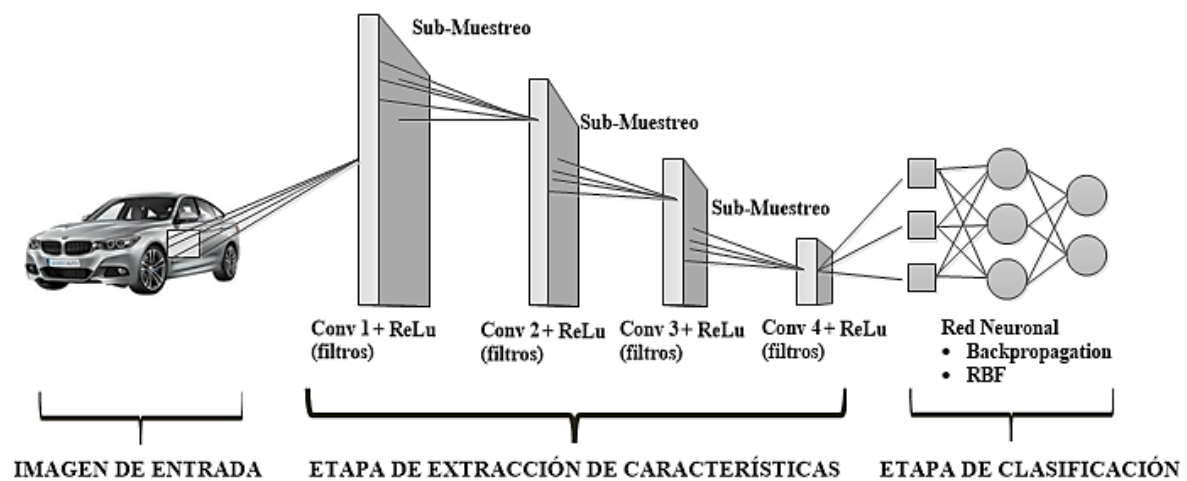


Figura 5. 3 Arquitectura de la CNN. Elaboración propia.

### 5.7. Diseño de la arquitectura del sistema de inspección.

El diseño de la arquitectura del sistema de inspección propuesto se determinará a partir de los resultados obtenidos en las etapas de experimentación, identificación de piezas y detección de defectos. El sistema de inspección se divide en dos etapas, la primera en la identificación del tipo de piezas y la segunda en la detección de defectos de las piezas. Las técnicas empleadas para el reconocimiento de piezas se determinarán partiendo del resultado de la comparación entre el algoritmo BOF y la CNN analizando el tamaño del vector descriptor obtenido por cada método y el tiempo de procesamiento utilizado por cada uno. La detección de defectos se realizará con la CNN por su capacidad de extraer características en las imágenes y se realizará un análisis de las diferentes técnicas de clasificación para clasificar los patrones obtenidos. Se utilizarán para la clasificación redes neuronales como la Backpropagation y la RBF, y las máquinas de soporte vectorial en los casos donde los datos que conforman los vectores descriptores tengan un alto grado de solapamiento entre clases.

#### *5.8. Validación.*

Se analizarán los resultados de la experimentación de cada una de las técnicas propuestas y se realizará un proceso de inspección utilizando la arquitectura final del sistema propuesto con la base de datos de validación, la cual contiene imágenes diferentes a las que fueron entrenadas cada una de las técnicas.

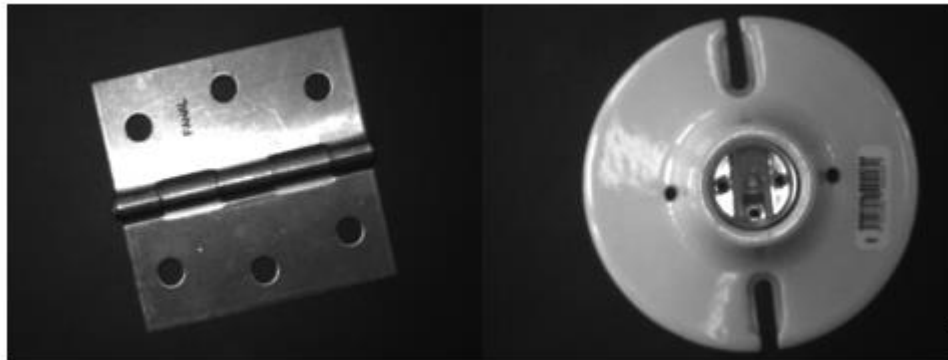
## **6. EXPERIMENTACIÓN Y RESULTADOS**

En el capítulo se presenta el proceso de experimentación y resultados donde se exponen las diferentes técnicas utilizadas para desarrollar el sistema de inspección propuesto, el cual se dividió en dos etapas de experimentación. La primera se centró en la búsqueda de técnicas adecuadas para la identificación de piezas donde se utilizó el algoritmo BOF y la CNN para determinar los vectores descriptores de contornos de las piezas. En la clasificación de los vectores descriptores del BOF se utilizó la Backpropagation y en la clasificación de los vectores descriptores de la CNN se utilizaron las redes Backpropagation y RBF propias de la arquitectura de la CNN. Dependiendo de los resultados obtenidos con cada método considerando el tamaño del vector y el tiempo de procesamiento, se determinan las técnicas que conforman la etapa de identificación del tipo pieza. Para la segunda etapa de detección de defectos, teniendo en cuenta el comportamiento de las técnicas utilizadas en la primera etapa, se propuso la CNN para determinar los vectores descriptores de los defectos de cada pieza debido a su método de extracción de características en la imagen y el clasificador SVM para lograr clasificar los vectores de la CNN que presentaban un alto grado de solapamiento en los datos que los componen.

### *6.1. Adquisición de imágenes*

Para conformar la base de datos a partir de la adquisición de imágenes, fueron asignadas por la Corporación, dos piezas diferenciables tanto en formas como en material, debido a que permite en el proceso de extracción de características enfrentarse a diferentes fenómenos ocasionados por el reflejo de la luz en diferentes materiales y por tanto a un comportamiento variable en el análisis de las intensidades de los píxeles (valor del píxel en

la imagen) ante un proceso de filtrado. En la figura 6.1 se muestran las piezas empleadas en la experimentación, con las cuales se creó la base de datos ubicándolas en diferentes posiciones del área de captura y con diferentes defectos de fabricación, además de aplicarles diferentes transformaciones de rotación y mirror (del inglés: espejo) para obtener una cantidad aceptable de datos de entrenamiento, pertenecientes a cada clase.



a) Bisagras

b) Soporte de lámparas

*Figura 6. 1 Piezas de entrenamiento. Bisagras y Soportes de lámparas.*

Se creó una primera base de datos que contiene las diferentes imágenes de los tipos de piezas de tamaño (640x480 píxeles) y se dividió en dos categorías. La clase con etiqueta (-1) representa las bisagras metálicas y la etiqueta (+1) representa la clase de Soporte de lámparas de cerámica, como se muestra en la Tabla 6.1. Esta base será analizada en la primera etapa del sistema de inspección, el cual conlleva a identificación de piezas.

*Tabla 6. 1 Equivalencia entre las categorías de las etiquetas y las piezas a analizar.*

<b>Categorías</b>	<b>Tipo de piezas</b>
-1	Bisagras
+1	Soportes de lámparas

La segunda base de datos contiene las diferentes imágenes de tamaño (640x480 píxeles), etiquetadas por sus defectos a partir del tipo de pieza. En la Tabla 6.2 se representan algunas de las categorías de los defectos etiquetados. Esta base de datos será utilizada en la segunda etapa del sistema de inspección, el cual está enfocado en detectar los defectos en las piezas.

Tabla 6. 2 Equivalencia entre las categorías de las etiquetas y los defectos de las piezas a analizar.

<b>Categorías</b>	<b>Defectos Bisagras</b>	<b>Categorías</b>	<b>Defectos Soporte de lámparas</b>
1	Orificio dañado	5	Sin sello
2	Perno dañado	6	Bornes dañados
3	Sellos dañados	7	Fisuras
4	Sin defecto	8	Sin defecto

En la figura 6.2 se representan algunas muestras de la base de datos que contiene los defectos a analizar en las piezas. Las imágenes están etiquetadas según el tipo de defecto contenido en el objeto. Se pueden observar las diferentes posiciones en las imágenes para obtener variedad en los datos. Con las bases de datos conformadas se da paso a la siguiente etapa de experimentación, donde se determinan los métodos para la identificación de piezas, utilizando el BOF y la CNN para determinar los vectores descriptores de contornos de cada pieza y las redes Backpropagation y Función de Base Radial para la clasificación de los vectores obtenidos.

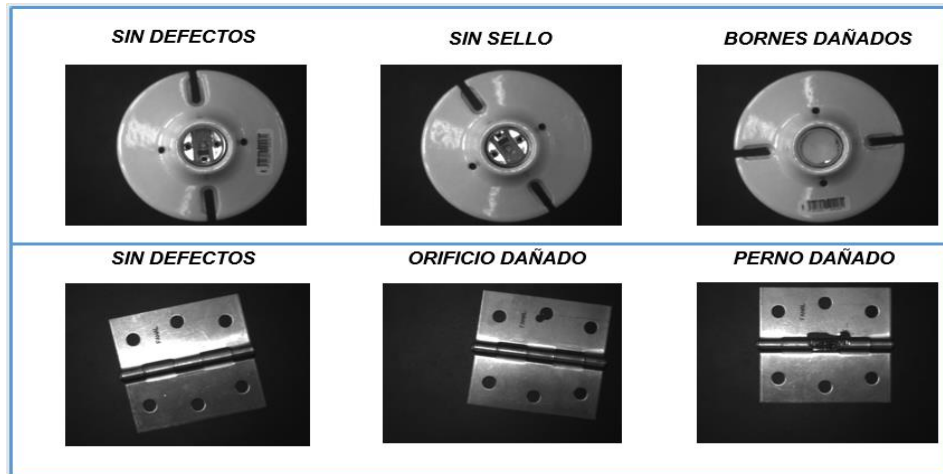


Figura 6. 2 Defectos etiquetados.

### 6.2. Etapa de identificación del tipo de pieza.

Se diseñan las diferentes técnicas para identificar el tipo de pieza analizando el tamaño del vector descriptor obtenido por cada técnica y el tiempo de procesamiento de cada método, con la finalidad de seleccionar la que presente un desempeño superior. Lo anterior se expone en la figura 6.3.

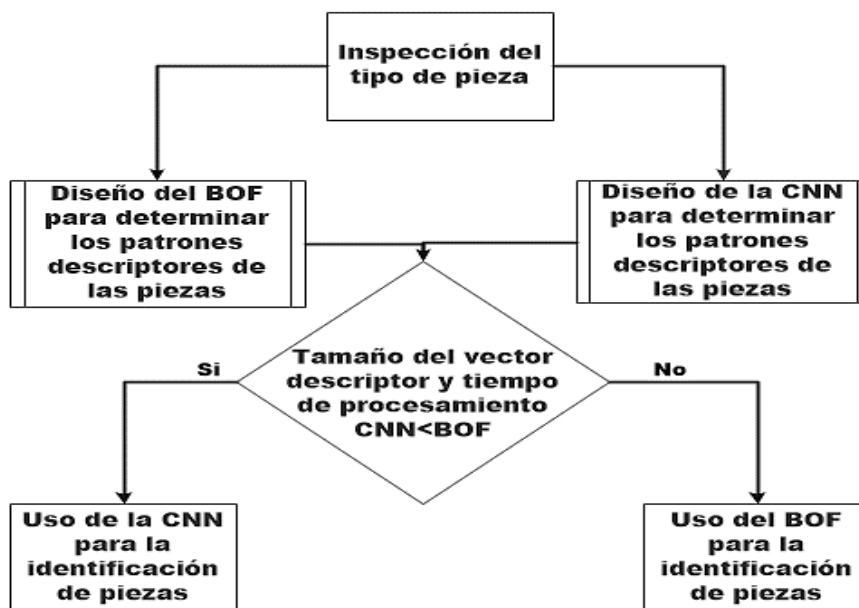


Figura 6. 3 Pasos lógicos para la identificación del tipo de piezas.

### 6.2.1. *Implementación del algoritmo BOF.*

El primer paso para obtener el BOF, es la detección de bordes de la imagen al realizar una segmentación entre el objeto y el fondo. Para ello inicialmente se aplicó la técnica del umbral para diferenciar el objeto del fondo y así obtener el contorno de la pieza, se determinó el valor del píxel que dividía la frontera entre la pieza y el fondo (umbral) y se transformaron los píxeles menores al umbral en negros y los mayores en blancos. Debido a que las imágenes fueron tomadas en diferentes posiciones, la luz provocó variación en las intensidades de los píxeles ocasionando que el valor del umbral no fuera el mismo para cada pieza, lo cual dio como resultado imágenes con piezas incompletas. Este problema se detectó con las bisagras que presentan el mismo material, por lo que al hacer el análisis con los Soportes de lámparas que poseen otro tipo de material también se tendría que variar el umbral, demostrando que la técnica del umbral en este caso no permite un mismo código de programación para procesar todas las imágenes de la base de datos tomada.

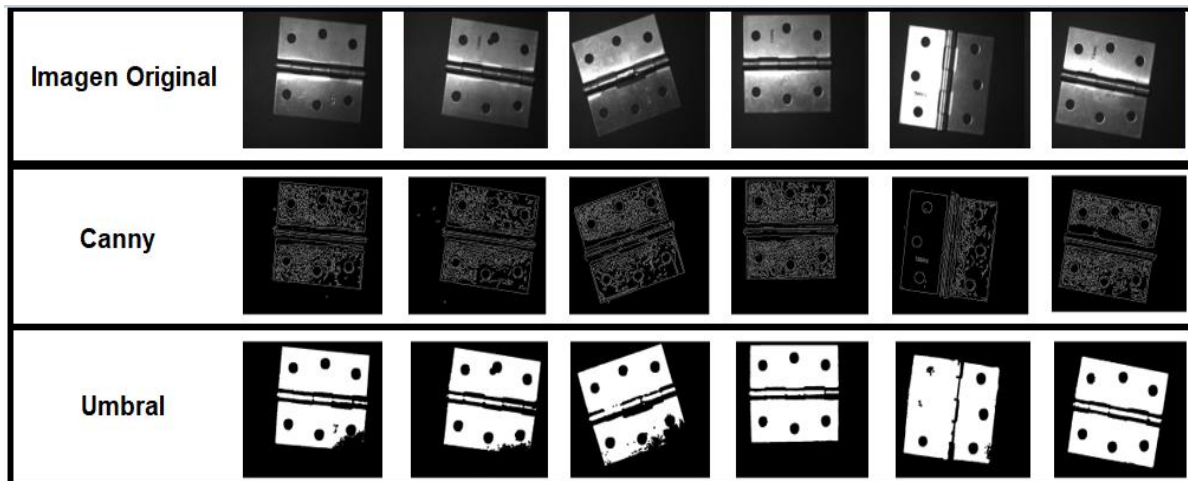
Para la primera etapa del BOF, se utilizó el algoritmo de Canny para detectar los bordes existentes en la imagen. Este algoritmo realiza la detección de contornos mediante la utilización de máscaras de convolución y basado en la primera derivada. Los puntos de contorno son zonas de píxeles en las que existe un cambio representativo en los niveles de gris. En el tratamiento de imágenes se trabaja con píxeles, es así que en el algoritmo de Canny se utilizan máscaras, las cuales representan aproximaciones en diferencias finitas.

El algoritmo de Canny consiste en tres pasos:

- Obtención del gradiente: en este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.

- **Supresión no máxima:** El objetivo de este paso es obtener bordes de un píxel de grosor, al considerar únicamente como píxel aquellos cuya magnitud es máxima en bordes gruesos y descartar aquellos cuyas magnitudes no alcancen ese máximo.
- **Histéresis de umbral:** en este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

En la figura 6.4 se muestra una comparación entre los resultados obtenidos por la técnica del umbral y el algoritmo de Canny, en la primera fila se representan las imágenes tomadas por la cámara, en la segunda y la tercera fila el resultado a las técnicas aplicadas. Como se puede observar el algoritmo de Canny obtuvo mejores resultados en la búsqueda del contorno independientemente de los ruidos ocasionados por la intensidad de la luz.



*Figura 6. 4 Comparación entre las técnicas de umbral y algoritmo de Canny aplicadas.*

Con la información del contorno del objeto se realiza un procedimiento para buscar su área y poder determinar el centroide, el cual es uno de los parámetros fundamentales del BOF.



El área es la sumatoria de todos los píxeles que pertenecen a la forma del objeto, por lo que es necesario lograr que todas las piezas de la base de datos tengan el mismo valor de intensidad de píxel que representen un mismo color, en este caso blancos y diferenciables al fondo que en este caso sería negro, para lograr encontrar la suma de todos los píxeles semejantes se utiliza la ecuación (4.2).

Para determinar el área se toma la imagen con el borde resaltado y se hace un barrido píxel a píxel de izquierda a derecha desde la primera hasta la última fila, cambiando el valor de intensidad del píxel a 255 (blancos) en el área entre el primer píxel blanco encontrado y el último definido por el contorno correspondiente a cada fila de la matriz de la imagen. De igual manera se vuelve a recorrer la imagen de arriba hacia abajo recorriendo desde la primera columna hasta la última cambiando el valor de intensidad del píxel a 255 (blancos) en el área entre el primer píxel blanco encontrado y el último definido por el contorno correspondiente a cada columna. En la figura 6.5 se presenta una muestra del resultado del procedimiento llevado a cabo en la descripción para determinar el área.



*Figura 6. 5 Resultado del procedimiento para determinar el área.*

El vector del BOF está formado por las distancias entre el centroide y cada uno de los puntos del perímetro del objeto, por lo que es necesario determinar las coordenadas del centro de cada uno de los objetos en la imagen.

Las coordenadas del centroide están definidas por la sumatoria de todas las posiciones en la matriz de píxeles que contienen al objeto con respecto a los dos ejes de coordenadas, dividido entre el área del objeto, expresado con la ecuación (4.4).

Los puntos del perímetro necesarios para calcular las distancias que conforman el BOF se determinaron a partir de la ubicación del objeto en un eje de coordenadas con su origen en el centroide, realizando en cada cuadrante un barrido cada  $5^\circ$  para encontrar el último punto en la recta formada por cada ángulo que pertenece al objeto.

Para ello se utilizó la ecuación de la recta:

$$y = mx + b \quad (6.1)$$

Donde la pendiente está dada por el ángulo:

$$m = \arctan(\theta) \quad (6.2)$$

Se evaluaron las coordenadas correspondientes con  $x$  representando el valor que le corresponde en el eje de coordenadas en píxeles, aumentando en 1 su valor, partiendo del centroide y determinando el valor de la coordenada correspondiente a  $y$ . Cada valor de  $(x, y)$  perteneciente a la recta se comprueba si forma parte del objeto, es decir, si el valor del píxel en la posición  $(x, y)$  es igual a 255 (blanco). En la figura 6.6 se representa el centroide del objeto y dos rectas con diferentes ángulos con los puntos finales de cada recta pertenecientes a la pieza,  $(x_1, y_1)$  recta con  $\theta = 5$  y  $(x_2, y_2)$  recta  $\theta = 10$ .

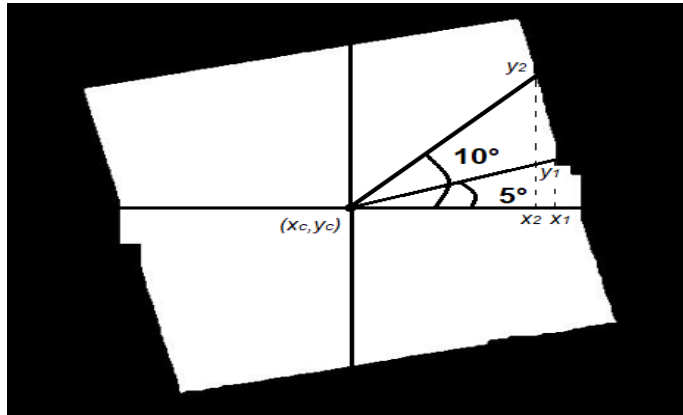


Figura 6. 6 Puntos del perímetro del objeto.

Con el centroide y los puntos del perímetro se comienza a construir el vector del BOF determinando la distancia euclidiana entre el centro y los puntos, ecuación (4.5).

Se tomaron 68 puntos del perímetro para cada objeto analizado y se calcularon las distancias, creando los patrones que describen las bisagras y los Soporte de lámparas. Se muestran los patrones gráficos obtenidos por el BOF para bisagra y soporte de lámparas, figura 6.7.

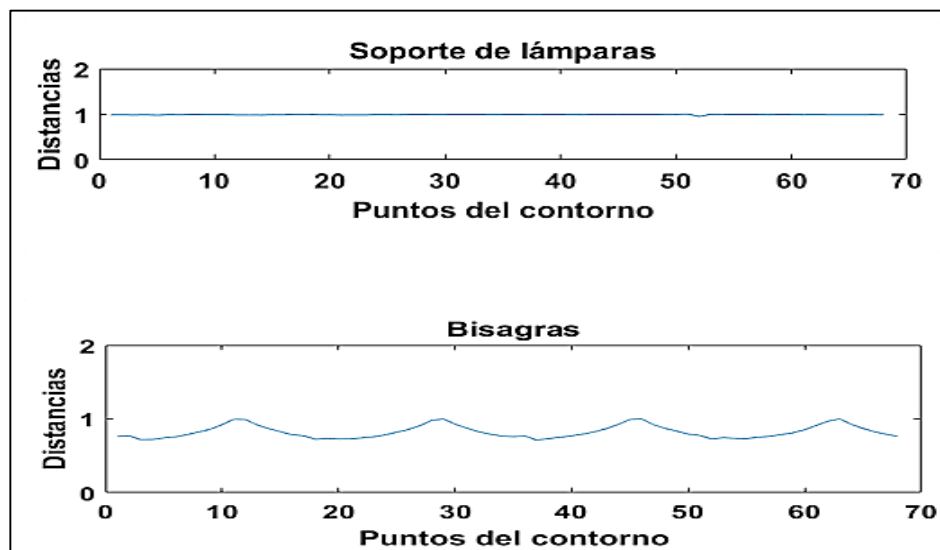


Figura 6. 7 Patrones descriptores de la bisagra y el Soporte de lámparas.

El descriptor obtenido con el algoritmo del BOF será la entrada de una red neuronal artificial, la cual se entrenó para identificar el patrón característico de cada tipo de pieza. A medida que el vector de entradas sea de menor tamaño, el entrenamiento de la red y su clasificación demandarán menos recursos de procesamiento de datos y tiempo. Como se mencionaba anteriormente, el vector obtenido del BOF es de dimensión  $1 \times 68$ , con el objetivo de lograr un tamaño menor sin perder las características del vector se realizó un sub-muestreo del vector. Se aplicó la operación del mean-pooling obteniendo el valor promedio como resumen de características de la ventana de muestra de tamaño  $1 \times 2$  con la que se hizo el mapeo del vector y la operación de max-pooling para obtener el valor máximo. Se concluyó utilizar el max-pooling debido a que los resultados mantenían una mejor descripción del patrón realizando en mayor grado las características descriptivas que se deseaban mantener al aplicar una operación de sub-muestreo. Se obtuvo un vector de tamaño  $1 \times 9$  como se muestra en la figura 6.8.

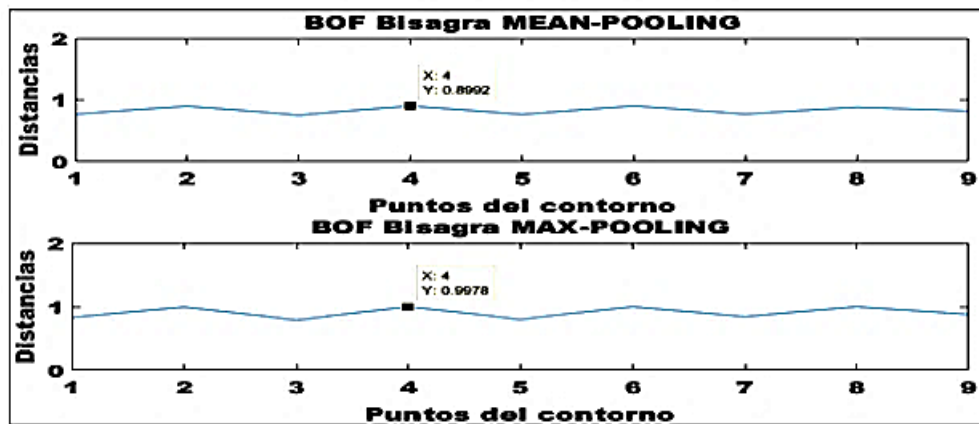


Figura 6. 8 Comparación entre la operación max-pooling y la operación mean-pooling.

### 6.2.2. *Implementación de la CNN para la identificación de piezas.*

Se diseñaron dos configuraciones de CNN para la inspección del producto, un primer diseño para determinar el tipo de pieza (bisagras o soporte de lámparas), con el objetivo de comparar el tiempo de procesamiento de la CNN y el procesamiento del BOF en la obtención del vector descriptor y los tamaños del mismo. El segundo diseño se implementó para el reconocimiento de defectos de los productos, debido al potencial que presenta esta arquitectura en el reconocimiento de imágenes.

Para el desarrollo de la etapa de identificación del tipo de pieza, se realizó un estudio de los diferentes filtros o kernels más utilizados en la etapa de convolución para resaltar las características deseadas del objeto. El objetivo al aplicar los diferentes filtros fue mantener las características representativas de la pieza durante el proceso de reducción del tamaño de la imagen. La reducción de la imagen es necesaria debido a que la resolución de las imágenes a inspeccionar es de 640x480, equivalente a un vector de entradas para la clasificación por medio de una RNA, de 307200 patrones por imágenes, lo que representa un mayor tiempo de procesamiento. Las características representativas de las piezas que se deseaban resaltar son el borde de la pieza y el de los elementos internos que la componen. El realce de bordes muestra en la imagen los detalles de fronteras o contornos, los cuales se utilizan en operaciones de análisis de imágenes para el reconocimiento de objetos o rasgos.

En el proceso de extracción de características se emplearon diferentes etapas de convolución aplicando una variedad de filtros que resaltarán los rasgos distintivos de cada pieza y sus defectos en la imagen. En la figura 6.9 se exponen 3 ejemplos que forman parte del proceso de búsqueda de los filtros adecuados para representar los patrones descriptores de cada una de las clases a clasificar en tipo de pieza y defectos. Cada ejemplo representa

la imagen original y las imágenes resultantes al aplicar cada filtro. En el ejemplo 1 la primera máscara está compuesta por un filtro identidad  $[0\ 0\ 0; 0\ 0.2\ 0; 0\ 0\ 0]$ , con el objetivo de disminuir los valores de intensidad en la imagen y contrarrestar los efectos de luz. Luego se aplica un filtro de borde laplaciano, formado por la máscara  $[-1\ -1\ -1; -1\ 8\ -1; -1\ -1\ -1]$ , el cual resalta los bordes de la pieza. Además, se aplica un filtro de identidad  $[0\ 0\ 0; 0\ 0.5\ 0; 0\ 0\ 0]$ , debido a que la intensidad de los píxeles de la imagen resultante fue muy baja y con esta operación se realzarían los colores claros. Posteriormente se mapea la imagen con el mismo filtro de borde laplaciano y se obtiene una imagen con los contornos resaltados, pero con exceso de ruido, por eso se aplica un filtro gaussiano  $[1\ 2\ 1; 2\ 4\ 2; 1\ 2\ 1] \cdot (1/16)$  para atenuar ese ruido. Los filtros aplicados en este ejemplo no representaron las características deseadas en las piezas y se realizaron nuevas combinaciones de máscaras convolucionales. En el ejemplo 2 de la imagen se aplicó a la imagen original un filtro paso bajo, también llamado filtro de medias, para atenuar las altas frecuencias evitando resaltar el ruido al aplicar los filtros de bordes. Además, se aplicó el gradiente de Prewitt obteniendo los contornos que definían el defecto de la pieza, posteriormente un filtro de gauss para atenuar el ruido y una máscara laplaciana para definir los bordes de cada elemento de la pieza. Se observó una mejora considerable entre el procesamiento del ejemplo 1 y el ejemplo 2. Con la investigación se comprobó que el filtro de Sobel, que posee características similares al de Prewitt, provocaba un mayor resalte de bordes en la pieza, lo cual definía en mayor grado las clases etiquetadas en la etapa de clasificación. En el ejemplo 3 se aplicó un procesamiento a la imagen equivalente al ejemplo 2, sustituyendo el gradiente de Prewitt por el filtro de Sobel. En la imagen se observa como la combinación de los filtros adecuados representan las características deseadas que permiten identificar el tipo de pieza y los diferentes defectos en las piezas.

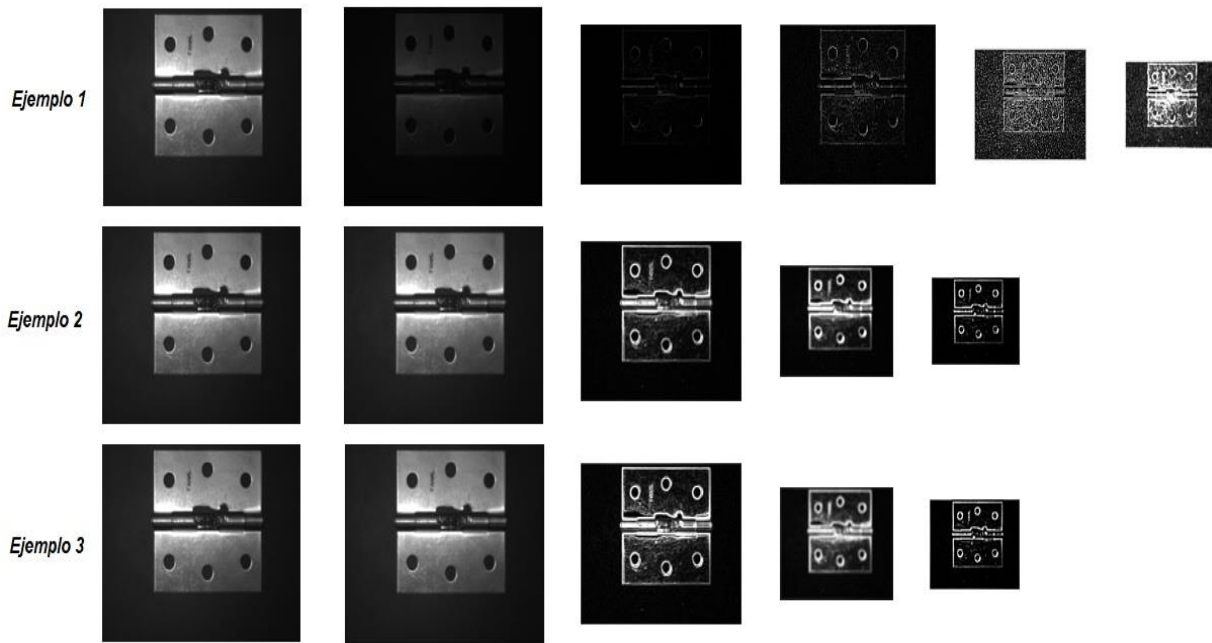


Figura 6. 9 Ejemplos de procesamiento de imágenes aplicando diferentes filtros en la etapa de convolución.

Posteriormente de una etapa de convolución y de ReLu (función rectificadora), se realiza un sub-muestreo lo cual al usar tanto la operación de max-pooling como la de mean-pooling, existe una reducción de la imagen, lo que puede provocar pequeños cambios en la imagen con respecto a la información que posee y variaciones en las intensidades de los píxeles. En la arquitectura propuesta se utilizaron los filtros de medias, de Sobel, de Gauss y Laplacianos con el objetivo de lograr extraer las características descriptivas de las piezas. Este proceso se utilizó en las dos etapas de experimentación, para identificar el tipo de piezas y en la segunda etapa para detectar los defectos.

En el diseño de la CNN utilizado para la identificación del tipo de piezas, las deformaciones de bordes en los elementos internos del producto no eran representativas, lo que permitió agregar otras etapas de convolución y sub-muestreo a la arquitectura definiendo el contorno exterior de cada pieza. En la etapa de extracción de características

de la red CNN, se obtuvo un vector descriptor de tamaño  $1 \times 1200$  capaz de representar la diferencia entre piezas.

El vector obtenido en la etapa de extracción de características de la CNN, es la entrada a la etapa de clasificación de la arquitectura, donde se utiliza una RNA completamente conectada para la clasificación de los vectores de la primera etapa de la red CNN. Inicialmente se analizó la representación de los datos de las diferentes clases en un sistema cartesiano para determinar el método de clasificación según el solapamiento de los datos. En la figura 6.10 se muestran dos vectores tomados de diferentes clases (bisagras y soportes de lámpara) representados en el gráfico. En el diagrama se puede observar que existe un alto grado de solapamiento entre los datos de ambas clases mostrando la poca diferenciación entre los patrones que identifican el tipo pieza, debido a este comportamiento de los datos es necesario un estudio detallado de la RNA a utilizar en el reconocimiento de patrones, la cual debe ser capaz de separar en regiones las diferentes clases etiquetadas.

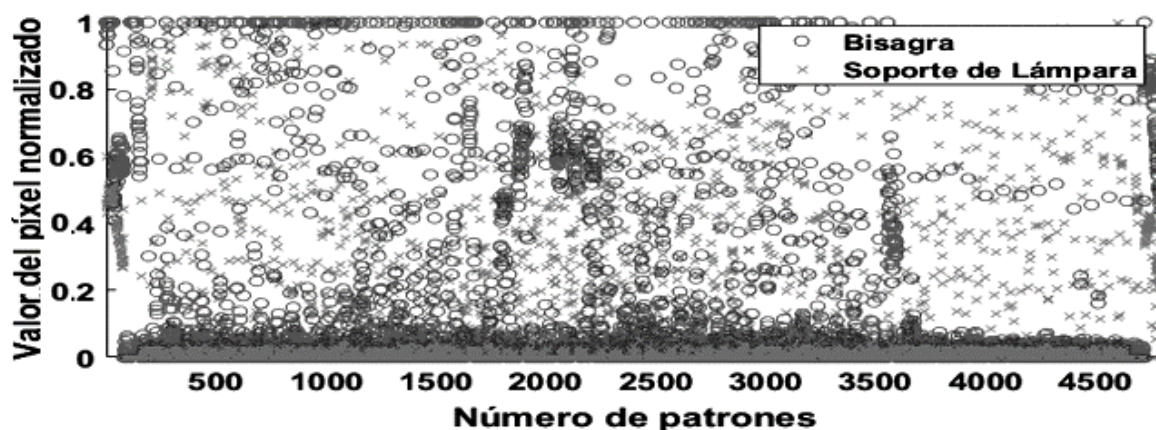


Figura 6. 10 Gráfico que representa un vector de las bisagras y un vector de los soportes de lámparas.

Con el análisis de los diferentes vectores descriptores obtenidos en la etapa de extracción de características de la CNN y los obtenidos con el algoritmo BOF desarrollado,



se comenzó a realizar comparaciones entre ambas técnicas para determinar la funcionalidad de las mismas dentro del sistema propuesto.

Inicialmente se evaluaron los tiempos de procesamiento empleado por cada técnica para determinar el vector descriptor del tipo de pieza, donde el algoritmo BOF realizó el procesamiento en 0.595s aproximadamente y la etapa de extracción de características de la CNN en 1.543s. Con respecto a los tamaños de los vectores descriptores obtenidos por cada técnica, el BOF logró obtener un vector descriptor de tamaño  $1 \times 9$  y el la CNN en su etapa de extracción de características obtuvo un vector descriptor de tamaño  $1 \times 1200$ . El algoritmo BOF demostró que necesita menor información para determinar los patrones que describen el tipo de pieza y menor tiempo de procesamiento para determinar esos patrones.

Para el reconocimiento de patrones que identifican las piezas con los vectores obtenidos por ambas técnicas, se utilizaron diferentes arquitecturas. Para los patrones del BOF se utilizó la Backpropagation y para los patrones obtenidos en la etapa de extracción de características de la CNN, se utilizaron las redes neuronales propias de su arquitectura, como la Backpropagation y la RBF para analizar el comportamiento de los datos en la clasificación. A continuación, se muestran los procesos de clasificación.

*6.2.3. Clasificación de los vectores descriptores obtenidos con el algoritmo BOF y la CNN.*

*6.2.3.1. Clasificación de los vectores descriptores del BOF con la red Backpropagation.*

Los vectores descriptores obtenidos con el algoritmo BOF se clasificaron con una red Backpropagation donde se eligieron 160 vectores descriptores de tamaño  $1 \times 9$ , 104

representaban las bisagras y 56 a los soportes de lámparas. Se inició con una arquitectura básica de una capa de entrada, una capa oculta y una capa de salida. En la figura 6.11 se muestra el esquema de la red implementada, la capa oculta está compuesta por 9 neuronas que corresponde al número de patrones de entrada y una neurona en la capa de salida.

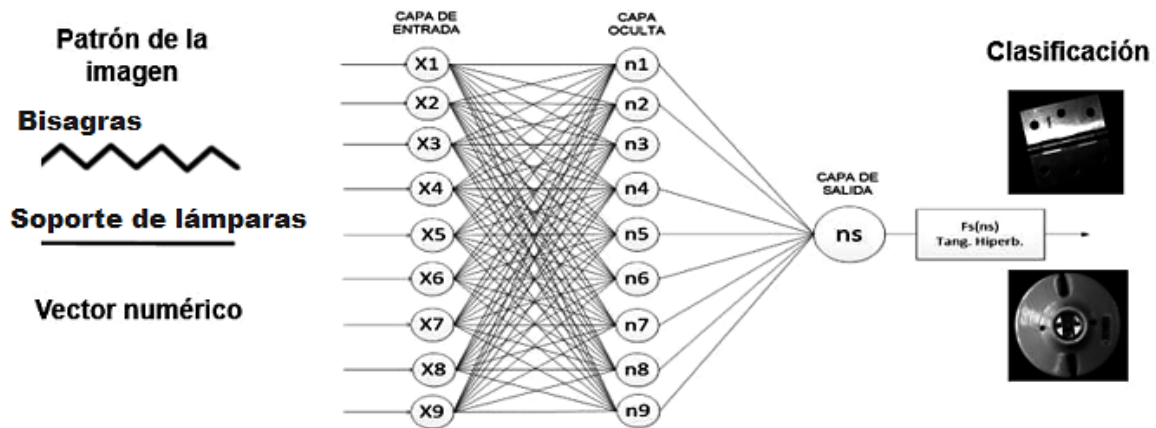


Figura 6. 11 Esquema de la red Backpropagation utilizada para la identificación de las piezas.

La función de activación que se utilizó en la capa oculta y en la de salida fue la tangente hiperbólica, ecuación (4.9), su salida está acotada entre  $[-1 \ 1]$ .

El algoritmo que se utilizó fue basado en el número de épocas, con las cuales se pretendía corregir las fallas de clasificación durante el entrenamiento dependiendo del error de salida. En el entrenamiento se modificó el número de épocas buscando en la respuesta un error menor de  $\pm 0.05$  como se muestra en la tabla 6.3. Los valores con la letra más oscura representan la no clasificación y las clasificaciones que tuvieron un error mayor de 0.05. El valor deseado de salida de las bisagras es (-1) y el de los soportes de lámparas (+1). Con la arquitectura de la red Backpropagation aplicada se logró clasificar el tipo de pieza en el proceso de inspección.

Tabla 6. 3 Clasificación de la red Backpropagation en la detección de piezas con los vectores del BOF.

Tipo de pieza	Clasificación de la red para cada número de épocas				
	10	100	200	300	400
Bisagras	<b>0.895</b>	<b>0.567</b>	<b>-0.934</b>	-0.999	-0.997
Bisagras	<b>0.901</b>	<b>0.739</b>	<b>0.444</b>	<b>-0.922</b>	-0.988
Bisagras	<b>0.902</b>	<b>0.697</b>	<b>-0.118</b>	-0.989	-0.996
Bisagras	<b>0.889</b>	<b>0.433</b>	<b>-0.944</b>	-0.999	-0.999
Bisagras	<b>0.888</b>	<b>0.397</b>	-0.954	-0.999	-1
Bisagras	<b>0.893</b>	<b>0.488</b>	<b>-0.934</b>	-0.999	-0.999
Bisagras	<b>0.89</b>	<b>0.451</b>	<b>-0.941</b>	-0.999	-0.999
Bisagras	<b>0.904</b>	<b>0.68</b>	<b>-0.822</b>	-0.997	-0.98
Bisagras	<b>0.897</b>	<b>0.736</b>	<b>0.209</b>	-0.974	-0.991
Bisagras	<b>0.889</b>	<b>0.44</b>	<b>-0.946</b>	-0.999	-0.999
Bisagras	<b>0.892</b>	<b>0.575</b>	<b>-0.751</b>	-0.998	-0.999
Bisagras	<b>0.89</b>	<b>0.383</b>	<b>-0.927</b>	-0.999	-0.999
Bisagras	<b>0.892</b>	<b>0.575</b>	<b>-0.895</b>	-0.999	-0.999
Bisagras	<b>0.889</b>	<b>0.438</b>	<b>-0.936</b>	-0.999	-0.999
Bisagras	<b>0.888</b>	<b>0.408</b>	<b>-0.949</b>	-0.999	-1
Soporte de lámparas	<b>0.927</b>	<b>0.924</b>	<b>0.927</b>	0.983	0.993
Soporte de lámparas	<b>0.927</b>	<b>0.924</b>	<b>0.928</b>	0.984	0.993
Soporte de lámparas	<b>0.926</b>	<b>0.918</b>	<b>0.914</b>	0.966	0.99
Soporte de lámparas	<b>0.926</b>	<b>0.923</b>	<b>0.926</b>	0.982	0.993
Soporte de lámparas	<b>0.926</b>	<b>0.924</b>	<b>0.928</b>	0.984	0.993
Soporte de lámparas	<b>0.927</b>	<b>0.924</b>	<b>0.927</b>	0.983	0.993
Soporte de lámparas	<b>0.926</b>	<b>0.923</b>	<b>0.925</b>	0.981	0.992
Soporte de lámparas	<b>0.927</b>	<b>0.924</b>	<b>0.926</b>	0.981	0.993
Soporte de lámparas	<b>0.926</b>	<b>0.923</b>	<b>0.926</b>	0.982	0.993
Soporte de lámparas	<b>0.926</b>	<b>0.923</b>	<b>0.924</b>	0.98	0.992
Soporte de lámparas	<b>0.926</b>	<b>0.922</b>	<b>0.924</b>	0.98	0.992
Soporte de lámparas	<b>0.925</b>	<b>0.917</b>	<b>0.913</b>	0.966	0.989
Soporte de lámparas	<b>0.927</b>	<b>0.923</b>	<b>0.925</b>	0.982	0.993

6.2.3.2. *Etapas de clasificación de la CNN: Patrones generados por la extracción de características de CNN y analizados con Backpropagation.*

Partiendo de los datos obtenidos de la CNN de tamaño  $1 \times 1200$  que representan el tipo de pieza, se realizó una primera configuración con una red Backpropagation. La arquitectura de la Backpropagation se fue modificando en base al error para obtener una mejor clasificación. Las modificaciones se realizaron en el número de capas, las funciones

de activación de las capas ocultas y las de salida, el número de neuronas de las capas ocultas y el número de épocas. La red no logró clasificar los datos, alcanzando solo el 50% de la clasificación de los datos de entrenamiento. A pesar de las variaciones que se realizaron en la arquitectura, se comprobó que, el valor del error de salida con respecto a la salida deseada alcanzaba un valor constante en las primeras épocas de cada entrenamiento realizado. En la tabla 6.4 se resumen algunas de las arquitecturas de la red Backpropagation donde se varió el número de capas ocultas, las funciones de activación de las mismas y las funciones de activación de la capa de salida, como resultado se muestra que la red Backpropagation no clasificó de manera aceptable los datos de las diferentes clases. Debido al nivel de solapamiento mencionado anteriormente, los errores de salidas se mantuvieron constante desde épocas tempranas.

Tabla 6. 4 Entrenamiento de la Backpropagation en la red CNN.

<b>Backpropagation</b>				
<b>Número de capas ocultas</b>	<b>Función de activación capa oculta</b>	<b>Función de activación capa de salida</b>	<b>Época donde el error es constante</b>	<b>% de Clasificación</b>
1	Tangente hiperbólica	Tangente hiperbólica	2	50%
2	Gaussiana Gaussiana	Gaussiana	No Clasificó	No Clasificó
2	Tangente hiperbólica Tangente hiperbólica	Gaussiana	2	50%
3	Tangente hiperbólica Sigmoidal Tangente Hiperbólica	Sigmoidal	1	No se obtuvo una clasificación en 10000 épocas

10	Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica Tangente hiperbólica	Tangente hiperbólica	7	50%
----	--	----------------------	---	-----

*6.2.3.3. Etapa de clasificación de la CNN: Patrones generados por la extracción de características de CNN y analizados con RBF.*

Otra de las redes neuronales artificiales que utiliza la CNN en su etapa de clasificación, es la RBF. El método de entrenamiento usado fue pseudo-inversa. Se calcularon los puntos medios entre los patrones que conforman los vectores correspondientes a las diferentes clases y las amplitudes que definen la campana de gauss de la función (distancia euclidiana entre cada patrón del vector de entradas y los puntos medios correspondientes a cada patrón). Con los puntos medios, las amplitudes y el vector de entradas se calculó la matriz que contiene las funciones gaussianas y se multiplicó la inversa de esa matriz por el vector de salida deseado obteniendo la matriz de pesos. La red obtuvo una clasificación de un 50% con los datos de entrenamiento.

Como resultado de las clasificaciones mencionadas anteriormente se puede observar que la Backpropagation logró la clasificación con los vectores del BOF, sin alcanzar los mismos resultados con los vectores de la CNN los cuales no lograron su clasificación del tipo de pieza con las dos redes utilizadas. A partir de este resultado se determina utilizar el algoritmo BOF para generar el patrón característico de cada pieza para la etapa de identificación del tipo de pieza, la etapa de extracción de características de la CNN para

determinar los vectores descriptores de los defectos de las piezas y las máquinas de soporte vectorial para la clasificación de los patrones que identifican los defectos debido a que es un clasificador potente donde se representarán los datos en una dimensión mayor a la analizada en el plano 2D y separar los datos a través de un hiperplano.

### *6.3. Etapa de detección de defectos de las piezas.*

#### *6.3.2. Implementación de la CNN para la detección de defectos.*

Para la determinar el vector descriptor de los defectos de las diferentes piezas, se utilizó la arquitectura de la CNN descrita en la etapa anterior ajustando el número de capas para obtener una mejor resolución de la imagen resultante y lograr un vector descriptor con la información necesaria para la clasificación de los defectos. Con la eliminación de operaciones de convolución y sub-muestreo en la etapa de extracción de características de la arquitectura CNN utilizada en la etapa de identificación de piezas, se obtuvo un vector descriptor de tamaño  $1x4800$  sin perder las características de bordes que identificaban los defectos en las piezas para su clasificación.

Para la determinar el método de clasificación de los vectores que representan los defectos de las piezas, se debe tener en cuenta los resultados mostrados en el gráfico de la figura 6.12. Como se observa existe un elevado grado de semejanzas entre los datos que conforman los vectores descriptores de los defectos de una misma pieza.

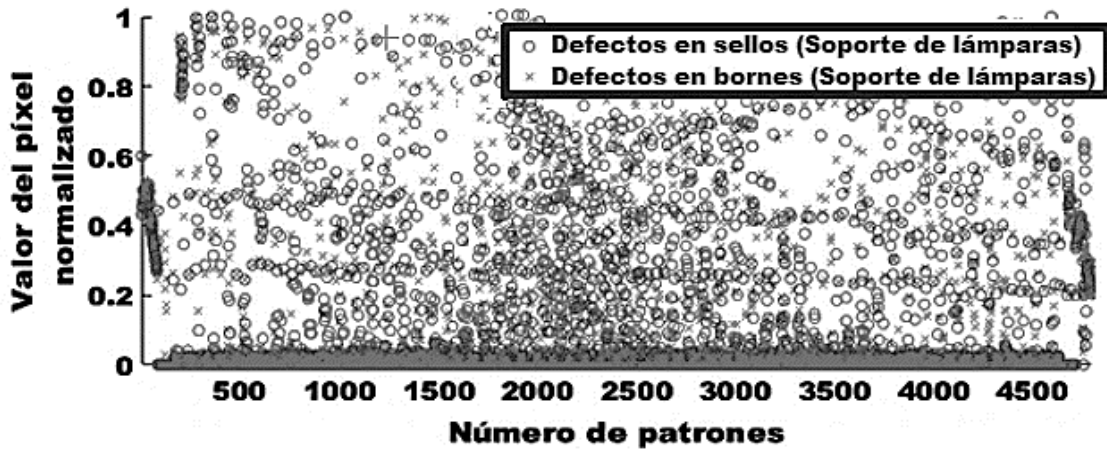


Figura 6. 12 Gráfico que representa dos vectores de defectos de los soportes de lámparas.

6.3.3. *Implementación de SVM en la arquitectura de la CNN para la detección de defectos.*

Los vectores que representan los defectos tienen un tamaño de  $1 \times 4800$ , para reducir el tiempo de procesamiento de la etapa de clasificación de SVM sin afectar el mapa de características que describe los defectos etiquetados obtenidos con la CNN, se diseñó un algoritmo que crea una nueva imagen de menor tamaño que la imagen capturada, enmarcando solo la pieza a analizar como se muestra en la figura 6.13 y logrando obtener vectores de  $1 \times 3249$ .

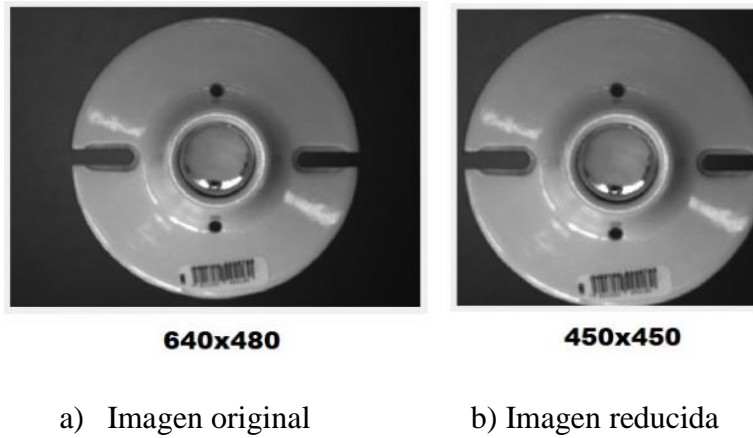


Figura 6. 13 Reducción de la imagen al área de interés.

La clasificación obtenida con SVM se obtuvo un valor de clasificación mayor al realizar la separación de los datos por medio de un hiperplano, representando los datos de su proyección en el espacio a un espacio de mayor dimensión. Para el desempeño de la SVM se emplearon dos procesos de entrenamiento, uno empleando el toolbox del Matlab (Classification Learner) y otro empleando un algoritmo basado en la búsqueda de los parámetros  $C$  y  $\sigma$  para encontrar el hiperplano que logre separar los datos.

Para ambos procesos se utilizó el método *uno contra todos* donde la clasificación se realizó con cada clase por separado, donde la clase a clasificar se le asignó el valor de 1 y el resto de las clases el valor de -1, el proceso se repitió con las 8 clases etiquetadas en la tabla 2 que representan los defectos de las piezas.

Uno de los métodos de clasificación de SVM es determinado los parámetros  $C$  y  $\sigma$ . La sintonización de estos parámetros en el proceso de entrenamiento se realizó a través de estrategias como la validación cruzada y la búsqueda de malla. La validación cruzada separa los datos en dos grupos, uno para entrenar y el otro para validar el entrenamiento. La búsqueda de los parámetros  $C$  y  $\sigma$  para lograr la clasificación deseada, se realiza variando los valores de los mismos y registrando los diferentes



porcientos de clasificación de cada clase para determinar el valor de los parámetros que proporcionó un mayor porcentaje de clasificación en el entrenamiento. Esta estrategia se conoce como búsqueda de malla. Un ejemplo de los resultados obtenidos con el método, fue con  $C= 30.47$  y  $\sigma= 8.539 e-05$  se obtuvo una precisión de 99% para la validación de la clase 1 de los defectos etiquetados (Bisagra con calidad adecuada).

En la figura 6.14 se muestra un ejemplo de la clasificación con el Toolbox. Como se puede observar en la figura, se utilizó en la clasificación con el Toolbox Classification Learner una función kernel cuadrático obteniendo una precisión de clasificación de 95.1%. En la matriz de confusión se observa el desempeño del algoritmo en cuanto a los porcentajes de clasificación de las clases entrenadas. Este proceso se le aplicó a cada clase y se seleccionaron las clasificaciones de mayor por ciento entre el Toolbox y el método de búsqueda de los parámetros  $C$  y  $\sigma$  para conformar el método de clasificación del proceso de reconocimiento de defectos de piezas.

**Matriz de Confusión**

<b>Clase Verdadera</b>	<b>-1</b>	<b>1</b>	<b>99%</b>	<b>1%</b>
	<b>1</b>	<b>23%</b>	<b>77%</b>	<b>23%</b>
	<b>-1</b>	<b>1</b>	<b>Tasa verdadera positiva</b>	<b>Tasa falsa negativa</b>
	<b>Clase Predicha</b>			

Figura 6. 14 Clasificación con el Toolbox del Matlab Classification Learner.

#### 6.4. Arquitectura del sistema de inspección propuesto.

Con los resultados obtenidos en la experimentación se determinó las técnicas que conforman cada una de las etapas del sistema de inspección, la identificación de piezas y la detección de defectos. En la identificación de piezas se utilizará el algoritmo BOF para extraer el vector característico que describe el contorno de los objetos y para la clasificación de los mismos se utilizará la red Backpropagation, con la que se logró clasificar a un 100%. Para la detección de defectos se determina el uso de la CNN para la extracción de características en la imagen y debido al grado de solapamiento de los datos se propone en la etapa de la CNN de clasificación utilizar las máquinas de soporte vectorial. La figura 6.15 muestra la arquitectura propuesta para el sistema de inspección.

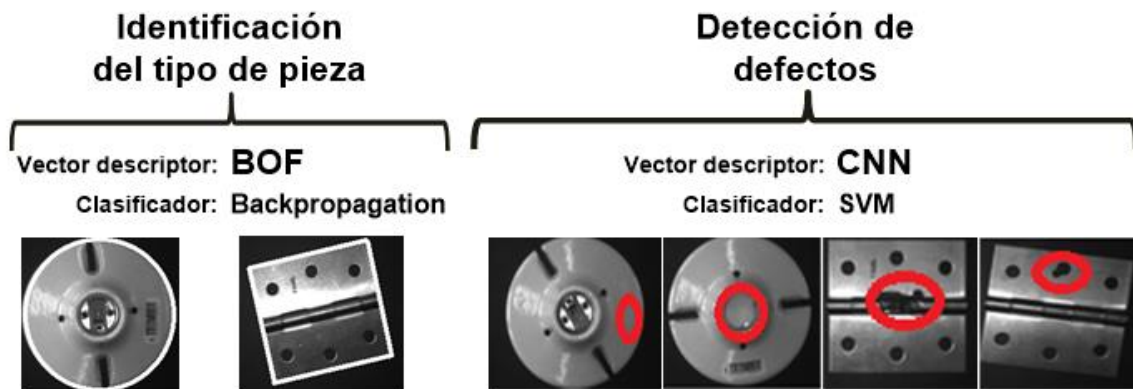


Figura 6. 15 Arquitectura del sistema de inspección propuesto.

#### 6.5. Validación del sistema.

Con la arquitectura propuesta para el sistema de inspección se inició la etapa de validación. Se creó la base de datos conformada por 108 imágenes distintas al conjunto de entrenamiento e incluían los 8 defectos etiquetados de los diferentes tipos de piezas. Se programó la validación con las imágenes en orden aleatorio de tamaño  $640 \times 480$  y se les

realizó el pre-procesamiento de datos para determinar el vector del BOF con su clasificación del tipo de piezas a través de la red Backpropagation y los vectores descriptores de los defectos con la CNN y su clasificación con la SVM. En la tabla 6.5 a continuación se muestran los resultados obtenidos del proceso de validación. La clasificación del tipo de piezas obtenida fue de 97.22% entre bisagras y soportes de lámparas. El procesamiento de la imagen con la red CNN para determinar los vectores descriptores de los defectos y la clasificación con SVM obtuvo un 83.33% de precisión entre los defectos de los dos tipos de piezas etiquetados en la base de datos de validación.

*Tabla 6. 5 Validación de las técnicas propuestas.*

<b>VALIDACIÓN</b>	
<b>Clasificaciones de las etapas de experimentación</b>	<b>Precisión</b>
<b>Tipo de pieza</b>	<b>97.22 %</b>
<b>Tipo de defectos</b>	<b>83.33 %</b>

## 7. CONCLUSIONES Y TRABAJO FUTURO.

### Conclusiones

- Se diseñó un sistema de identificación del tipo de piezas utilizando el algoritmo BOF para determinar el vector descriptor del contorno del objeto y se realizó la extracción del vector descriptor de contorno del objeto con la red CNN, donde el algoritmo BOF brindó un menor tiempo de procesamiento en la etapa de reconocimiento al generar vectores descriptores de tamaños  $1 \times 9$  en un tiempo de 0.595s, a diferencia de los obtenidos por la red CNN en el procesamiento de la imagen donde el vector descriptor fue de tamaño  $1 \times 1200$  en un tiempo de 1.543s. Esta comparación demuestra que el procedimiento del BOF es adecuado para la descripción de contorno que identifica los objetos, dando cumplimiento al objetivo específico número 2.
- Se logró reducir el tamaño de las imágenes analizadas de un tamaño de  $640 \times 480$  a  $450 \times 450$  enfocando el análisis de la imagen solo en la pieza, logrando reducir el vector descriptor de los defectos etiquetados de  $1 \times 4800$  a  $1 \times 3249$  disminuyendo el tiempo de procesamiento del entrenamiento de la SVM y el tiempo del sistema de inspección al clasificar una pieza en tipo y defecto. Se le da cumplimiento al objetivo específico número 3.
- Para la identificación del tipo de pieza, la red Backpropagation clasificó los vectores descriptores obtenidos con el algoritmo BOF, mientras que las redes empleadas en la etapa de clasificación de la CNN (Backpropagation y Función de Base Radial) no lograron una clasificación mayor del 50% debido al grado de solapamiento de los datos, lo que demuestra que para determinar los patrones que describen el contorno en la etapa de identificación de piezas la técnica adecuada es el algoritmo BOF el cual logró con menor cantidad de

información en sus vectores, determinar un vector descriptor del tipo de piezas y una clasificación de esos vectores con la red Backpropagation con una precisión de 97.22%. Se le da cumplimiento al objetivo específico número 4 y 5.

- En la detección de defectos, la etapa de extracción de características de la CNN generó vectores descriptores de tamaño  $1 \times 3249$ , con las características descriptivas de los diferentes defectos etiquetados, tanto de borde como defectos internos en la pieza. Debido a la poca diferenciación entre los patrones que identifican los defectos, se implementó las máquinas de soporte vectorial en la etapa de clasificación de la CNN para los defectos de las piezas obteniendo en la validación un valor de 83.33% de precisión de la red CNN+SVM. Se le da cumplimiento al objetivo específico número 1, 4 y 5.

- Se lograron los supuestos de la hipótesis al diseñar un sistema de inspección capaz de identificar los dos tipos de piezas para los cuales se implementó y logrando identificar en un gran porcentaje los defectos que se le etiquetaron a las piezas. Con la integración del BOF en la etapa de identificación de piezas, se logró reducir el tiempo de procesamiento general del proceso de inspección en comparación con un sistema de inspección implementado solo con una arquitectura CNN, la cual demostró ser muy útil en la extracción de características en la identificación de defectos. Con esta investigación, se da paso a la búsqueda de nuevas estrategias de inspección capaces de adaptarse a los diferentes problemas existentes en los procesos de producción, en la etapa de inspección de calidad de fabricación de productos.

- Con la integración de estas técnicas se logra un sistema robusto ante pequeños cambios de luz en el ambiente, debido a que las redes se entrenaron con diferentes imágenes donde la intensidad de los píxeles varió en correspondencia a los cambios de luz. Aunado

a las características propias de las técnicas utilizadas que permiten el reconocimiento de objetos ante pequeños cambios de rotación, traslación y luz ambiental.

### **Trabajo futuro**

En base a la investigación realizada, se propone que en trabajos futuros referentes a la identificación de piezas y detección de defectos las consideraciones siguientes:

- Crear una base de datos con una cantidad mayor de imágenes para el enteramiento, con lo cual el porcentaje de clasificación obtenido puede incrementarse dando solución a las exigencias de producción en procesos de manufactureras.
- Utilizar otros métodos de pre-procesamiento de imagen que permitan dividir la misma en las diferentes regiones de interés, ubicando las zonas de mayor probabilidad de fallas en el proceso de fabricación y brindando la posibilidad de identificar más de un defecto en una misma pieza.
- Profundizar en otras técnicas del Deep Learning que permitan el análisis de imagen en grupos de manera simultánea utilizando mayores recursos de procesamiento de datos.

## 8. **BIBLIOGRAFÍA.**

- Ahmad, J., Muhammad, K., Bakshi, S., & Baik, S. W. (2018). Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets. *Future Generation Computer Systems*. <https://doi.org/10.1016/j.future.2017.11.002>
- Aldalur, B. . M. S. (2002). Realce de imágenes: filtrado espacial.
- Avelino Manzano Lizcano, J. (2015). Sistemas de ayuda al diagnóstico y reconocimiento de microcalcificaciones en mamografía mediante descriptores de escala y redes jerárquicas. Retrieved from [http://oa.upm.es/39251/1/JOSE\\_AVELINO\\_MANZANO\\_LIZCANO.pdf](http://oa.upm.es/39251/1/JOSE_AVELINO_MANZANO_LIZCANO.pdf)
- Ayala, R. A. R. (2014). *IMPLEMENTACIÓN EN UN FPGA DE LA FUNCIÓN DE FRONTERA PARA EL RECONOCIMIENTO DE OBJETOS*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.
- Baptista, D., & Morgado-Dias, F. (2013). NEW APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS IN MODELING & CONTROL A survey of artificial neural network training tools. <https://doi.org/10.1007/s00521-013-1408-9>
- BELL, J. (2015). *Machine Learning: Hands-On for Developers and Technical Professionals*.
- Bondi, L., Baroffio, L., Cesana, M., Tagliasacchi, M., Chiachia, G., & Rocha, A. (2016). Rate-energy-accuracy optimization of convolutional architectures for face recognition. *Journal of Visual Communication and Image Representation*, 36, 142–148. <https://doi.org/10.1016/j.jvcir.2015.12.015>

- Cabrera, M. P., Juarez, I. L., Cabrera, R. R., Osorio, R., & Gomez, H. (2010). Real Time Object Recognition Methodology. In *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference* (pp. 439–444). IEEE.  
<https://doi.org/10.1109/CERMA.2010.116>
- Cai, N., Chen, Y., Liu, G., Cen, G., Wang, H., & Chen, X. (2017). A vision-based character inspection system for tire mold. *Assembly Automation*, *37*(2), 230–237.  
<https://doi.org/10.1108/AA-07-2016-066>
- Camila, M. A. (2017). DEEP LEARNING APLICADO A IMÁGENES DE RETINA COMO HERRAMIENTA DE TAMIZAJE EN UN PROCESO DE DIAGNÓSTICO MÉDICO. Retrieved from <http://red.uao.edu.co/bitstream/10614/9568/1/T07237.pdf>
- Campos, V., Jou, B., & Giró-i-Nieto, X. (2017). From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction. *Image and Vision Computing*.  
<https://doi.org/10.1016/j.imavis.2017.01.011>
- Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B. (1992). Fuzzy ARTMAP: an adaptive resonance architecture for incremental learning of analog maps. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks* (Vol. 3, pp. 309–314). IEEE. <https://doi.org/10.1109/IJCNN.1992.227156>
- Carrillo, H. C. (2007). CONVERSIÓN DE TEXTO MANUSCRITO A FORMATO DIGITAL UTILIZANDO MÁQUINAS DE SOPORTE VECTORIAL T E S I S. Retrieved from [http://tesis.ipn.mx/bitstream/handle/123456789/3478/2342\\_2007\\_CIC\\_MAESTRIA\\_cisneros\\_carrillo\\_hermenegildo.pdf?sequence=1](http://tesis.ipn.mx/bitstream/handle/123456789/3478/2342_2007_CIC_MAESTRIA_cisneros_carrillo_hermenegildo.pdf?sequence=1)



- Castro, M. F. J., Castelán, M., & López, J. I. (2010). Light source intensity adjustment for enhanced feature extraction. Paper presented at the Mexican Conference on Pattern Recognition.
- Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). Convolutional Neural Network Committees for Handwritten Character Classification. In *2011 International Conference on Document Analysis and Recognition* (pp. 1135–1139). IEEE. <https://doi.org/10.1109/ICDAR.2011.229>
- Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3642–3649). IEEE. <https://doi.org/10.1109/CVPR.2012.6248110>
- Cruz, P. P. (2010). Inteligencia artificial con aplicaciones a la Ingeniería.
- Demuth, H. B., Beale, M. H., Jess, O. De, & Hagan, M. T. (2014). *Neural Network Design* (2nd ed.).
- Duan, G., Wang, H., Liu, Z., & Chen, Y.-W. (2012). A Machine Learning-Based Framework for Automatic Visual Inspection of Microdrill Bits in PCB Production. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1679–1689. <https://doi.org/10.1109/TSMCC.2012.2216260>
- El-Naqa, I., Yongyi Yang, Wernick, M. N., Galatsanos, N. P., & Nishikawa, R. M. (2002). A support vector machine approach for detection of microcalcifications. *IEEE Transactions on Medical Imaging*, *21*(12), 1552–1563. <https://doi.org/10.1109/TMI.2002.806569>
- Elleuch, M., Maalej, R., & Kherallah, M. (2016). A New Design Based-SVM of the CNN

Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition.

*Procedia - Procedia Computer Science*, 80, 1712–1723.

<https://doi.org/10.1016/j.procs.2016.05.512>

Galdámez, P. L., Raveane, W., & González Arrieta, A. (2017). A brief review of the ear recognition process using deep neural networks. *Journal of Applied Logic*, 24, 62–70.

<https://doi.org/10.1016/j.jal.2016.11.014>

Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs.

*Expert Systems with Applications*, 66, 295–301.

<https://doi.org/10.1016/j.eswa.2016.08.057>

Grágeda, A. R., & Maldonado, L. P. (2007). Sistema Integrado de Manufactura-“SIM.”

Guillermo Manuel, S., Mario Sergio, C., & Diego Orlando, L. (2011). Separación de granos en contacto con spline – cúbico para la aplicación en visión artificial, (1). Retrieved

from

[http://sedici.unlp.edu.ar/bitstream/handle/10915/57673/Documento\\_completo.pdf-PDFA.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/57673/Documento_completo.pdf-PDFA.pdf?sequence=1)

Guo, Q., Wang, F., Lei, J., Tu, D., & Li, G. (2016). Convolutional feature learning and Hybrid CNN-HMM for scene number recognition. *Neurocomputing*, 184, 78–90.

<https://doi.org/10.1016/j.neucom.2015.07.135>

Hernández, J. J. C., Fernández, L. P. S., Bautista, I. H., & López, J. H. (2017). *Tecnología y Ciencias del Agua. Tecnología y Ciencias del Agua* (Vol. 8). Retrieved from

<http://www.revistatyca.org.mx/ojs/index.php/tyca/article/view/1369/1290>

- Iorio, J., & Taylor, J. E. (2014). Boundary object efficacy: The mediating role of boundary objects on task conflict in global virtual project networks. *International Journal of Project Management*, 32(1), 7–17. <https://doi.org/10.1016/j.ijproman.2013.04.001>
- Kamnitsas, K., Ledig, C., Newcombe, V. F. J., Simpson, J. P., Kane, A. D., Menon, D. K., ... Glocker, B. (2017). Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis*, 36, 61–78. <https://doi.org/10.1016/j.media.2016.10.004>
- Kawahara, J., Brown, C. J., Miller, S. P., Booth, B. G., Chau, V., Grunau, R. E., ... Hamarneh, G. (2017). BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146, 1038–1049. <https://doi.org/10.1016/j.neuroimage.2016.09.046>
- Liberda, A., Lilja, A., & Lindström, J. (2016). Image segmentation and convolutional neural networks as tools for indoor scene understanding Final report. Retrieved from <http://publications.lib.chalmers.se/records/fulltext/242058/242058.pdf>
- Liu, F., Lin, G., & Shen, C. (2015). CRF learning with CNN features for image segmentation. *Pattern Recognition*, 48(10), 2983–2992. <https://doi.org/10.1016/j.patcog.2015.04.019>
- Lopez-Juarez, I. (2012). The Use of Contour, Shape and Form in an Integrated Neural Approach for Object Recognition.
- López-Juárez, I., Castelán, M., Castro-Martínez, F. J., Peña-Cabrera, M., & Osorio-Comaran, R. (2013). Using Object's Contour, Form and Depth to Embed Recognition Capability into Industrial Robots. <https://doi.org/10.1016/S1665->

6423(13)71511-6

López-Juárez, I., Rios-Cabrera, R., Peña-Cabrera, M., Méndez, G. M., & Osorio, R. (2012).

Fast Object Recognition for Grasping Tasks using Industrial Robots. *Computación y Sistemas*, 16(4), 421–432. Retrieved from

<http://scielo.unam.mx/pdf/cys/v16n4/v16n4a5.pdf>

Lopez, I., Peña, C. M., & Reyes, A. A. (2010). Using Object's Contour and Form to

Embed Recognition Capability into Industrial Robots Robot Manipulators Trends and Development.

Lopez, I. Rios, C. R., Peña, C. M., & Osorio, C. R. (2010). Learning and fast object

recognition in robot skill acquisition: a new method.

López, A., Valveny, E., & José Villanueva, J. (2005). Real-time quality control of surgical material packaging by artificial vision. *Assembly Automation*, 25(3), 223–229.

<https://doi.org/10.1108/01445150510610944>

Mauricio, G., Rey, Z., Parra Rodríguez, C. A., Ruth, M., Torres, M., Julio, C., & Medina,

B. (2007). ESTACIÓN DE CONTROL DE CALIDAD POR VISIÓN ARTIFICIAL PARA UN CENTRO DE MANUFACTURA INTEGRADA POR COMPUTADOR (CIM)\* QUALITY CONTROL STATION VIA ARTIFICIAL VISION FOR A

COMPUTER INTEGRATED MANUFACTURING CENTER (CIM), 11(1), 33–55.

Retrieved from <http://www.redalyc.org/pdf/477/47711103.pdf>

Mcculloch, W. S., & Pitts, W. (1943). A LOGICAL CALCULUS OF THE IDEAS

IMMANENT IN NERVOUS ACTIVITY. *BULLETIN OF MATHEMATICAL*

*BIOPHYSICS*, 5. Retrieved from

<https://link.springer.com/content/pdf/10.1007%2F02478259.pdf>

Morales, G., & Gómez, A. (2005). Estudio e implementación de una herramienta basada en Máquinas de Soporte Vectorial aplicada a la localización de fallas en sistemas de distribución.

Navarro-Gonzalez, J. L., Lopez-Juarez, I., Rios-Cabrera, R., & Ordaz-Hernández, K. (2015). On-line knowledge acquisition and enhancement in robotic assembly tasks. *Robotics and Computer-Integrated Manufacturing*, 33, 78–89.  
<https://doi.org/10.1016/j.rcim.2014.08.013>

Nian Cai, Jianfa Lin, Qian Ye, Han Wang, Shaowei Weng, & Bingo Wing-Kuen Ling. (2016). A New IC Solder Joint Inspection Method for an Automatic Optical Inspection System Based on an Improved Visual Background Extraction Algorithm. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 6(1), 161–172. <https://doi.org/10.1109/TCPMT.2015.2501284>

Olabe, X. B. (1998). REDES NEURONALES ARTIFICIALES Y SUS APLICACIONES. Retrieved from [https://ocw.ehu.es/file.php/102/redes\\_neuro/contenidos/pdf/libro-del-curso.pdf](https://ocw.ehu.es/file.php/102/redes_neuro/contenidos/pdf/libro-del-curso.pdf)

Peña-Cabrera, M., Osorio, R., López-Juárez, I., & Ríos-Cabrera CIATEQ, R. A. (2004). Un Proceso de Aprendizaje para Reconocimiento de Objetos en Línea en Tareas Robotizadas. Retrieved from [http://www.iiisci.org/Journal/CV\\$/risi/pdfs/P219137.pdf](http://www.iiisci.org/Journal/CV$/risi/pdfs/P219137.pdf)

Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). ScienceDirect Convolutional Neural Networks for Diabetic Retinopathy. *Procedia - Procedia*

- Computer Science*, 90(2016), 200–205. <https://doi.org/10.1016/j.procs.2016.07.014>
- Ratha, N. K., & Jain, A. K. (1999). Computer vision algorithms on reconfigurable logic arrays. *IEEE Transactions on Parallel and Distributed Systems*, 10(1), 29–43.  
<https://doi.org/10.1109/71.744833>
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.  
<https://doi.org/10.1037/h0042519>
- Rumelhart, D. (1986). McClelland. *Parallel distributed processing*: Cambridge.
- Santoni, M. M., Sensuse, D. I., Arymurthy, A. M., & Fanany, M. I. (2015). Cattle Race Classification Using Gray Level Co-occurrence Matrix Convolutional Neural Networks. *Procedia Computer Science*, 59, 493–502.  
<https://doi.org/10.1016/j.procs.2015.07.525>
- Segal, R., Kothari, M. L., & Madnani, S. (2000). Radial basis function (RBF) network adaptive power system stabilizer. *IEEE Transactions on Power Systems*, 15(2), 722–727. <https://doi.org/10.1109/59.867165>
- Smirnov, E. A., Timoshenko, D. M., & Andrianov, S. N. (2014). Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks. *AASRI Procedia*, 6, 89–94. <https://doi.org/10.1016/j.aasri.2014.05.013>
- Stutz, D. (2014). Understanding Convolutional Neural Networks. Retrieved from <http://davidstutz.de/wordpress/wp-content/uploads/2014/07/seminar.pdf>
- Sun, T.-H., Tien, F.-C., Tien, F.-C., Kuo, R.-J., Sun, T.-H., Tien, F.-C., & Kuo, R.-J.

(2016). Automated thermal fuse inspection using machine vision and artificial neural networks. *Journal of Intelligent Manufacturing*, 27, 639–651.

<https://doi.org/10.1007/s10845-014-0902-y>

Tim, J. M. (2008). *Artificial Intelligence—A System Approach*. Computer Science Series, Infinity Science Press.

Wagner, R., Thom, M., Schweiger, R., Palm, G., & Rothmel, A. (2013). Learning convolutional neural networks from few samples. In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). IEEE.

<https://doi.org/10.1109/IJCNN.2013.6706969>

Weimer, D., Scholz-Reiter, B., & Shpitalni, M. (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1), 417–420. <https://doi.org/10.1016/j.cirp.2016.04.072>

Wernick, M., Yang, Y., Brankov, J., Yourganov, G., & Strother, S. (2010). Machine learning in medical imaging. *IEEE Signal Processing Magazine*, 27(4), 25–38. <https://doi.org/10.1109/MSP.2010.936730>

Yu, Q., & Wang, K. (2013). 3D vision based quality inspection with computational intelligence. *Assembly Automation*, 33(3), 240–246. <https://doi.org/10.1108/AA-12-2013-065>