



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

“IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL EN
LÍNEAS DE PRODUCCIÓN DE ATÚN, PARA LA DETECCIÓN DE
FALLAS DE CODIFICADO EN LA EMPRESA PUERTOMAR S.A
PROVINCIA DE MANABÍ.”

WILSON SIGIFREDO PALMA SANTACRUZ

Trabajo de Titulación modalidad: Proyectos de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito parcial para la obtención del grado de

**MAGÍSTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN
INDUSTRIAL**

RIOBAMBA – ECUADOR

Septiembre 2020



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado: “IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL EN LÍNEAS DE PRODUCCIÓN DE ATÚN, PARA LA DETECCIÓN DE FALLAS DE CODIFICADO EN LA EMPRESA PUERTOMAR S.A PROVINCIA DE MANABÍ.”, de responsabilidad del Ing. Wilson Sigifredo Palma Santacruz, ha sido prolijamente revisado y se autoriza su presentación.

ING. RAÚL MARCELO LOZADA YÁNEZ; PH.D

PRESIDENTE

ING. YOLANDA EUGENIA LLOSAS ALBUERNE; PH.D

DIRECTORA

ING. JULIO CESAR GUAMÁN SEGARRA; Mag.

MIEMBRO DEL TRIBUNAL

ING. ALFONSO JAVIER GUNSHA MORALES; Mag.

MIEMBRO DEL TRIBUNAL

Riobamba, Septiembre 2020

DERECHOS INTELECTUALES

Yo, WILSON SIGIFREDO PALMA SANTACRUZ, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, y que el patrimonio intelectual generado por la misma pertenece a la Escuela Superior Politécnica de Chimborazo.

Ing. Wilson Sigifredo Palma Santacruz

Nº Cédula: 1312913567

© 2020, **Wilson Sigifredo Palma Santacruz**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

DECLARACIÓN DE AUTENTICIDAD

Yo, Wilson Sigifredo Palma Santacruz, declaro que el presente Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación de Maestría.

Ing. Wilson Sigifredo Palma Santacruz

Nº Cédula: 1312913567

DEDICATORIA

Este trabajo es la suma en conjunto de grandes esfuerzos, desde lo más profundo de mi corazón se la dedicado a mis padres Wilson Y Martha por creer en mí y en lo que soy capaz de hacer, porque con su ejemplo me ha enseñado que cuando se quiere todo es posible, porque son ellos los ángeles terrenales que han hecho de este servidor una persona de bien, a mi amada esposa Gabriela mi compañera fiel por estar a mi lado siempre por su tenacidad que no se rompe, por su confianza y su apoyo incondicional, a mi hija Gianella por ser la fuerza que me impulsa todos los días a querer dejarle para ella un mundo mejor, a mis hermanos Segundo, Geoconda y Vitelio a quienes amo con todo mi corazón, a mis sobrinos Fernando y Lady quienes me enseñaron a ser padre a muy corta edad.

Wilson

AGRADECIMIENTO

Agradezco este trabajo a todos esos seres que hicieron posible su consecución, a la empresa PuertoMar S.A por brindarme facilidad a sus instalaciones para realizar los respectivos estudios de campo y posteriormente la implementación.

A la Escuela Superior Politécnica de Chimborazo por haberme permitido escalar un peldaño en mi vida profesional.

A los miembros del tribunal de tesis, a la Ing. Yolanda Llosas como directora, Ing. Julio Guamán la Ing. María Rodríguez Gámez y al Ing. Alfonso Gunsha Morales quienes supieron guiar de forma eficiente y oportuna en cada fase del desarrollo de este proyecto, por ende mi gratitud siempre será eterna.

Wilson Palma Santacruz

CONTENIDO

RESUMEN

ABSTRACT

CAPÍTULO I

1.	INTRODUCCIÓN	1
1.1.	Planteamiento del problema.....	1
<i>1.1.1.</i>	<i>Situación problemática</i>	<i>1</i>
1.2.	Formulación del Problema.	2
1.3.	Preguntas directrices.	2
1.4.	Justificación.....	2
1.5.	Objetivo.	4
<i>1.5.1.</i>	<i>Objetivo General.</i>	<i>4</i>
<i>1.5.2.</i>	<i>Objetivos Específicos.....</i>	<i>4</i>
1.6.	Hipótesis.	4
<i>1.6.1.</i>	<i>Hipótesis General.....</i>	<i>4</i>
1.7.	Alcance.....	4
1.8.	Estructura del documento.	5

CAPÍTULO II

2.	MARCO DE REFERENCIA.	6
2.1.	Antecedentes.....	6
2.2.	Codificación Industrial.	6
2.3.	Fallas en la codificación industrial.....	9
<i>2.3.1.</i>	<i>Falla por código en las latas.</i>	<i>11</i>
<i>2.3.2.</i>	<i>Fallas por censado.</i>	<i>11</i>
<i>2.3.3.</i>	<i>Fallas por operación.</i>	<i>11</i>
2.4.	Sistema de automatización.	11
2.5.	Sistema electrónico.....	12
<i>2.5.1.</i>	<i>Sensores de proximidad.</i>	<i>12</i>

2.5.1.1.	<i>Fotoeléctricos.</i>	12
2.5.1.2.	<i>Inductivos.</i>	13
2.5.1.3.	<i>Capacitivos.</i>	14
2.5.1.4.	<i>Magneto-Resistivos.</i>	15
2.6.	Sistema de Visión Artificial.	15
2.6.1.	<i>Imágenes.</i>	15
2.6.2.	<i>Etapas en un sistema de Visión Artificial.</i>	18
2.6.3.	<i>Componentes de un sistema de visión artificial.</i>	20
2.6.4.	<i>Tipos de iluminación en Sistemas de Visión artificial.</i>	20
2.6.5.	<i>Cámaras en sistemas de visión artificial.</i>	23
2.7.	Redes Neuronales Artificiales.	25
2.7.1.	<i>Ventajas que de las redes neuronales.</i>	26
2.7.2.	<i>Elementos básicos que constituyen una red neuronal.</i>	27
2.7.2.1.	<i>Función de entrada (Input function).</i>	27
2.7.2.2.	<i>Función de activación (activación function).</i>	28
2.7.2.3.	<i>Función de salida (output function).</i>	28
2.7.3.	<i>Redes neuronales Convolucionales.</i>	28
2.7.3.1.	<i>Convolución.</i>	28
2.7.3.2.	<i>Neocognitron.</i>	30
2.8.	Open CV.	31
2.9.	Tensor Flow.	32
 CAPÍTULO III		
3.	DISEÑO E IMPLEMENTACIÓN.	33
3.1.	Criterios de selección de componentes electrónicos.	33
3.1.1.	<i>Selección de la tarjeta de control.</i>	34
3.1.2.	<i>Selección de la cámara.</i>	35
3.1.3.	<i>Selección del sensor de presencia.</i>	37
3.2.	Diseño del sistema de visión artificial.	38

3.3.	Esquema general del sistema de visión artificial.....	38
3.4.	Implementación del sistema de visión artificial.	39
3.4.1.	<i>Adquisición de imágenes.....</i>	40
3.5.	Pre procesamiento.....	41
3.5.1.	<i>Descripción de las etapas.</i>	41
3.6.	Red neuronal	45
3.6.1.	<i>Creación y entrenamiento de la red neuronal.</i>	45
3.7.	Programación de la tarjeta de control.....	51
3.8.	GUI (Interfaz Gráfica de Usuario)	52
3.9.	Conexión del sistema de visión artificial.....	55

CAPÍTULO IV

4.	RESULTADOS EXPERIMENTALES Y DISCUSIÓN.....	57
4.1.	Clasificación de la red neuronal.....	57
4.2.	Validación de la hipótesis.	63
4.2.1.	<i>Eficiencia de la red neuronal aplicada en Puerto Mar a 84%.....</i>	66
4.2.2.	<i>Validación de la hipótesis.</i>	68
4.3.	Análisis financiero.....	70

CAPÍTULO V

5.	PROPUESTA	72
5.1.	Desarrollo de la propuesta alternativa	72
5.2.	Prueba de clasificación de latas de atún	75
	CONCLUSIONES.....	76
	RECOMENDACIONES.....	77

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 2-1. Componentes de un sistema electrónico.	12
Tabla 2-2. Técnicas de iluminación.....	21
Tabla 2-3. Ventajas y desventajas de la Cámara Matricial.....	23
Tabla 2-4. Ventajas y desventajas de una Wed Cam HD Pro C920.....	24
Tabla 2-5. Ventajas desventajas de una Cámara Térmica.....	25
Tabla 3-1. Matriz de conceptos para la selección de la tarjeta de control	33
Tabla 3-2. Especificaciones Técnicas Arduino Uno.....	34
Tabla 3-3 Matriz de conceptos para la selección de la cámara.....	35
Tabla 3-4. Especificaciones Técnicas Wed Cam HD Pro C920.....	36
Tabla 3-5. Características de un sensor de proximidad fotoeléctrico infrarrojo.....	37
Tabla 4-1. Clasificación de la red neuronal de las latas de atún según la presencia o ausencia del codificado.	57
Tabla 4-2. Resultados de las pruebas realizadas.....	62
Tabla 4-3. Pérdidas totales de la empresa por fallos en la codificación.....	63
Tabla 4-4. Pérdidas totales de la empresa por fallos en la codificación luego de implementar el sistema de visión artificial.....	64
Tabla 4-5. Pérdidas totales de la empresa por fallos en la codificación	65
Tabla 4-6. Datos adquiridos en PuertoMar S.A.....	66
Tabla 4-7. Análisis económico de los elementos utilizados en la elaboración del proyecto.....	70
Tabla 5-1. Pérdidas totales de la empresa por fallos en la codificación.....	74

ÍNDICE DE FIGURAS

Figura 2-1. Proceso productivo del atún.....	8
Figura 2-2. Información básica en latas de atún.....	9
Figura 2-3. Lata de atún no codificada.....	10
Figura 2-4. Frecuencia de las fallas de codificación.	10
Figura 2-5. Estructura de un sistema automatizado.....	11
Figura 2-6. Aplicación Fotoeléctrica.....	13
Figura 2-7. Sensor Inductivo	14
Figura. 2-8. Sensor Capacitivo	14
Figura 2-9. Sensor Magneto Resistivo.	15
Figura 2-10. Composición de las imágenes digitales	16
Figura 2-11. Imagen RGB.	17
Figura 2-12. Imagen en escala de grises.....	18
Figura 2-13. Etapas de un sistema de Visión Artificial.....	19
Figura 2-14. Diagrama de bloques de sistema de visión.....	19
Figura 2-15. Componentes de un sistema de visión artificial.	20
Figura 2-16. Cámara Matricial AP-3200-T	23
Figura 2-17. Web Cam HD Pro C920	24
Figura 2-18. Cámara Térmica Bobcat-640.....	25
Figura 2-19. Arquitectura de una red neuronal	27
Figura 2-20. Aplicación de una convolución con matrices.	30
Figura 2-21. Diagrama esquemático que ilustra las interconexiones entre capas en el neocognitron.....	31
Figura 3-1. Tarjeta de control Arduino Uno.....	34

Figura 3-2. Web Cam HD Pro C920	36
Figura 3-3. Sensor de proximidad fotoeléctrico infrarrojo.....	37
Figura 3-4. Esquema del sistema de visión artificial.....	39
Figura 3-5. Procedimiento del sistema de visión artificial	40
Figura 3-6. Etapas del pre procesamiento de imagen.....	41
Figura 3-7. Conversión a escala de grises y filtro de	43
Figura 3-8. Resultado de la aplicación de la máscara	44
Figura 3-9. Resultado de aplicación filtro	45
Figura 3-10. a) Librerías de Python dentro de una carpeta scripts b) Carpetas con las clases que se quieren clasificar.	46
Figura 3-11. a) Dataset de imágenes de latas con código b) Dataset de imágenes de latas sin código	47
Figura 3-12. Código para el entrenamiento de la red neuronal.....	48
Figura 3-13. Resultados finales de la red neuronal	48
Figura 3-14. Archivos generados luego de la creación	49
Figura 3-15. Creación del entorno de la GUI.....	52
Figura 3-16. Estructura del panel de diseño del GUI de QtCreator.....	53
Figura 3-17. Partes que conforma el GUI.....	54
Figura 3-18. Conexión del sistema de visión artificial.....	55
Figura 3-19. Descripción del proceso de visión artificial.....	56
Figura 4-1. Distribución Normal	69
Figura 5-1. Resultado del análisis a las latas de atún con codificación correcta.....	75
Figura 5-2. Actuador activado	75

ÍNDICE DE GRÁFICOS

Gráfico 4-1 Latas con errores vs pérdidas	65
Gráfico 4-2 Pre Implementación vs Post Implementación con 8 minutos de falla	66
Gráfico 4-3 Pre Implementación Vs Post Implementación con 15 minutos de falla.	67
Gráfico 4-4 Pre Implementación Vs Post Implementación con 10 minutos de falla.	68

ÍNDICE DE ANEXOS.

Anexo A: Especificaciones Técnicas De Los Elementos Implementados

Anexo B: Código Programa Qt

Anexo C: Recolección De Datos

Anexo D: Tabla de distribución normal

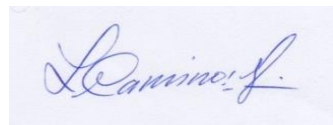
Anexo E: Carta de aceptación Puerto Mar

Anexo F: Registro Fotográfico

RESUMEN

En este proyecto se planteó el diseño de un sistema de visión artificial capaz de determinar las fallas del codificado en las latas de atún a tiempo real tomando como base el procesamiento digital de imágenes utilizando filtros y algoritmos a través de un modelo de red neuronal artificial convolucional, implementada mediante el lenguaje de programación Python y el uso de la biblioteca de software libre llamada TensorFlow, las cuales conjuntamente permiten identificar si la lata de atún esta codificada de una manera correcta, o por lo contrario si presenta algún tipo de problema. Es necesaria la realización del diagnóstico de fallos en tiempo real, llegando no sólo a su detección y aislamiento sino también a la estimación de su magnitud, con objetivo a interrumpir la propagación de los fallos en un determinado sistema. En el caso de tener que detectar fallos en procesos complejos, donde existe incertidumbre paramétrica y estructural como es el caso de estudio que ocupa (línea de producción de atún), no existen modelos matemáticos del mismo, por lo que se requieren técnicas de inteligencia artificial Como resultado de este proyecto se entrenó una red neuronal convolucional con una eficiencia del 84% mediante las librerías de TensorFlow y con ayuda del modelo Convolucional (Neocognitron Mobilenet_v1_1.0_224), para ello se utilizó un data set de 417 imágenes, y se entrenó mediante 4000 pasos de iteración, con la implementación del sistema de visión artificial la empresa mejoró su eficiencia en un 84 % dentro de la línea de codificación de latas de atún generando así una ganancia sustancial en sus líneas de producción. Finalmente, se recomienda, entrenar el modelo con un data set más grande con el fin de mejorar la eficiencia de la red neuronal, y lograr una efectividad superior al 95%.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA CONTROL AUTOMÁTICO >; <VISION ARTIFICIAL>; < PROCESAMIENTO DE IMÁGENES>; <REDES NEURONALES>; <COFIFICACIÓN INDUSTRIAL>;



02-07-2020

0102-DBRAI-UPT-2020

ABSTRACT

In this project it was proposed: the design of an artificial vision system capable of determining the coding faults in the tuna cans in real time based on the digital image processing using filters and algorithms through an artificial neural network model convolutional, implemented through the Python programming language and the use of the free software library called TensorFlow, which together allow us to identify if the can of tuna is encoded in a correct way, or on the contrary if it presents any type of problem. It is necessary to carry out fault diagnosis in real time, reaching not only its detection and isolation but also the estimation of its magnitude, with the aim of interrupting the propagation of faults in a given system. In the case of having to detect failures in complex processes, where there is parametric and structural uncertainty such as the case study that it occupies (tuna production line), there are no mathematical models of it, so artificial intelligence techniques are required. As a result of this project, a convolutional neural network with an efficiency of 84% was trained using TensorFlow libraries and with the help of the Convolutional model (Neocognitron Mobilenet_v1_1.0_224), for which a data set of 417 images was used, and was trained by 4000 iteration steps, with the implementation of the artificial vision system the company improved its efficiency by 84% within the tuna cans coding line, thus generating a substantial profit in its production lines. Finally, it is recommended to train the model with a larger data set in order to improve the efficiency of the neural network, and achieve an effectiveness greater than 95%.

KEY WORDS: <AUTOMATIC CONTROL>; <ARTIFICIAL VISION>; <IMAGE PROCESSING>; <NEURAL NETWORKS>; <INDUSTRIAL CODING>;

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema

1.1.1. *Situación problemática*

En la actualidad, la competencia dentro de un mercado en crecimiento exponencial pone en amenaza a todas las industrias a producir en mayor cantidad sus productos con una alta calidad y flexibilidad a un menor costo y tiempo posible, pero en estos tiempos hay muchos procesos industrializados y más aquellos que son de visualización donde se ejecutan en base a la experiencia de la inspección humana y su agudeza visual para reconocer fallas en los productos, en vista de este problema la industria se ha visto en la obligación de promover y desarrollar sistemas que sean capaces de resolver tareas de forma automática.

Para lograr lo anterior es necesario tanto automatizar las partes del proceso como integrar las diferentes áreas, entregando a los usuarios información confiable y en tiempo real.

Respecto al control de manufactura en un proceso, se pueden identificar como problemas, la obsolescencia o falta de equipo automático de medición y control lo que dificulta la visualización y control de las variables operativas, la necesidad de ampliación o modernización, el hardware (HW) y software (SW) con tecnología poco modular y dificultad de programación e integración con otros sistemas, la falta de sincronización de las operaciones de producción y la cadena de suministros y la necesidad de alimentar al Enterprise Resource Planning (ERP) con información de la planta actualizada en forma automática.

En el Ecuador y en la provincia de Manabí, la producción del atún es uno de los cimientos más robustos dentro de la economía ya que brinda una fuente sustentable de capital y trabajo, sean estas de participación directa o indirecta, es por tal motivo que el proceso cuenta con una gran infraestructura y posibilidades continuas de mejora, para efectivizar y hacer más eficiente las diferentes líneas.

Una de las exigencias más grandes a las que se enfrenta la modernización de la industria en proceso, son los tiempos y costos de producción que sufren por falta de HW de instrumentación y

control para la medición de variables de producción y la falta de información totalizada y confiable de las mismas, las cuales deben realizarse a mano con riesgo de cometer errores, por lo que el uso de maquinaria electrónica y su integración hacen que se diseñen sistemas automátats, los que aún tienen que ser perfeccionados y en casos totalmente desarrollados debido a que las condiciones ambientales de producción son variables o tienen aspectos diversos que no pueden ser despreciados ya que estos traen consigo muchos errores técnicos en equipos electrónicos tan sensibles.

Ante este escenario es fundamental efectuar un estudio de beneficios y perturbaciones para alcanzar esta optimización y el análisis de su correlación para determinar su influencia en el incremento de la eficiencia.

1.2. Formulación del Problema.

¿Las fallas en la codificación de las latas de atún, se soluciona mediante la integración de un sistema de visión artificial sustentado en redes neuronales que actúa a tiempo real en la línea de producción de la empresa Puertomar S.A, Provincia de Manabí.?

1.3. Preguntas directrices.

¿La implementación de un sistema de visión artificial en líneas de codificado permite reducir los costos de operación en el proceso industrial del atún?

¿Si se implementa un sistema automatizado en el proceso industrial del atún se agilizará la producción y eficiencia en la línea de producción?

¿Los sistemas actuales en la línea industrializada del atún cumplen con los requerimientos técnicos específicos para reducir las tareas de inspección que realizan los operadores?

1.4. Justificación.

El desarrollo industrial abre la posibilidad de nuevas aplicaciones automatizadas, todas enmarcadas en la optimización de sus recursos y la generación de eficiencia en todo el proceso. Al mismo tiempo este desarrollo crea una brecha tanto investigativa como aplicada, debido a la variedad de etapas y las condiciones de trabajo en los diferentes procesos productivos haciendo que no se pueda diseñar un estándar fijo sino tentativo (Prateek, 2017).

El presente proyecto, surge ante la necesidad de la constante evolución que sufren los cambios en el sector industrial para lo cual se pretende desarrollar un sistema automático que lleve a cabo una inspección visual artificial que sea objetiva, rápida y confiable en el sistema ante una falla en el codificado de las latas de atún.

Una correcta codificación de los productos, en este caso latas de atún es necesario para los fabricantes de este alimento de consumo rápido, porque se mejora la eficiencia y se agiliza la visibilidad de la secuencia del suministro, así mismo que se ofrece a los clientes la información veraz sobre los productos que están adquiriendo, permitiendo que este tipo de alimento cuente con los códigos correctamente impresos en las latas (González José, 1973)

Las fallas en el proceso de codificación de un producto son altamente costosas para la industria, puesto que esto implica el coste de reprocesamiento del producto suponiendo el caso de que este se pueda reprocesar y que la planta industrial tenga la capacidad para hacerlo, dentro de un ámbito de producción constante en el que se trabaja 24 horas al día y los 7 días de la semana tener este tipo de fallas está relacionado directamente con las mediciones de la eficiencia de la línea de producción, de manera que muy a menudo se desconoce el efecto de estos errores y que su efecto es acumulativo.

Mediante la implementación de un sistema de visión artificial que actúe a tiempo real, y sea capaz de detectar fallas en el codificado de las latas de atún se pretende a primera instancia reducir los problemas de salud en los operarios técnicos, que vienen dado por el estrés, cansancio visual y fatiga, que se origina por la realización de tareas repetitivas.

También tiene como finalidad el incremento de la productividad a costos reducidos, puesto que eliminaría el proceso de reprocesamiento de las latas de atún que no se codifican por lo que se detectan los errores a tiempo real, dado que las máquinas que realizan el trabajo de codificado tienen un alto índice de fallas, de acuerdo a una encuesta realizada en las fábricas atuneras cierta población de ellas expuso que tienen problemas con la codificación al menos dos veces diaria, y un cuarto de estas expusieron que tenían fallas en la codificación al menos de tres y cuatro veces al día.

Ante esta consecuencia se debe desarrollar estudios permanentes para obtener la optimización de la producción y en el sector atunero esta implementación marcará un inicio al manejo del proceso de producción del sector, puesto que será pionero a nivel de la provincia porque no hay ninguna industria atunera que cuente con un sistema inteligente para el desarrollo óptimo del proceso dejando grandes beneficios para el sector, que serán un modelo para todo este grupo industrial, los beneficiarios directos será las plantas procesadoras y el consumidor final, además se tendrá como beneficiario indirecto a una nación que fortalece los procesos en una línea de producción que es parte importante de su matriz productiva.

Es importante tomar en cuenta que las industrias atuneras en la provincia de Manabí, las que en su mayoría se encuentran en los cantones Manta, Montecristi y Jaramijó, y por ende tienen normas de control y seguridad interna que son muy estrictas por lo que exteriorizar sus procesos no se permite, ante estas condiciones de confidencialidad la empresa Puertomar S.A presta sus instalaciones para el desarrollo de este proyecto y se hará público el nombre de ella ya que será parte de este proceso de estudio.

1.5. Objetivo.

1.5.1. Objetivo General.

Implementar un sistema de visión artificial que se ejecute a tiempo real para la detección de fallas de codificado en líneas de producción de atún, en la empresa Puertomar S.A, Provincia de Manabí.

1.5.2. Objetivos Específicos.

- ✓ Analizar las condiciones técnicas para el uso de sistema de visión artificial en el proceso de producción del atún.
- ✓ Desarrollar un modelo de visión, sustentado en redes neuronales artificiales (neocognitron) para la detección y aviso emergente ante fallas de codificado mediante alarma sonora.
- ✓ Obtener el algoritmo de adquisición y el respectivo procesamiento de las imágenes.
- ✓ Analizar mediante una matriz de comparación la mejora del proceso con la implementación del sistema de visión artificial.

1.6. Hipótesis.

1.6.1. Hipótesis General.

Si se diseña un sistema de visión artificial sustentado en redes neuronales se puede mejorar la eficiencia del proceso de codificación del atún.

1.7. Alcance.

El propósito de este proyecto es diseñar un sistema de visión artificial sustentado en redes neuronales artificiales para el análisis por unidad en el proceso de codificación de las latas de atún en la empresa atunera Puerto Mar S.A. Para esto se llevara a cabo una investigación bibliográfica sobre de las fallas y repercusiones en el proceso de codificación, luego se investigará sobre el sistema de transporte, sistema de alimentación con el fin de obtener datos reales que sirvan para el proceso de entrenamiento del algoritmo.

Posteriormente, se seleccionarán los componentes para el sistema de visión artificial como son: cámara, sistema de iluminación, sensores actuadores y de más aspectos que influyen en la etapa de procesamiento digital de imágenes, para luego de esto poder desarrollar el respectivo algoritmo de detección de fallas, que permita dar aviso cuando la maquina codificadora presente alguna falla.

A continuación se llevara a cabo la implementación del sistema mecánico - electrónico en la Empresa Puerto Mar S.A, para realizar el análisis por el sistema de visión artificial y que permita la detección de fallas en la codificación, finalmente se desarrollará una Interfaz Humano Máquina (HMI) para la configuración de los respectivos parámetros y visualizar las variables del sistema; se llevara a cabo las respectivas pruebas de funcionamientos del sistema, con la debida autorización de los representantes de la Empresa para poder validar la hipótesis, y el prototipo de visión artificial diseñado.

1.8. Estructura del documento.

El capítulo 1 trata sobre la problemática que se estima resolver, de igual forma los antecedentes de trabajos similares que ayudarán en la elaboración e implementación del sistema de visión artificial. A su vez, se proponen los objetivos a desarrollar y también un resumen del alcance en el presente trabajo de titulación.

El capítulo 2 describe la investigación bibliográfica sobre la cual se fundamentara todo el desarrollo del proyecto, además de la conceptualización de las diferentes causas que provocan fallas en el proceso de codificado de las latas de atún, sistemas electrónicos, técnicas y metodologías dentro del campo de la visión artificial y redes neuronales artificiales.

El capítulo 3 detalla el proceso de diseño del sistema integral y la implementación del sistema, además el desarrollo del algoritmo de control de visión artificial y la comunicación con los diferentes sistemas.

El capítulo 4 se detallan las pruebas realizadas con el sistema de detección de fallas en el proceso de codificación en latas de atún, los mecanismos de alimentación y comunicación así como la activación de la alarma emergente en el proceso, de igual forma la validación de la hipótesis, y su estudio financiero.

El capítulo 5 presenta la propuesta desarrollada de una forma cronológica en función de la realización y cumplimiento de los objetivos planteados, además de la validación del sistema integral de vision artificial de acuerdo al desarrollo del estudio.

CAPÍTULO II

2. MARCO DE REFERENCIA.

2.1. Antecedentes.

Industrialmente el sector atunero es una de las industrias con mayor fuerza en el mercado nacional y es por eso que se ha catapultado internacionalmente, siendo hoy por hoy una de las industrias con mayor fuerza a nivel de exportación a nivel de país, debido a su gran volumen de producción ha ganado gran espacio en diferentes sectores de la sociedad, por eso que la modernización de la industria ligado con los grandes avances tecnológicos exigen los más altos niveles de continuidad en sus procesos de producción, un reto a cumplir, será implementar un sistema de visión artificial, que sea capaz de detectar fallas en el codificado con el afán de reducir los tiempos por el reprocesamiento del producto que básicamente son condiciones propias del proceso.

Según la Organización de las Naciones Unidas (ONU) los principales productores para la alimentación y la agricultura son Tailandia, Ecuador, España, China y Filipinas, tienen como reto principal superar los valores de decrecimiento de las exportaciones lo que en los últimos tiempos ha sido afectada (FAO, 2018).

Debido a estas circunstancias se debe reducir costos y mejorar los tiempos de producción ya que estos factores llevan a una falta de competitividad en el mercado en el país donde se camina a pasos agigantados hacia la modernización de la industria se debe incluir sistemas inteligentes en los procesos capaces de detectar fallas mejorara considerablemente las líneas de producción y en este caso de estudio las líneas atuneras.

En consecuencia a esto la zona industrial asume riesgos en el intento de mejorar la trazabilidad de sus productos pero por lo general sufren adversidades económicas en este desarrollo poco técnico de acuerdo a la continua innovación que la competitividad global requiere.

Cabe tomar en cuenta y dejar marcado como precedente que ninguna fabrica dedicada a la actividad de enlatado de atún dentro de la provincia de Manabí cuenta con un sistema de visión artificial inteligente para la detección de fallas en el codificado de las latas.

2.2. Codificación Industrial.

Los sistemas de codificación industriales son conjuntos de normas que identifican un determinado producto o algún servicio. Sin duda alguna el sistema de codificación por excelencia es útil tanto

para el fabricante de la misma forma para el comerciante y el consumidor.(González José, 1973). Al fabricante le es útil porque le ayuda en el control de sus inventarios y de igual forma a la distribución de sus productos, además le permite identificar su producto en cualquier parte del mundo.

Para el sector comercial también tiene sus grandes ventajas porque permite mantener un registro real de su stock y obviamente facilita la llevada de una forma ideal la contabilidad. El consumidor también se ve beneficiado en el proceso ideal de codificación ya que recibe un producto con una codificación adecuada veras y precisa, y de tal forma que un error en la codificación acarrea grandes problemas al momento de consumir un producto.

Lógicamente para que esto sea posible, el código debe ser único para cada producto y el sistema de codificación empleado debe de estar normalizado o estandarizado (González José, 1973). Las tecnologías existentes de las máquinas de codificación industrial en los que podemos citar como principales:

- Inyección de chorro de tinta.
- Transferencia térmica.
- Codificación laser.

Una codificación industrial adecuada tiene un alto costo técnico y obviamente de insumos por lo que las empresas tratan de simplificar el uso de codificadores, además de medir siempre los niveles de eficiencia en cada uno de sus procesos.

Los principales aspectos que determinan las características y usos de un codificador van a ser el tipo de mensaje, las velocidades de trabajo, el sustrato el área, entre otras condiciones que determinara la elección de determinadas marcas y modelos.

Dentro de un sistema de automatización, el conocimiento del proceso es importante al momento de elegir los equipos que de manera confiable marcarán la fiabilidad y confianza en el desarrollo de cualquier proyecto de automatización.

Conocer la información legal de los productos, denominada trazabilidad, en los sectores de producción y en este caso el sector atunero es porque aquí se suscribe el detalle fundamental para su comercialización ya que un sistema de codificación es un conjunto de normas que en nuestro caso identifican a los productos o servicios(González José, 1973).

Pero la inclemencia de las condiciones físicas en el proceso atunero pone alta la expectativa de la conveniencia de la automatización del proceso dentro de la línea que proporciona la información

básica del codificado de la lata específicamente, puesto que las máquinas que realizan el trabajo de codificación presentan fallas al momento de codificar.



Figura 2-1. Proceso productivo del atún.

Fuente(Garzón Juan, 2013).

De la forma como lo describe el proceso y en base a algunas experiencia en plantas productoras de atún, el desarrollo de la automatización de la codificación se plantea de una forma ideal puesto que es la etapa final en la transición de producción y culminación de producto, puesto que luego serán almacenados con sus respectivos lotes y productos de acuerdo a las normas establecidas en la comercialización que rigen dentro del país.

Al hablar de un sistema de inspección visual automática en el proceso de codificado en las latas de atún, básicamente se pretende realizar la combinación de hardware y software que permitan capturar la imagen y aplicarle una serie de técnicas que garantizan que el sistema tome una decisión al momento de extraer la información relevante al codificado en latas.



Figura 2-2. Información básica en latas de atún.

Elaborado por: Palma Wilson ,2019

LOTE: el lote permite identificar la fabricación de un producto concreto en un periodo de tiempo determinado. Se codifica con una L seguida de 4 dígitos.

HORA: la hora nos permite conocer el momento exacto en el que se ha producido la lata. Se identifica con 4 dígitos, los 2 primeros indicarán la hora y los 2 segundos los minutos.

LÍNEA: la línea de producción se identifica con un conjunto de letras y números que permiten conocer en qué lugar de la planta se ha fabricado el producto.

2.3. Fallas en la codificación industrial.

Se entiende por fallo todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros característicos) de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado (Puig, Quevedo, Escobet, Morcego, & Ocampo, 2004),(Blanke, Kinnaert, Lunze, & Staroswiecki, 2016).

Las fallas de codificación afectan a la calidad del producto y generan costes inaceptables en toda la empresa debido a los desechos, el reprocesamiento, las multas regulativas, el daño a la reputación de la marca, etc(Technologies, 2018).En la figura 2-2 se muestra la lata de atún no codificada.



Figura 2-3. Lata de atún no codificada.

Elaborado por: Palma Wilson ,2019

Las fallas de codificación son costosas, no solo para las operaciones de la planta sino también para la totalidad de la empresa.

Está el coste de reprocesamiento, suponiendo que el producto se puede reprocesar y que la planta tenga la capacidad de hacerlo. En un entorno de producción que trabaja las 24 horas del día y los 7 días de la semana, el reprocesamiento quizás no sea posible (Technologies, 2018). En la figura 2-3 se muestra la frecuencia de fallas en las industrias atuneras.

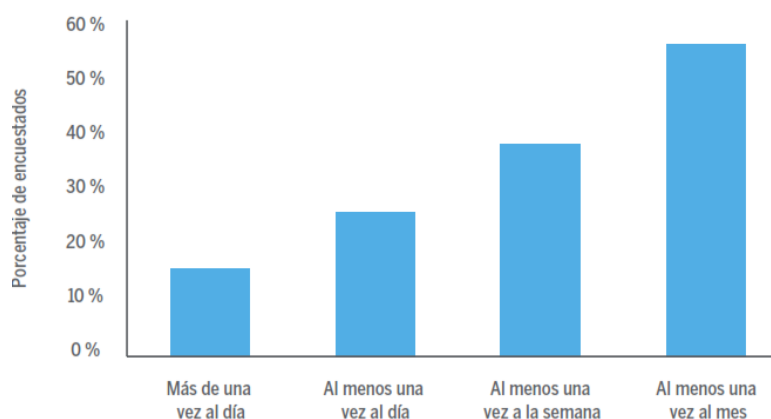


Figura 2-4. Frecuencia de las fallas de codificación.

Fuente: (Technologies, 2018)

2.3.1. *Falla por código en las latas.*

Es cuando alguna perturbación ha provocado paralización de impresión o errores en el mensaje, sea este por problemas de censado, o cortes de líneas por pérdida de viscosidad de la tinta (García Flores De Valga, 2018).

2.3.2. *Fallas por censado.*

Es cuando durante el proceso de codificación hay problemas de lectura del producto, sea este por vibraciones, textura, distancias o por humedad sobre el sensor, es por eso que la ubicación y selección del sensor es primordial durante el proceso (García Flores De Valga, 2018).

2.3.3. *Fallas por operación.*

Después de la jornada de trabajo la correcta limpieza de los componentes al finalizar los turnos es de importancia ya que repercute en la eficiencia del proceso, acción que solo puede ser efectuada por un operador (García Flores De Valga, 2018).

2.4. **Sistema de automatización.**

Conjunto de equipos, sistemas de información, y procedimientos que van a permitir asegurar un desempeño independiente del proceso, a través de operaciones de control y supervisión (Mendiburu, 2013). En la figura 2-5 se muestra la estructura de un sistema de automatización.

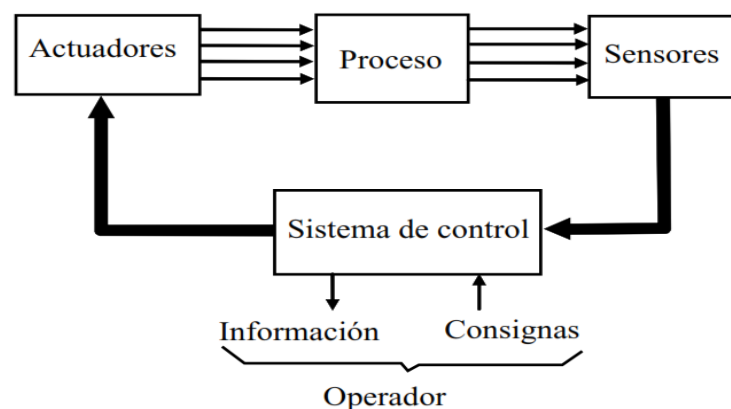


Figura 2-5. Estructura de un sistema automatizado.

Fuente:(Sanchis Roberto, Romero Julio, 2018)

Los nuevos paradigmas de la automatización industrial tienden a la distribución de la inteligencia entre los componentes de la cadena productiva y está encaminada hacia la integración de los diversos sistemas y aplicaciones existentes en las empresas, en la búsqueda de mejorar los procesos

y de disponer de información en tiempo real, que permita tomar decisiones acertadas (Bravo, Aguilar, & Rivas, 2004).

2.5. Sistema electrónico.

Son circuitos electrónicos cuya misión es controlar automáticamente el funcionamiento de algunas máquinas u operadores. En todo sistema electrónico tendremos dispositivos de estos tres tipos (Tecnología, 2010) que se muestran en la tabla 2-1.

Tabla 2-1. Componentes de un sistema electrónico.

<i>Dispositivos de un sistema electrónico</i>	Conceptos
<i>Dispositivos de entrada</i>	A partir de una señal exterior generan una señal eléctrica.
<i>Dispositivos de Proceso de señales.</i>	Componentes eléctricos que reciben las señales de los dispositivos de entrada y deciden cual es la acción a realizar
<i>Dispositivos de salida</i>	Tienen como misión ejecutar las acciones que deciden los dispositivos de los de proceso

Fuente (Tecnología, 2010).

2.5.1. Sensores de proximidad.

Dispositivos electrónicos que actúan de manera tal que en detectan la presencia de objetos para poder determinar exactamente la ubicación de un objeto en un determinado tiempo, son muy usados en aplicaciones como envasados, sistemas de control, para monitoreo y llenado, detección de obstáculos en sistemas inteligentes y en algunas configuraciones específicas. (Outeiriño, 2016)

2.5.1.1. Fotoeléctricos.

Los sensores fotoeléctricos pueden detectar la presencia o la ausencia de virtualmente cualquier tipo de objeto sin tener ningún contacto físico con este. Por lo que, estos pueden satisfacer un amplio rango de necesidades de control: estos pueden contar, notar la altura o el tamaño, la posición, monitorear velocidades de operación entre muchos otros (Lab-Volt (Quebec) Ltda., 2001). En la figura 2-6 se muestran aplicaciones de sensores fotoeléctricos.



Figura 2-6. Aplicación Fotoeléctrica.

Fuente:(Lab-Volt (Quebec) Ltda., 2001)

2.5.1.2. *Inductivos.*

Los sensores inductivos denominación abreviada IAS (interruptores inductivos de proximidad) contienen un transistor-oscilador, cuyo consumo de corriente es influenciado por la proximidad de metales y de otros materiales conductores de la electricidad (Rechner, 2019b).

El mismo efecto también puede conseguirse mediante materiales no conductores de la corriente.

La modificación de la corriente del oscilador es aumentada según el modelo de sensor del que se trate, formando una señal de salida de corriente de tipo lineal, o bien produce señales binarias que se emiten a través de un conmutador intensificador (Rechner, 2019b). En la figura 2-7, se observa un sensor inductivo.



Figura 2-7. Sensor Inductivo

Fuente: (Rechner, 2019b)

2.5.1.3. Capacitivos.

Los sensores capacitivos contienen un oscilador transistorizado, el cual se activa tan pronto como se acercan materiales metálicos o no metálicos, incluso líquidos. Ello sucede, siempre y cuando se sobrepase cierta capacidad. Cuanto menor es la constante dieléctrica ϵ_r tanto más se tiene que acercar el objeto (Rechner, 2019a).

Es posible detectar a través de materiales no metálicos, cuando la constante dieléctrica del objeto a detectar sea mayor en un factor aproximadamente de 5. La variación de la corriente del oscilador va aumentando según el modelo de sensor del que se trate, formando una de salida de corriente de tipo lineal, o bien produce señales binarias que se emiten a través de un amplificador (Rechner, 2019a). En la figura 2-8 se muestra el sensor capacitivo.



Figura. 2-8. Sensor Capacitivo

Fuente: (Rechner, 2019a)

2.5.1.4. *Magneto-Resistivos.*

Los sensores magneto-resistivos (MRS) detectan el movimiento de materiales ferromagnéticos por el cambio del flujo magnético. Estos sensores son aptos para la detección de la velocidad de ruedas dentadas y para control de parada (Rechner, 2016).

Las gamas de utilización son máquinas de construcción, vehículos sobre carriles, grandes motores diesel, turbinas y similares, mostrado en la figura 2-8.



Figura 2-9. Sensor Magneto Resistivo.

Fuente: (Rechner, 2016)

2.6. Sistema de Visión Artificial.

Se puede definir a la visión artificial (VA) o comprensión de imágenes ya que describe la deducción automática de la estructura y propiedades de un mundo tridimensional posiblemente dinámico, a partir de una o varias imágenes bidimensionales del mundo.(Gonzales et al., 2006)

Por tanto, la Visión Artificial o también llamada Visión por Computador, pretende capturar la información visual del entorno físico para extraer características relevantes visuales, utilizando procedimientos automáticos. Según Marr, “Visión es un proceso que produce a partir de imágenes del mundo exterior una descripción útil para el observador y no tiene información irrelevante”(Platero, 2009).

2.6.1. *Imágenes.*

Una imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por ejemplo su brillo o su matiz). (Velez Serrano, 2017)

Según (Vargas Víctor, 2010) una imagen digital, es una imagen $f(x; y)$ que ha sido discretizada en coordenadas espaciales y en brillo, puede ser considerada como una matriz o mallado cuyos índices de renglón y columna identifican un punto en la imagen y el correspondiente valor del elemento de la matriz que identifica el nivel de intensidad de luz en ese punto.

Los elementos de tal arreglo digital son llamados: elementos de imagen, elementos de pintura, pixel o pels cada uno con un lugar y valor específico(Vargas Víctor, 2010).En la figura 2-10 se muestra la composición digital de las imágenes.

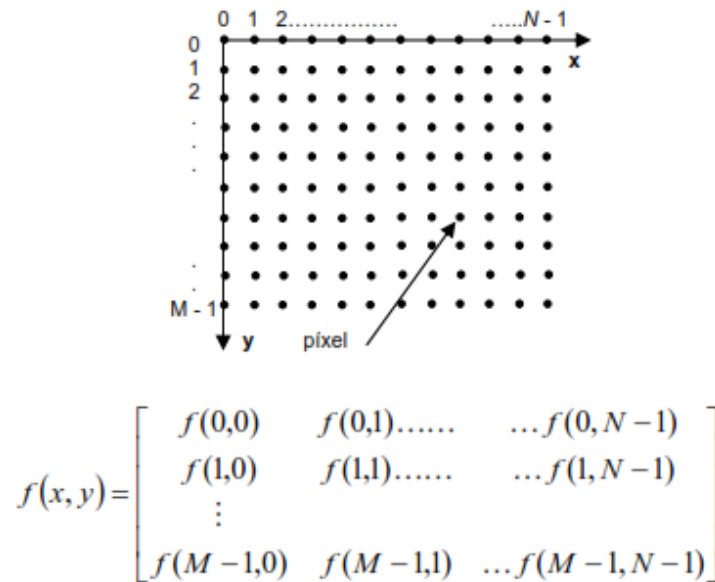


Figura 2-10. Composición de las imágenes digitales

Fuente: (Vargas Víctor, 2010)

- **Imagen RGB.**

Una imagen de color RGB es un arreglo de tres imágenes monocromáticas independientes de tamaño $m \times n$ correspondientes a la escala de rojos (R –red-), verdes (G –green-) y azules (B –blue-)(Alonso Marco, 2009).

En la figura 2-11 se observa la imagen de una lata de atún en correcta codificación.

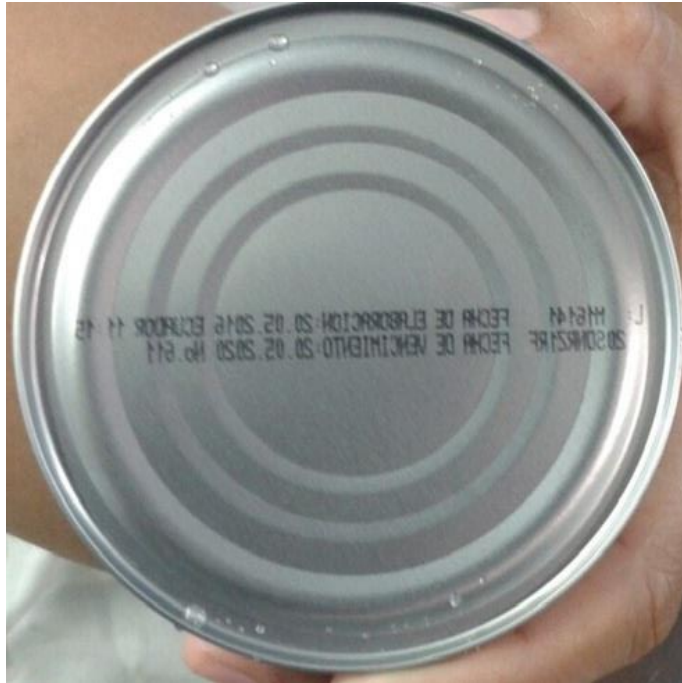


Figura 2-11. Imagen RGB.

Elaborado por: (Palma Wilson ,2019)

- **Imagen en escala de grises**

Según (Alonso Marco, 2009), una imagen en escala de grises es una función bidimensional de la intensidad de la luz que es denotada por $f(x, y)$, donde el valor o amplitud de f en las coordenadas espaciales (x, y) da la intensidad, iluminación o nivel de gris, de la imagen en aquellas coordenadas, en la figura 2-12 se observa una imagen en escala de grises.

$$I = 0.3 * R + 0.5 * G + 0.11 * B \quad (2.1)$$

El valor de intensidad puede ir desde 0 hasta 255; el cero representa el negro absoluto y 255 el blanco absoluto

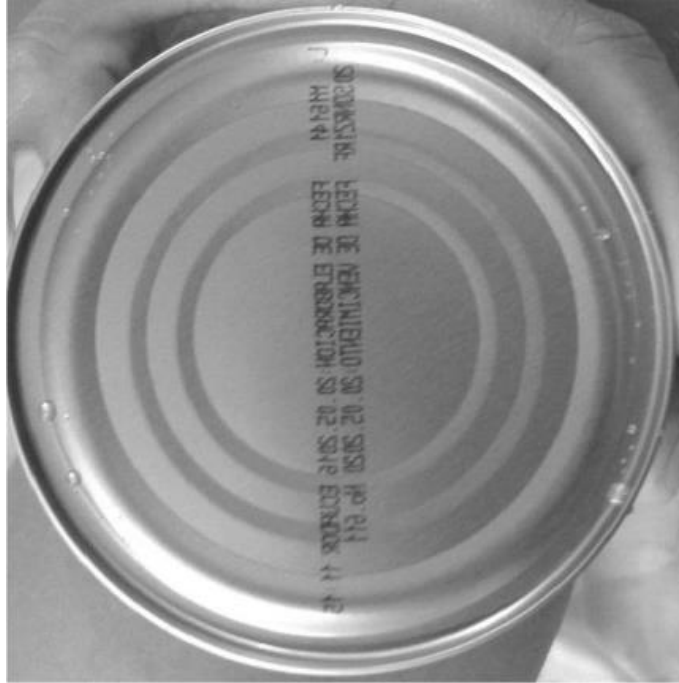


Figura 2-12. Imagen en escala de grises

Elaborado por: (Palma Wilson ,2019)

2.6.2. *Etapas en un sistema de Visión Artificial.*

En un sistema de visión artificial la función de entrada es una imagen adquirida por un elemento de adquisición mientras que la función de salida es la descripción básica de la escena la cual es obtenida a partir de la imagen. De algún modo, esta representación debe estar vinculada con la realidad que produce la imagen, que contiene toda la información que se requiere para realizar la tarea de interacción con el ambiente donde se va a llevar a cabo, un actuador por ejemplo (Gonzales et al., 2006). En la figura 2-13 se muestra las cuatro etapas en forma gráfica.

- **Primera fase:** Adquisición de imágenes digitales mediante algún tipo de sensor, típicamente un CCD.
- **Segunda fase:** Preparar la imagen eliminando las partes no útiles y/o realzando las partes interesantes.
- **Tercera fase:** Aislar los elementos de interés para su posterior interpretación.
- **Cuarta fase:** Distinguir los diferentes objetos segmentados en función de sus características



Figura 2-13. Etapas de un sistema de Visión Artificial

Elaborado por: (Palma Wilson ,2019)

Para fines industriales, un sistema autónomo de inspección basado en la técnica de visión artificial en codificado de latas de atún debe de ser capaz de identificar tres aspectos importantes como son ausencia, posición, y extensión del código impreso en la lata, estos aspectos se obtendrán mediante el uso de un algoritmo de procesamiento de imagen entrenado mediante una red neuronal convolutiva.

En resumen un sistema de visión artificial.

- Capta una imagen de un objeto real.
- La transforma en formato digital.
- La procesa mediante un algoritmo.
- Se obtienen los resultados del proceso.

En la figura 2-14 se encuentra el diagrama de bloques del sistema de visión artificial.

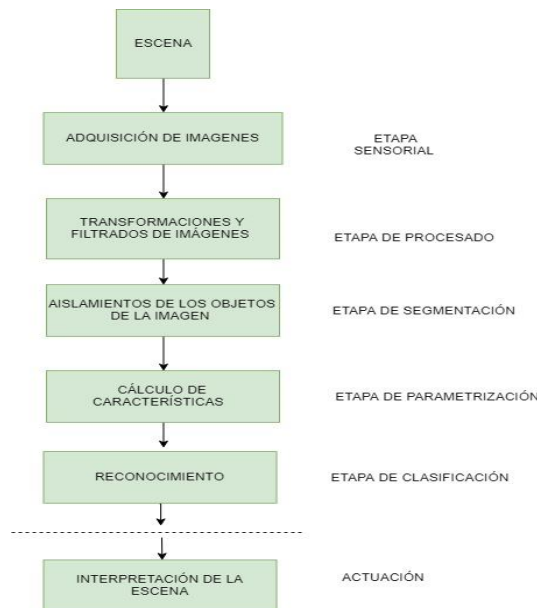


Figura 2-14. Diagrama de bloques de sistema de visión artificial.

Fuente: (Gonzales et al., 2006)

2.6.3. Componentes de un sistema de visión artificial.

Según (Platero, 2009), los dos pilares del sistema físico de visión artificial son: el sistema de formación de imágenes y el sistema de procesamientos de aquellas. En la primera etapa estaría constituido por el subsistema de iluminación, de captación de la imagen y de adquisición de la señal en el computador.

Una vez introducida la señal en el ordenador es procesada mediante los algoritmos para transformarla en información de alto nivel, la cual puede ser utilizada para su representación visual, para actuar en el planificador de un robot o ser fuente de datos para un autómata programable etc. En definitiva, múltiples periféricos pueden ser receptores de esta información y vincularse con el sistema de procesamiento de las imágenes, en la figura 2-15 se observa los componentes de un sistema de visión artificial.

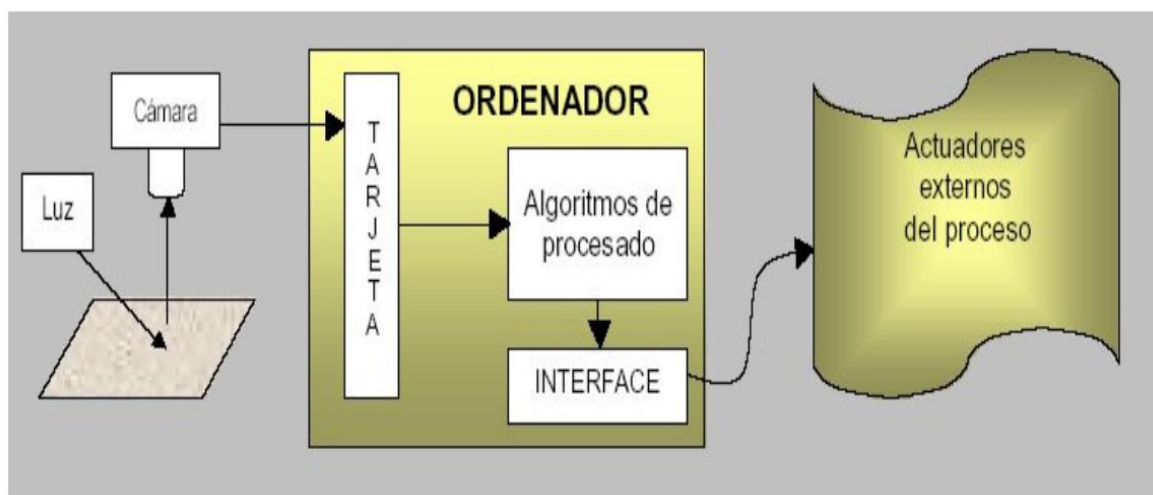





Figura 2-15. Componentes de un sistema de visión artificial.

Fuente: (Platero, 2009)

2.6.4. Tipos de iluminación en Sistemas de Visión artificial.

Las técnicas y los sistemas de iluminación son unos de los aspectos más importantes al momento de analizar una imagen con aplicaciones de visión artificial. Si el objetivo de referencia no se ilumina adecuadamente, se puede perder información y productividad. En la tala 2-2 están las técnicas para un sistema de visión artificial.

Tabla 2-2 Técnicas de iluminación.

Técnicas de iluminación	Definición	Imagen
<p>Iluminación frontal</p>	<p>Se posiciona la cámara de tal forma que este mirando al objeto y en la misma dirección que la luz, suaviza texturas y disminuye la influencia de rayas, la cámara recibe la luz reflejada del objeto, esta técnica de iluminación es perfecta para la detección de marcas de diferentes colores, detección de caracteres.</p>	
<p>Iluminación Lateral</p>	<p>Se posiciona la cámara de forma tal que este de frente al objeto mientras que la dirección de luz es lateral al objeto, el grado de inclinación del elemento emisor de luz será determinado por el grado que se desea resaltar los relieves, es perfecta para resaltar bordes, rayas y fisuras en una dirección determinada.</p>	
<p>Iluminación por campo oscuro</p>	<p>La luz es emitida lateralmente con un ángulo muy pequeño mediante un anillo en todas las direcciones rebotando en los defectos del objeto a ser analizado y teniendo una incidencia directa en la cámara, indicada para resaltar incrustaciones y códigos alfanuméricos, este tipo de iluminación es muy utilizada en la verificación de grabados tipo láser o troquel.</p>	

<p>Iluminación por contraste</p>	<p>En este tipo de técnica la luz es emitida desde la parte posterior del objeto quedando este entre la fuente de iluminación y la cámara. La iluminación tiene que ser uniforme en toda la superficie del objeto. La cámara inspecciona la silueta del objeto por contraste pudiendo realizar mediciones de alta precisión.</p> <p>Indicada para la inspección de la silueta del objeto. Utilizada también en materiales translúcidos o transparentes para detectar manchas, rayas, grietas.</p>	
<p>Iluminación Axial</p>	<p>La luz es emitida lateralmente siendo reflejada 90° por un espejo semitransparente que desvía los haces de luz en la misma dirección que el eje de la cámara consiguiendo una luz difusa homogénea. En superficies planas reflectantes si no se utiliza este método de iluminación, la cámara vería reflejado su propio objetivo.</p>	
<p>Iluminación difusa Tipo Domo</p>	<p>La luz es emitida dentro de una cúpula esférica resultando una luz difusa desde todas direcciones, eliminando sombras y reflejos, suavizando texturas y minimizando la influencia de rayas, polvo, relieves y curvaturas que pueda tener el objeto inspeccionado. Indicada para la inspección de superficies tales como: instrumental médico, espejos, compact disk.</p>	

Fuente (BcnVision, 2017)

2.6.5. Cámaras en sistemas de visión artificial.

Las cámaras son dispositivos que utilizando un juego de lentes (objetivos) reconstruyen una imagen sobre un elemento sensible (sensor), y la transmite al sistema de adquisición de la computadora (Vargas Víctor, 2010).

Las cámaras utilizadas en sistemas de visión artificial requieren de una serie de características que permitan el control del disparo de la cámara para capturar los objetos que pasan por delante de ella en la posición requerida.

- **Cámara Matricial.**

Este tipo de cámaras el sensor cubre un área que está formada por una matriz de píxeles, produce una imagen de un área, normalmente con una relación de aspecto de 4 a 3, los sensores de las cámaras matriciales están formados por miles de diodos fotosensibles posicionados de forma muy precisa en la matriz (Infaimon, 2019).



Figura 2-16. Cámara Matricial AP-3200-T

Fuente (Infaimon, 2019)

Tabla 2-3. Ventajas y desventajas de la Cámara Matricial.

Ventajas	Desventajas
<ul style="list-style-type: none">• Variedad de tamaños	<ul style="list-style-type: none">• Costo elevado
<ul style="list-style-type: none">• Gran estabilidad	<ul style="list-style-type: none">• Sensible a la luz
<ul style="list-style-type: none">• Cuenta con procesamiento de imagen	<ul style="list-style-type: none">• Baja calidad de imagen
	<ul style="list-style-type: none">• Tamaño de imagen dependiente de la matriz

Fuente (Infaimon, 2019)

- **Cámara Web.**

Una cámara Web Cam es un dispositivo de captura de imagen que se conecta al puerto USB del ordenador y que nos permite enviar imágenes a otro ordenador o bien simplemente visualizar en la pantalla una zona vigilada por dicha cámara (Serra José, 2018), mostrada en la figura 2-17.

Su circuitería es sencilla por lo que únicamente cuenta con un sensor y un lente para capturar la imagen de forma instantánea.



Figura 2-17. Web Cam HD Pro C920

Fuente : (Logitech, 2018)

Tabla 2-4 Ventajas y desventajas de una Web Cam HD Pro C920

Ventajas	Desventajas
<ul style="list-style-type: none"> • Tiene una corrección de iluminación automática 	<ul style="list-style-type: none"> • Resolución regular
<ul style="list-style-type: none"> • Tamaño reducido. 	<ul style="list-style-type: none"> • Software con errores.
<ul style="list-style-type: none"> • Adaptabilidad al medio 	<ul style="list-style-type: none"> • Limitación en el movimiento del lente
<ul style="list-style-type: none"> • Variedad de aplicaciones 	
<ul style="list-style-type: none"> • Costo moderado 	
<ul style="list-style-type: none"> • Captura fotos al instante 	

Fuente: (InformaticaModerna, 2014)

- **Cámara térmica**

Las cámaras térmicas son capaces de determinar la temperatura de los cuerpos a partir de su radiación infrarroja. Existen dos longitudes de onda por excelencia donde trabajan las cámaras térmicas de 2 a 5 μm y de 7 a 12 μm (Infaimon, 2019), mostrada en la figura 2-18.



Figura 2-18. Cámara Térmica Bobcat-640

Fuente: (Infaimon, 2019)

Tabla 2-5 Ventajas desventajas de una Cámara Térmica.

Ventajas	Desventajas
<ul style="list-style-type: none"> • Alta robustez 	<ul style="list-style-type: none"> • Manipulación compleja
<ul style="list-style-type: none"> • Análisis rápido 	<ul style="list-style-type: none"> • Velocidad de toma de imágenes regular
<ul style="list-style-type: none"> • Detecta problemas sin necesidad de detener la producción 	<ul style="list-style-type: none"> • Trabajan con software complejo
<ul style="list-style-type: none"> • Enfoque automático 	<ul style="list-style-type: none"> • Costo elevado
<ul style="list-style-type: none"> • Funcionan a temperatura ambiente 	<ul style="list-style-type: none"> • Trabajan con radiación
<ul style="list-style-type: none"> • Variedad de tamaños 	

Fuente: (Infaimon, 2019)

2.7. Redes Neuronales Artificiales.

Según (Simon, 1999) una red neuronal es un procesador distribuido masivamente paralelo compuesto por unidades simples de procesamiento, que tiene una tendencia natural a almacenar conocimiento experiencial y a hacerlo disponible para ser usado, esta se asemeja al cerebro en dos aspectos:

1. La red adquiere conocimiento del medio por medio de un proceso de aprendizaje.
2. Las fuerzas de conexión entre neuronas, denominados pesos sinápticos, se utilizan para almacenar el conocimiento adquirido.

Según (Matich Damián, 2001) define a las Redes Neuronales Artificiales como “Redes interconectadas masivamente en paralelo de elementos simples (Usualmente adaptativos) y con organización jerárquica las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema biológico”.

Los sistemas de redes neuronales son modelos computacionales que tratan de simular el funcionamiento del cerebro a partir del desarrollo de una arquitectura que toma rasgos del funcionamiento de este órgano sin llegar a desarrollar una réplica del mismo. (Galushkin Alexander, 2007)

El cerebro puede ser visto como un equipo integrado por aproximadamente 10 billones de elementos de procesamiento (neuronas) cuya velocidad de cálculo es lenta, pero que trabajan en paralelo y con este paralelismo logran alcanzar una alta potencia de procesamiento. (Constante, Chang, Pruna, & Escobar, 2016)

2.7.1. Ventajas que de las redes neuronales.

- **Aprendizaje adaptativo.**

La capacidad de aprendizaje adaptativo es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos.

- **Auto organización.**

Las redes neuronales emplean su capacidad de aprendizaje adaptativo para auto-organizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto-organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

- **Tolerancia a fallos.**

Las redes neuronales fueron los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales.

- **Operación a tiempo real.**

Las redes neuronales se adaptan bien a esto debido a su implementación paralela. Para que la mayoría de las redes puedan operar en un entorno de tiempo real, la necesidad de cambio en los pesos de las conexiones o entrenamiento es mínima.

2.7.2. Elementos básicos que constituyen una red neuronal.

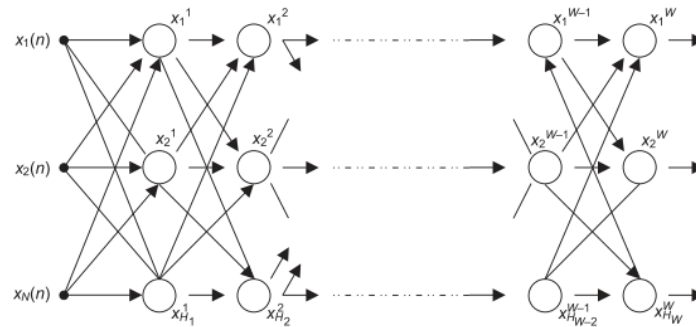


Figura 2-19. Arquitectura de una red neuronal.

Fuente:(Galushkin Alexander, 2007)

En la figura 2-19 se observa la arquitectura de una red neuronal la misma que está conformada por neuronas interconectadas y constituida por tres capas (dependiendo la arquitectura). Los datos que ingresan por la capa de entrada pasan a través de la capa oculta y salen por la capa de salida, vale mencionar que la capa oculta puede estar constituida por varias capas.

2.7.2.1. Función de entrada (Input function).

La neurona trata a muchos valores de entrada como si fueran uno solo; esto recibe el nombre de entrada global. Por lo tanto, ahora nos enfrentamos al problema de cómo se pueden combinar estas simples entradas (in_{i1}, in_{i2}, \dots) dentro de la entrada global, gin_i .

Esto se logra a través de la función de entrada, la cual se calcula a partir del vector entrada. La función de entrada puede describirse como sigue:

$$input_i = (in_{i1} \cdot w_{i1}) * (in_{i2} \cdot w_{i2}) * \dots (in_{in} \cdot w_{in})$$

Dónde: * representa al operador apropiado (por ejemplo: máximo, sumatoria, productoria, etc.), n al número de entradas a la neurona N_i y w_i al peso.

Los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona. Por consiguiente, los pesos que generalmente no están restringidos cambian la medida de influencia que tienen los valores de entrada. Es decir, que permiten que un gran valor de entrada tenga solamente una pequeña influencia, si estos son lo suficientemente pequeños (Matich Damián, 2001).

2.7.2.2. *Función de activación (activación function).*

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado (Matich Damián, 2001).

La función de activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, Θ_i) en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (-1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o -1) o activa (1).

2.7.2.3. *Función de salida (output function).*

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona i (out_i); la función de salida determina qué valor se transfiere a las neuronas vinculadas (Matich Damián, 2001).

Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$. También pueden ser binarios $\{0, 1\}$ o $\{-1, 1\}$.

2.7.3. *Redes neuronales Convolutivas.*

Las redes convolutivas son las redes neuronales artificiales que se utilizan habitualmente para resolver múltiples problemas prácticos que requieren procesar imágenes (Berzal, 2018).

Según (Berzal, 2018) las redes convolutivas resultan ser especialmente adecuadas para aprovechar la estructura bidimensional de los píxeles de una imagen.

En las imágenes, suelen ser importantes las relaciones de adyacencia entre píxeles y las redes convolutivas explotan este hecho para obtener resultados que no se podrían conseguir con una red multicapa tradicional (Berzal, 2018).

Pero no sólo son útiles para imágenes, sino que también se han empleado con éxito en aplicaciones que trabajan sobre otros tipos de señales, como los sistemas de reconocimiento de voz.

2.7.3.1. *Convolución*

Una convolución es un operador matemático que trabaja con dos argumentos (funciones), f y g , y las transforma en una tercera. La convolución en tiempo continuo se define como la integral del producto de dos funciones después de desplazar una de ellas una distancia t (Kim, 2013).

Sean las funciones f y g . Su convolución está denotada como:

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)dt \quad (2.2)$$

El intervalo de integración depende del dominio sobre el cual están definidas las funciones. Para el caso discreto se tiene que la integral se intercambia por la sumatoria, dando como resultado:

$$g = \sum_{-\infty}^{\infty} f(k)g(n - k) \quad (2.3)$$

Donde una de las funciones está desplazada una distancia n .

Las propiedades de los diferentes operadores de convolución son:

- **Conmutatividad.**

$$f * g = g * f \quad (2.4)$$

- **Asociatividad.**

$$f * (g * h) = (f * g) * h \quad (2.5)$$

- **Distributividad.**

$$f * (g + h) = f * g + f * h \quad (2.6)$$

- **Asociatividad multiplicado por un escalar.**

$$a(f * g) = (af) * g = f * (ag) \quad (2.7)$$

- **Regla de la derivación.**

$$D(f * g) = Df * g + fDg \quad (2.8)$$

Donde Df denota la derivada de f , o en el caso discreto el operador de diferencia.

- **Teorema de convolución.**

$$F(f * g) = (F(f)) \cdot (F(g)) \quad (2.9)$$

Donde F denota la Transformada de Fourier de f . Este teorema aplica también para la transformada de Laplace

- **Convolución con delta de Dirac.**

$$f(t) * \delta(t) = f(t) \quad (2.10)$$

$$f(t) * \delta(t - t_0) = f(t - t_0) \quad (2.11)$$

$$f(t - t_1) * \delta(t - t_0) = f(t - t_0 - t_1) \quad (2.11)$$

Por lo general es de ayuda ver al operador de convolución como un producto matricial.

La manera más simple de verlo es como una función de tipo ventana deslizante aplicada a una matriz(Kim, 2013).

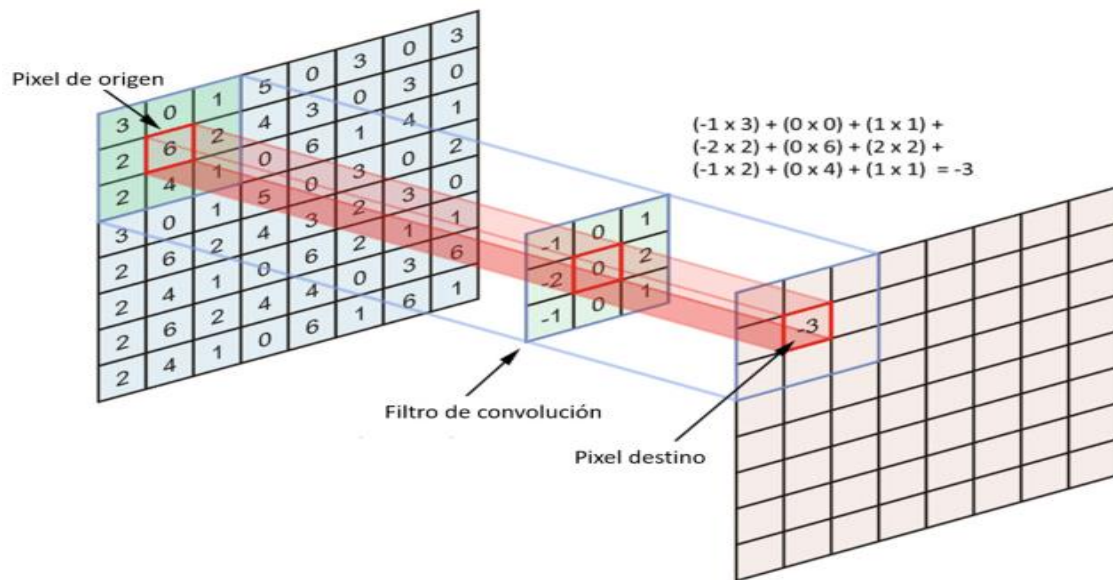


Figura 2-20. Aplicación de una convolución con matrices.

Fuente: (Kim, 2013) .

En la figura 2-20 se observa un ejemplo del operador de convolución aplicado sobre una matriz en forma de filtro. El resultado se obtiene multiplicando cada elemento del filtro de convolución con su correspondiente en la matriz de entrada para posteriormente sumarlos y así obtener el resultado del elemento de salida. Cabe mencionar que el tamaño del filtro toma importancia cuando se opera en las orillas de las matrices, ya que dependiendo de la distancia vertical u horizontal del centro del filtro a sus extremos es la cantidad de filas y columnas que desaparecerán en la matriz de salida(Kim, 2013).

2.7.3.2. Neocognitron.

Red neuronal artificial convolutiva multicapa jerárquica se utiliza para el reconocimiento de caracteres escritos a mano y reconocimiento de patrones, es una extensión natural de modelos en cascada consiste en múltiples tipos de células el más importante de los cuales se llama S-células y células C (Fukushima, 1980).

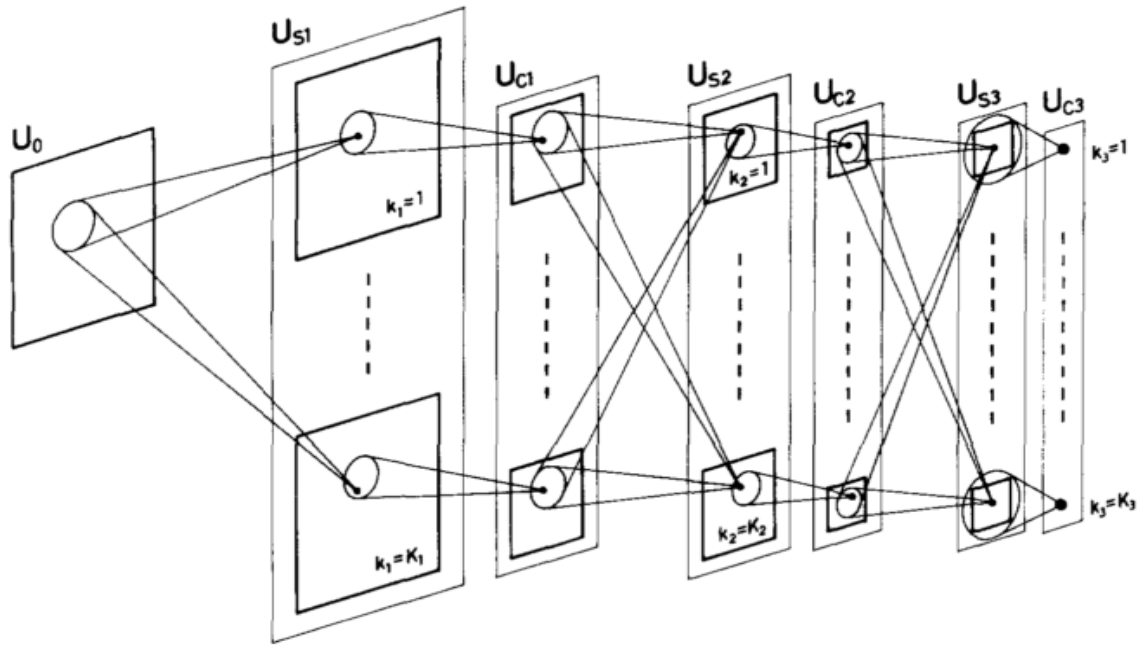


Figura 2-21. Diagrama esquemático que ilustra las interconexiones entre capas en el neocognitron.

Fuente: (Fukushima, 1980)

La red se autoorganiza mediante el "aprendizaje sin un maestro", adquiere la capacidad de reconocer patrones de estímulo basados en la similitud geométrica (Gestalt) de sus formas sin verse afectados por sus posiciones. Esta red recibe el sobrenombre de "neocognitron"(Fukushima, 1980)(Poli & Saito, 2006)

La red consta de una capa de entrada (matriz de foto receptores) seguida de una conexión en cascada de varias estructuras modulares, cada una de las cuales está compuesta por dos capas de células conectadas en una cascada. La primera capa de cada módulo consta de "células S", que muestran características similares a células simples o células hipercompletas de orden inferior, y la segunda capa consiste en "células C", similares a células complejas o hipercomplejos de orden superior. Las sinapsis aferentes de cada célula S tienen plasticidad y son modificables.(Fukushima, 1980; Gorokhovatskyi, 2016)

2.8. Open CV.

Por sus siglas en inglés (*Open Source Computer Vision Library*) es una biblioteca de software de aprendizaje automático y visión por computadora de código abierto.

OpenCV Esta librería se centra en el tratamiento de imágenes en tiempo real, luego es de gran utilidad para las aplicaciones en robots móviles, está escrita en su mayoría en C, esto la hace portátil a otras plataformas; y es un sistema multiplataforma (Windows, Linux...)(OpenCV, 2019)

Como el propósito de este estudio es la aplicación de la visión artificial se puede ahondar más en esta librería, que nos ofrece funciones para, como ya se ha dicho antes, trabajar con imágenes en tiempo real y además aporta también tipos de datos necesarios para implementar las funciones nombradas antes; algunas de las posibilidades que se nos presenta con esto es por ejemplo: reconocimiento de objetos, reconocimiento de rostro, seguimiento de movimiento.

2.9. Tensor Flow.

TensorFlow es una biblioteca de computación numérica que se basa en un sistema de redes neuronales, por lo que está orientada a problemas de *Deep Learning* en donde se la puede usar para reconocimiento de palabras, letras, textos e imágenes (Rungta Krishna, 2018).

La biblioteca de Tensorflow incorpora diferentes API (Programas de aplicación e interfaz) para construir a escala arquitecturas profundas de aprendizajes como CNN (Redes neuronales convolucionales) o RNN (Redes neuronales Recurrentes). TensorFlow se basa en un gráfico de cálculo que permite al desarrollador visualizar la construcción de la neuronal red (Rungta Krishna, 2018).

TensorFlow toma la entrada como una matriz multidimensional, también conocidos en forma de *tensores*, los cuales son valores primitivos que se ordenan dentro de una vector de 1 a N dimensiones.

La arquitectura Tensorflow funciona en tres partes:

- Pre procesando los datos
- Construye el modelo
- Entrenar y estimar el modelo

CAPÍTULO III

3. DISEÑO E IMPLEMENTACIÓN.

3.1. Criterios de selección de componentes electrónicos.

Luego de definir los conceptos con sus características, ventajas y desventajas detalladas en el estado del arte, se generan las matrices para seleccionar el mejor elemento para implementación del sistema de visión artificial. La ponderación va desde 1 hasta 5, siendo 5 el concepto que cumple con la característica deseable en el sistema, y 1 la ponderación que define al concepto como deficiente en dicha característica.

Tabla 3-1 Matriz de conceptos para la selección de la tarjeta de control

Criterios de selección	Peso	Beagle Bone		Arduino Uno		Raspberry Pi 4	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Velocidad de transmisión de datos	10%	4	0.4	3	0.3	4	0.4
Cantidad de pines	20%	4	0.8	5	1	4	0.8
Costos	20%	3	0.6	5	1	3	0.6
Programación	10%	3	0.3	5	0.5	4	0.4
Adaptabilidad con Qt	30%	4	1.2	5	1.5	4	1.2
Procesador	10%	5	0.5	3	0.3	5	0.5
	Total de puntos		3.8		4.6		3.9
	Lugar	3		1		2	
	Continuar?	No		Desarrollar		No	

Elaborado por: (Palma Wilson ,2019)

3.1.1. Selección de la tarjeta de control.

Se puede concluir que la mejor de tarjeta de control a implementarse en el sistema es una Arduino Uno, debido a su gran adaptabilidad con el software libre Qt Creator, por ser una plataforma de código abierto, además presenta la cantidad de pines suficientes para las entradas y salidas, su costo es reducido en comparación con otras tarjetas de adquisición de datos, además para poder realizar el control con el sensor y la comunicación con la PC es suficiente para la implementación del sistema, debido a la flexibilidad de sus librerías facilita la programación.

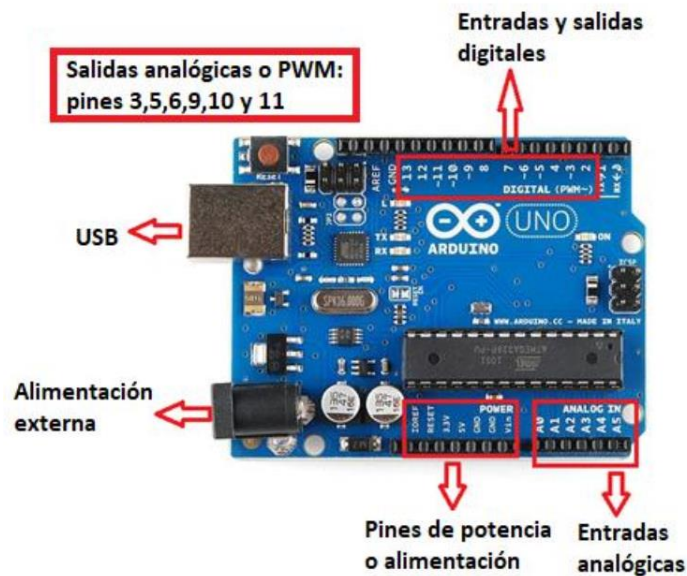


Figura 3-1. Tarjeta de control Arduino Uno

Fuente: (© 2020 Arduino, 2020)

Tabla 3-2 Especificaciones Técnicas Arduino Uno

Especificaciones técnicas de la tarjeta	Valor
Microcontrolador:	ATmega328
Voltaje Operativo	5v
Voltaje de Entrada (Recomendado):	7 – 12 v
Pines de Entradas/Salidas Digital:	14 (De las cuales 6 son salidas PWM)
Pines de Entradas Análogas:	6
Memoria Flash:	32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.
RAM:	2 KB (ATmega328)
EEPROM:	1 KB (ATmega328)
Velocidad del Reloj	16 MHZ

Fuente (© 2020 Arduino, 2020)

3.1.2. Selección de la cámara.

Se puede concluir que el mejor concepto del tipo de cámara a implementarse en el sistema es una cámara web, ya que su costo, adaptabilidad y facilidad de implementación son las mejores comparadas con otros sistemas, además su tamaño y resolución es buena en comparación de otras cámaras.

Para el sistema de visión artificial es necesario tener una buena resolución por lo cual se necesita una cámara con alta resolución, por lo cual se consideró la Web Cam HD Logitech Pro C920, como se muestra en la figura

Tabla 3-3 Matriz de conceptos para la selección de la cámara

Criterios de selección	Peso	Cámara Web		Cámara Térmica		Cámara Matricial	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Tamaño de la cámara	10%	4	0.4	3	0.3	3	0.3
Buena resolución	25%	4	1	4	1	3	0.75
Alta velocidad de toma de imágenes	25%	4	1	5	1.25	3	0.75
Adaptabilidad al medio	15%	5	0.75	4	0.6	1	0.15
Costos	15%	5	0.75	3	0.45	4	0.6
Fácil implementación	10%	5	0.5	2	0.2	4	0.4
	Total de puntos		4.4		3.8		2.95
	Lugar	1		2		3	
	¿Continuar?	Desarrollar		No		No	

Elaborado por: (Palma Wilson ,2019)



Figura 3-2. Web Cam HD Pro C920

Fuente : (Logitech, 2018)

Tabla 3-4. Especificaciones Técnicas Wed Cam HD Pro C920

WEB CAM HD PRO C920	Especificación
DIMENSIONES	<p>Altura x Anchura x Profundidad: 29 mm x 94 mm x 24 mm</p> <p>Longitud del cable: 1,5 m</p> <p>Altura x Anchura x Profundidad: 43,3 mm x 94 mm x 71 mm</p>
REQUISITOS DEL SISTEMA	<p>Compatible con: Windows® 10 o posterior, Windows 8, Windows 7</p> <p>Funciona en modo USB Video Device Class (UVC) con clientes de videoconferencia compatibles: macOS 10.10 o posterior</p> <p>Puerto USB</p>
ESPECIFICACIONES	<p>Resolución máx.: 1080p/30 fps - 720p/60 fps</p> <p>Tipo de enfoque: automático</p> <p>Micrófono integrado: estéreo</p>

Fuente (Logitech, 2018)

3.1.3. Selección del sensor de presencia.

Para la verificación de la presencia de latas de atún en la banda transportadora se optó por usar un sensor infrarrojo debido a las prestaciones ofrecidas y sobre todo al requerimiento del proyecto entonces el sensor seleccionado es el E18-D80NK.



Figura 3-3. Sensor de proximidad fotoeléctrico infrarrojo E18-D80NK

Fuente : (Lab-Volt (Quebec) Ltda., 2001)

Tabla 3-5 Características de un sensor de proximidad fotoeléctrico infrarrojo

Tensión de Operación	5V DC
Rango de Detección	3 a 80cm (ajustable)
Corriente de Salida Máxima	100mA
Salida	Tipo NPN Normalmente Abierto
Dimensiones	18 (diámetro) x 68 (largo) mm
Largo del cable	50 cm
Material	Polímero (Plástico)
Temperatura de Trabajo	-25 a 70°C

Fuente (Lab-Volt (Quebec) Ltda., 2001)

3.2. Diseño del sistema de visión artificial.

Para la ejecución del sistema de visión artificial en latas de atún de la empresa “Puertomar S.A, Provincia de Manabí” se determinó que la mejor técnica para la clasificación es una red neuronal convolucional, debido a que las entradas a la red neuronal son imágenes, lo que permite codificar ciertas propiedades; permitiendo reducir parámetros en la red lo que consecuentemente ocasiona mayor eficiencia y velocidad de procesamiento en el sistema, que es un parámetro fundamental al momento de implementar el sistema en el campo industrial.

Las características del hardware y software para la elaboración del sistema, así como los softwares necesarios para el entrenamiento de la red neuronal fueron:

- Sistema Operativo: Windows 10 profesional de 64 bits
- Procesador: i7 8va generación de 12 núcleos.
- Tarjeta gráfica: Nvidia GeForce GTX 1060 6GB DDR5
- Memoria: Disco de estado sólido nVme 256 GB

Programas necesarios:

- Python 3.7
- Tensorflow 1.14 para GPU
- Qt Creator 9.12 (Librerías Open cv 4.1.1-dev compiladas para GPU)
- Visual studio 15 y 16

En el Anexo A se detallan las características técnicas de los elementos utilizados en la elaboración del sistema de visión artificial, tales como: tarjeta de control, cámara web, sensor y actuador.

3.3. Esquema general del sistema de visión artificial.

El esquema empieza en el HMI (Human-Machine Interface) por sus siglas en inglés, donde se presiona el botón INICIAR, posteriormente el programa funciona continuamente, cuando el sensor infrarrojo detecte la presencia de una lata de atún en la banda transportadora, la cámara tomará una foto y empezará el pre procesamiento de la imagen a base de delimitación de bordes, eliminación del ruido, y extracción de capas, luego la red neuronal comparará la imagen captada con su data set, mediante algoritmos que se detalla posteriormente y entregará dos tipos de resultados: atunes con el codificado correcto y atunes con el codificado incorrecto, y cuando se detecte que las latas están sin

códigos, el sistema enviará una señal que activará una alarma con una baliza lumínica y sonora. Como se puede visualizar en la figura 3-4.

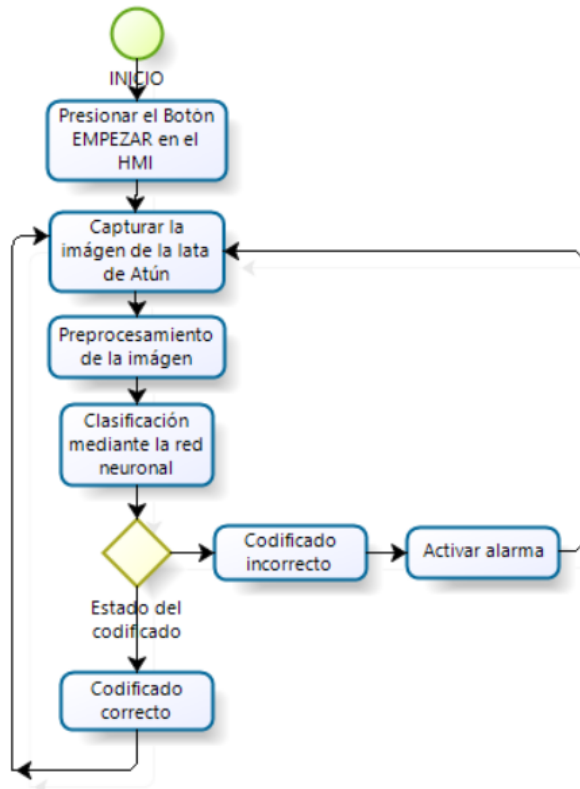


Figura 3-4. Esquema del sistema de visión artificial

Elaborado por: (Palma Wilson ,2019)

3.4. Implementación del sistema de visión artificial.

Para la implementación del sistema de visión artificial se utilizó el Software Qt Creator, el cual permite realizar toda la etapa de visión artificial desde la adquisición de la imagen, pre procesamiento, comparación mediante la red neuronal, comunicación con los sensores y actuadores y manipulación del sistema con el HMI, además de ser un software de baja complejidad y altas prestaciones al momento de editar el código tales como:

- Editor de código con soporte para C++, QML y ECMAscript
- Herramientas para la rápida navegación del código
- Resaltado de sintaxis y auto-completado de código
- Control estático de código y estilo a medida que se escribe
- Soporte para refactoring de código.
- Ayuda sensitiva al contexto.
- Plegado de código (code folding).
- Paréntesis coincidentes y modos de selección.

Para el entrenamiento de la red neuronal se utilizó la librería liberada por Google llamada TensorFlow, la cual comparada con otras librerías nos brinda mayor facilidad en su utilización, mejores tiempos de procesamiento y los mejores porcentajes de efectividad en la red neuronal, lo cual se detalla en el estado del arte anteriormente.

Los cómputos en TensorFlow se realizan con la ayuda de grafos dirigidos, donde los datos siguen un flujo determinado. Cada nodo que forma el grafo posee determinados inputs y outputs y representan una operación, la cual posee unos atributos necesarios para su ejecución. Las estructuras de los datos que van desde las salidas a las entradas se denominan tensores.

El procedimiento del sistema de visión artificial, se detalla en la figura3-5.

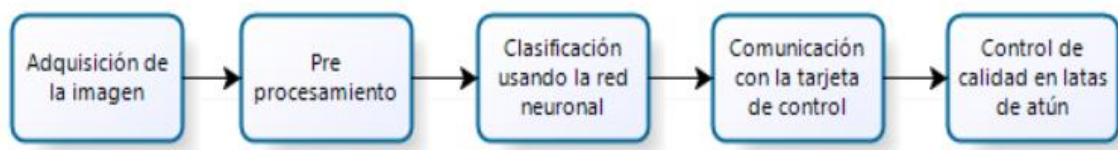


Figura 3-5. Procedimiento del sistema de visión artificial

Elaborado por: (Palma Wilson, 2019)

3.4.1. *Adquisición de imágenes.*

Para la adquisición de las imágenes se utilizó una cámara Logitech C920 y las librerías “QImage” y “QPixmap” del programa Qt Creator, en el HMI se presenta el video del proceso y con el comando “QImage” se genera una captura del fotograma cuando el sensor infrarrojo indique que existe la presencia de una lata de atún en la banda transportadora. Esta imagen es guardada en una dirección que se le especifique en el computador, para luego ser utilizada en el pre procesamiento de la misma.

Para la visualización de los videos e imágenes de análisis se utilizó el comando “QPixmap” que permite la visualización de las mismas en el HMI. Es necesario tener en cuenta las siguientes consideraciones al momento de adquirir las imágenes:

- **Resolución:** Tener en cuenta la resolución de la cámara debe ser la menor posible con las cual no se pierda nitidez, debido a que la imagen que recibe la red neuronal es de 224 x 224, y la resolución de la cámara que se utilizo es de 640 x 480, que en etapa de preprocesamiento se reduce a la mitad resultándonos en un factor de reducción de la imagen de 1.43.
- **Formato:** Tener en cuenta los formatos con los cuales se pueden almacenar las imágenes, por ejemplo, el formato JPG comprime la imagen, con el cual ganamos velocidad al

momento del procesamiento de la imagen, mientras que, si usásemos un formato diferente como, por ejemplo: TIF o PNG, que son formatos que no comprimen la imagen, tardaría más en procesar las imágenes la computadora, de esta forma se pierde velocidad al momento de procesar la imagen, antes de enviarla a la red neuronal.

3.5. Pre procesamiento.

Durante la etapa de preprocesamiento se encuentran varias subetapas como se muestra en la figura 3-6.

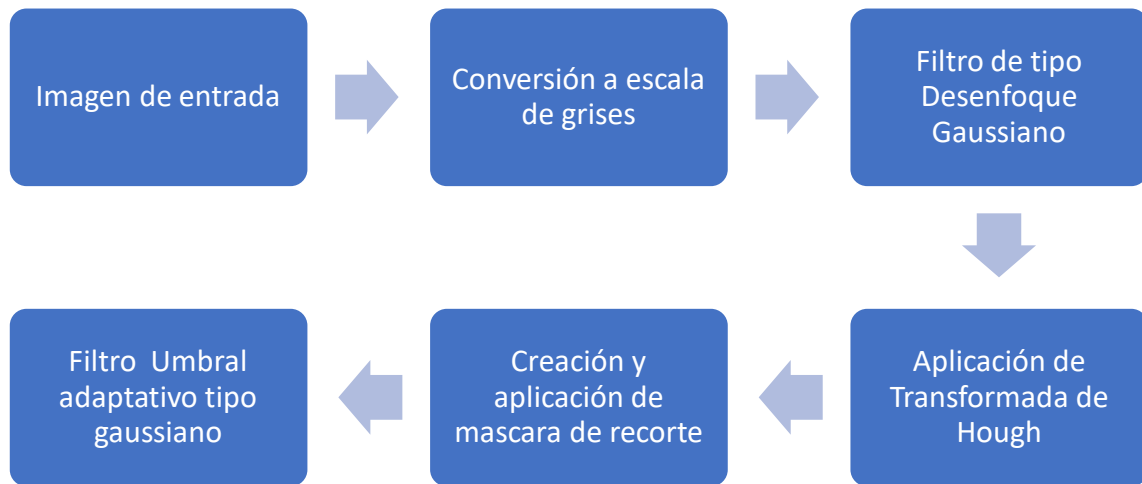


Figura 3-6. Etapas del pre procesamiento de imagen.

Elaborado por: (Palma Wilson ,2019)

3.5.1. Descripción de las etapas.

A continuación, se presenta una descripción de cada etapa del preprocesamiento de imagen además de una breve explicación del código utilizado que fue implementado en software QT Creator con librerías de OpenCV.

a. Imagen de entrada.

La imagen de entrada tiene una resolución de 640x480, tomada por la cámara como sensor de adquisición de imagen, se realiza un redimensionamiento a la imagen adquirida de modo que la resolución final es la mitad de la original (320x240), el tipo de escalamiento es lineal de modo que se pierde la menor cantidad de información, la imagen adquirida es de tipo RGB.

```
VideoCapture cap( 0 );
```

```
cap >> src;
flip(src,src,2);
cv::resize(src,src,Size(src.cols/2,src.rows/2),0);
```

Donde:

- cap es la variable utilizada para visualizar el video de la cámara, entre los paréntesis se escribe el número de la cámara que se desea utilizar, que depende de la cantidad de cámaras que se estén utilizando.
- flip es un comando que ayuda a girar la cámara y evitar el efecto espejo en el video.
- resize permite cambiar el tamaño de la imagen.

b. Conversión a escala de grises.

Para aplicar los filtros posteriores es necesaria una conversión a escala de grises, con esta conversión la imagen pasa de 3 canales (RGB) a ser mono canal (escala de grises), obteniendo 256 tonalidades de gris, no se ve alterada la resolución de la imagen de entrada.

```
cvtColor( src, src_gray, COLOR_BGR2GRAY );
```

Donde:

- src es la imagen de entrada.
- src_gray es la imagen de salida.
- COLOR_BGR2GRAY permite transformar los tres canales de la imagen inicial (red, green, blue) a un solo canal (escala de grises).

c. Filtro de tipo desenfoque gaussiano.

Este filtro permite reducir el ruido de la imagen a través de un desenfoque leve mediante el cual se reducen aquellos componentes que puedan afectar a las funciones de salidas posteriores o filtros, mejorando significativamente los resultados de salidas posteriores, es importante mencionar que el desenfoque gaussiano es una operación matemática entre matrices y que la matriz que se opera es de tamaño 9x9, el resultado se muestra en la figura 3-7.

```
GaussianBlur( src_gray, src_gray, Size(9,9), 1 );
```

Donde:

- src_gray es la imagen de entrada.
- src_gray es la imagen de salida.
- size(9x9) es el tamaño de la matriz gaussiana con la cual se va a generar la operación de eliminación del ruido.



Figura 3-7. Conversión a escala de grises y filtro de desenfoque gaussiano.

Elaborado por: (Palma Wilson, 2019)

d. Aplicación de la transformada de Hough.

Esta transformada se usa como técnica para la detección de figuras, en concreto se utilizó la transformada de Hough para circunferencias la cual permite detectar circunferencias en imágenes mono canal, como resultado se obtiene un vector de dimensión 3, en el cual se almacena la posición dentro de la imagen en coordenadas (x, y), además del radio de la circunferencia, es importante mencionar que la transformada de Hough solo permite detectar circunferencias perfectas, para lo cual es necesario que la cámara este alineada paralelamente con las latas, caso contrario no podrá ser detectada.

```
HoughCircles( src_gray, circles, HOUGH_GRADIENT, 1, src_gray.rows/8, 200, 100, 0, 0 );
```

Donde:

- src_gray en la matriz de entrada.
- circles es un vector de salida.
- HOUGH_GRADIENT es el método de generación de círculos por medio de gradientes.
- Los demás parámetros hacen referencia a los radios mínimos y máximos, así como parámetros de color necesarios para encontrar los círculos en la imagen.

e. Creación y aplicación de la máscara de recorte.

Se requiere crear una máscara de recorte con la cual se recortará la imagen que será enviada a la red neuronal, la máscara debe tener el mismo tamaño y tipo que la imagen adquirida por la cámara.

```
mask=cv::Mat::zeros(cv::Size(src.size()),src.type());
```

Luego de aplicada la máscara de recorte el resultado será una circunferencia, de esta forma se maximiza la cantidad de información que reciba la red neuronal. La máscara de recorte se crea a partir del resultado de la transformada de Hough para circunferencias, dibujando una circunferencia rellena de color blanco. Posteriormente se realiza la operación lógica AND entre la imagen de entrada y la máscara de recorte, el resultado será una imagen recortada en la que se muestra únicamente el código de la lata, como se muestra en la figura 3-8.

```
circle( mask, center, radius, Scalar(255,255,255), FILLED, 8, 0 );
```

```
bitwise_and(src,mask,resultado);
```

Donde:

- mask es una matriz generada de color negro, es importante que sea del mismo tamaño y tipo que la imagen inicial.
- center es la ubicación del centro de la circunferencia, en la Figura 5 se puede observar de color verde.
- radius es el radio a donde se va a dibujar la circunferencia, en la Figura 5 se puede observar de color rojo.
- FILLED ayuda a solo dibujar el contorno, si deseáramos rellenar la circunferencia, se debería modificar en este parámetro.

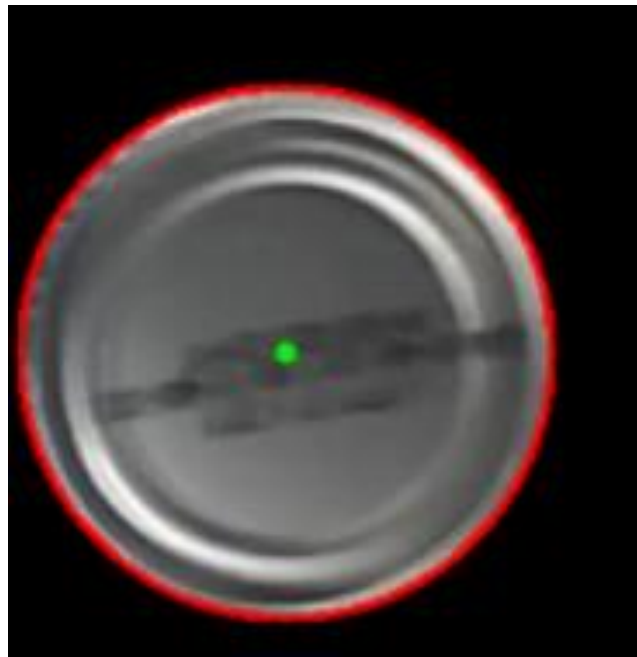


Figura 3-8. Resultado de la aplicación de la máscara de recorte.

Elaborado por: (Palma Wilson ,2019)

f. Filtro umbral adaptativo tipo gaussiano.

A través del filtro de umbral adaptativo se puede dar más connotación al código de la lata debido a que es un color diferente al del resto de la lata, el filtro de umbral adaptativo permite separar esto a través de un umbral de paso entre 0-255, el filtro de tipo gaussiano es más robusto y efectivo en cambios de luz como el reflejo de la lata, el resultado se muestra en la figura 3-9, en la que se observa que el código de la lata tiene un mayor realce frente al resto de la lata.

```
cvtColor(resultado,resultado_gray,COLOR_BGR2GRAY);
```

```
adaptiveThreshold(resultado_gray,gaus,255,ADAPTIVE_THRESH_GAUSSIAN_C,THRESH_BINARY,9,1);
```

Donde:

- ADAPTIVE_THRESH_GAUSSIAN es un método para filtrar los cambios de luz
- THRESH_BINARY es el valor del umbral que utiliza el código para separar los píxeles negros de los blancos.



Figura 3-9. Resultado de aplicación filtro umbral adaptativo tipo gaussiano.

Elaborado por: (Palma Wilson ,2019)

Finalmente, la imagen que recibe la red neuronal es una imagen cuadrada debido a que la máscara de recorte aplicada es circular.

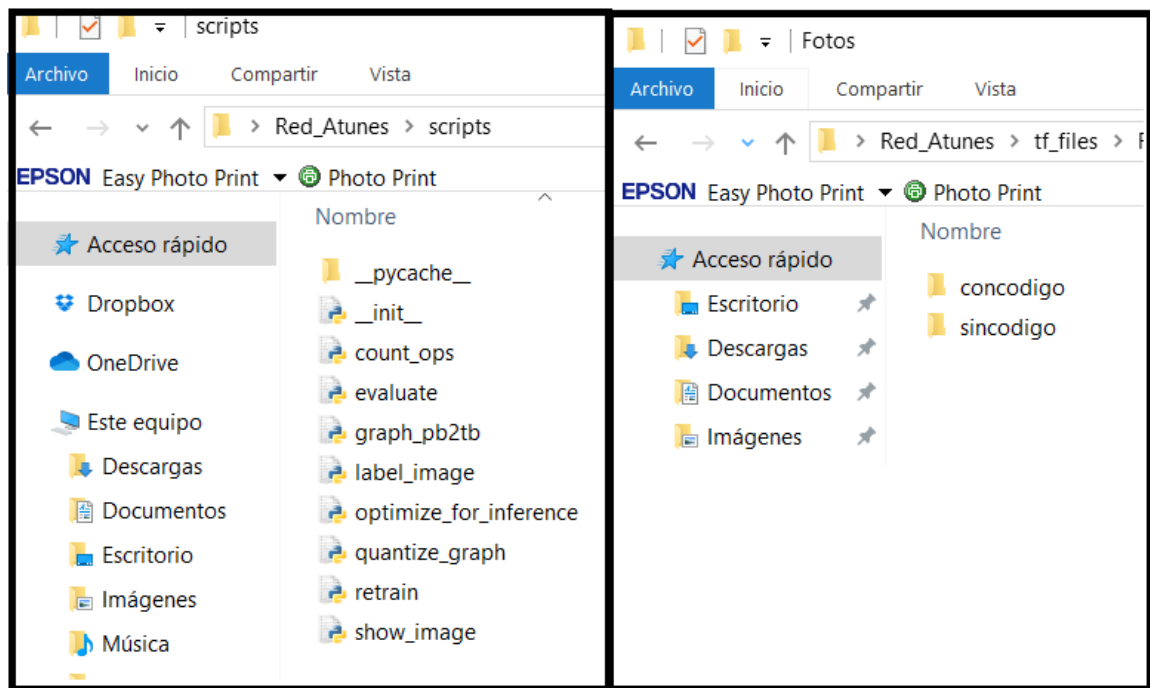
3.6. Red neuronal

3.6.1. Creación y entrenamiento de la red neuronal.

En la implementación del sistema se utilizó una red neuronal convolucional, que fue entrenada con la ayuda de la librería de TensorFlow, la cual es una de las librerías más populares en el ámbito de las redes neuronales y la visión artificial, con un 97.53% de popularidad, dicha librería tiene una

gran capacidad de manejo de redes neuronales convolucionales (CNN) y recurrentes (RNN), en el presente proyecto se utilizara un modelo convolucional que sobresalió de entre otros, por su gran velocidad de procesamiento y gran efectividad.

Para la creación y entrenamiento de la red neuronal fue necesario tener instalados algunos programas y librerías que se detallan en el punto 3.2, luego de tener instalado se procede a la creación de una carpeta en donde se van a colocar los archivos que se vayan generando. Dentro de esta carpeta deben estar 2 carpetas: la primera con las librerías necesarias para la creación y entrenamiento de la red neuronal, y en la segunda deben estar las fotos en una carpeta de las cuales se pretenden clasificar, en este caso se tienen dos carpetas con las imágenes con los atunes con códigos y las imágenes de los atunes sin código, como se puede apreciar en la Figura 3-7, en la carpeta tf_files se debe crear una carpeta en donde se ubicará una carpeta con las imágenes de entrenamiento y el modelo (el modelo puede descargarse y colocarse dentro de una carpeta).



a)

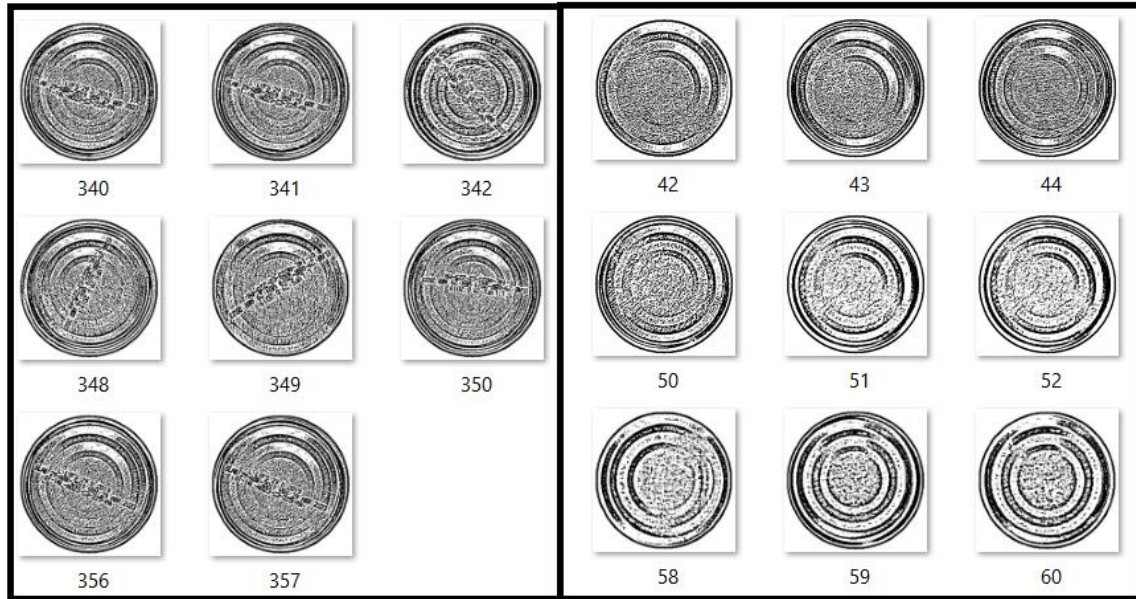
b)

Figura 3-10. a) Librerías de Python dentro de una carpeta scripts b) Carpetas con las clases que se quieren clasificar.

Elaborado por: (Palma Wilson ,2019)

El dataset (imágenes para el entrenamiento) que se tenga para el entrenamiento es muy importante dentro del entrenamiento de la red neuronal, ya que entre más imágenes se tengan la red neuronal tendrá más características para poder diferenciar una clase de la otra, estos detalles tienen que ver con la iluminación, bordes, contrastes, entre otras características propias de las imágenes. Para el entrenamiento se utilizaron un dataset de 357 imágenes con códigos y 60 imágenes sin códigos, con

un total de 417 imágenes, algunas de las muestras utilizadas se pueden visualizar en la figura 3-11, cabe recalcar que las imágenes utilizadas para el entrenamiento son las imágenes ya procesadas mediante el tratamiento de imagen mencionado en el apartado anterior. Además, es importante detallar que las imágenes deben tener un formato JPG y que deben estar numeradas.



a)

b)

Figura 3-11. a) Dataset de imágenes de latas con código **b)** Dataset de imágenes de latas sin código
Elaborado por:(Palma Wilson ,2019)

Inicialmente es necesario entrar al CoMmanD (cmd) del computador y direccionar a la carpeta anterior, con el comando: `cd C:/Users/Dell/Desktop/Red_Atunes`, con el modelo obtenido y colocado en la carpeta, se procede a generar el código de entrenamiento donde el parámetro más relevante es el número de pasos con los cuales se quiere entrenar (steps), el cual fue de 4000, con el que se obtuvo un resultado óptimo.

El número de pasos depende de la capacidad computacional que se tenga, además de la eficiencia deseada, dicho código se puede observar en la Figura 3-9 , también cabe recalcar que el modelo usado es una “Mobilenet_1.0_224”, es decir un modelo convolucional de 224 pixeles de análisis en su primera versión.


```

Microsoft Windows [Versión 10.0.17134.950]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dell>cd C:/Users/Dell/Desktop/Red_Atunes

C:\Users\Dell\Desktop\Red_Atunes>set IMAGE_SIZE=224

C:\Users\Dell\Desktop\Red_Atunes>set ARCHITECTURE="mobilenet_1.0_224"

C:\Users\Dell\Desktop\Red_Atunes>python C:/Users/Dell/Desktop/Red_Atunes/scripts/retrain.py --bottleneck_dir=tf_files/bottlenecks --how_many_training_steps=4000 --model_dir=tf_files/models/ --summaries_dir=tf_files/training_summaries/"%ARCHITECTURE%" --output_graph=tf_files/retrained_graph.pb --output_labels=tf_files/retrained_labels.txt --architecture="%ARCHITECTURE%" --image_dir=tf_files/Fotos

```

Figura 3-12: Código para el entrenamiento de la red neuronal.

Elaborado por: (Palma Wilson ,2019)

Una vez que generados los 4000 pasos, se obtuvieron los siguientes resultados:

- Eficiencia general de la red neuronal 97.2%
- Tiempo promedio de análisis 0.3 ms

Lo que se puede visualizar en la figura 3-13

```

INFO:tensorflow:2019-09-01 21:18:31.885769: Step 3980: Validation accuracy = 99.0% (N=100)
INFO:tensorflow:2019-09-01 21:18:32.732505: Step 3990: Train accuracy = 100.0%
INFO:tensorflow:2019-09-01 21:18:32.732505: Step 3990: Cross entropy = 0.000191
INFO:tensorflow:2019-09-01 21:18:32.815284: Step 3990: Validation accuracy = 100.0% (N=100)
INFO:tensorflow:2019-09-01 21:18:33.581236: Step 3999: Train accuracy = 100.0%
INFO:tensorflow:2019-09-01 21:18:33.581236: Step 3999: Cross entropy = 0.000303
INFO:tensorflow:2019-09-01 21:18:33.662022: Step 3999: Validation accuracy = 97.0% (N=100)
INFO:tensorflow:Final test accuracy = 97.2% (N=72)
INFO:tensorflow:Froze 2 variables.
INFO:tensorflow:Converted 2 variables to const ops.

```

Figura 3-13. Resultados finales de la red neuronal

Elaborado por:(Palma Wilson ,2019)

Luego de haber realizado el entrenamiento, se procede a realizar una optimización con el fin de mejorar los tiempos y la eficiencia, y paso siguiente una transformación del archivo a un ejecutable para el software OpenCV. Luego de haber realizado estos pasos se tendrá en la carpeta inicial los archivos que se observa en la Figura 3-11, archivos como los cuellos de botellas generados, los retrained_labels (nombres de las clases entrenadas “concodigo” o “sincodigo”) y el archivo más importante “final_graphopencv3”, el cual es el que se va a utilizar en el punto siguiente para el entrenamiento.

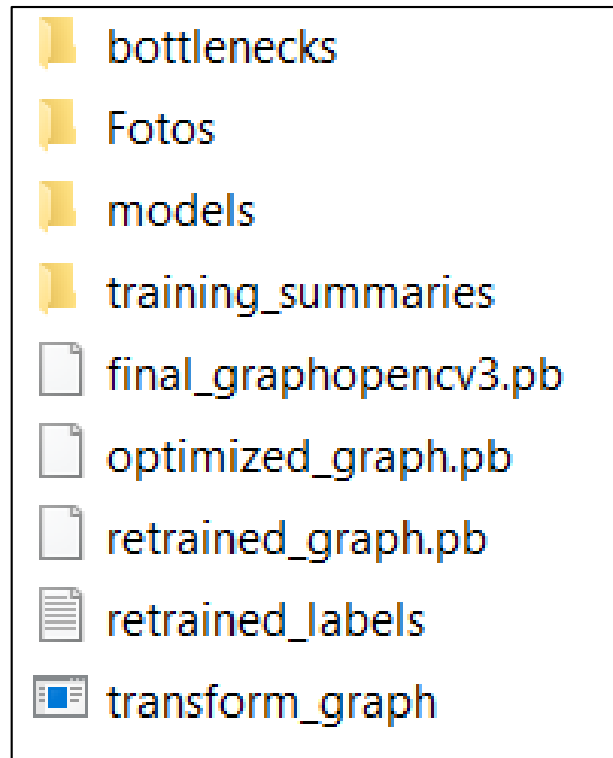


Figura 3-14. Archivos generados luego de la creación de la red neuronal.

Elaborado por: (Palma Wilson ,2019)

a. *Clasificación.*

Para utilizar la red neuronal creada es necesario crear dos vectores, el primero es para ingresar las clases con las cuales se va a clasificar la red, y el otro para setear los parámetros. Es muy importante direccionar muy bien la carpeta y el archivo que contiene los nombres de las clases, además tener instalado las librerías para usar las “CNN”, debido a que la falta de estas librerías ocasiona errores al momento de compilar el programa.

En el programa principal se presenta el video generado por la cámara y se inicializa el puerto por donde se va a leer la señal del sensor, sin paridad y con una velocidad de 9600 baudios, el código para leer la señal que viene del puerto serial es:

```
ReadFile(hSerial, VarSall, 1, &btsIO, NULL);
```

Dónde:

hSerial lee el puerto serial donde los parámetros inicializados son los siguientes:

- DCB ParSerial;
- GetCommState(hSerial,&ParSerial);
- ParSerial.BaudRate = CBR_9600;
- ParSerial.ByteSize = 8;

- ParSerial.Parity = NOPARITY;
- ParSerial.StopBits = ONESTOPBIT;
- SetCommState(hSerial, &ParSerial);

En donde se crea la variable VarSal1 que indica el estado del sensor, debido a que el sensor utilizado es un sensor infrarrojo, tendremos dos señales que se enviarán por el puerto serial, “a” cuando el sensor este activado y “b” cuando el sensor está inactivo, esto se realiza mediante condicionales “if” como se muestra en el siguiente código:

```
if (VarSal1[0] == 'a') {
    ui->label_12->setStyleSheet("background-color: rgb(0,255,100)");
    sensorindicador=1;
}

if (VarSal1[0] == 'b') {
    ui->label_12->setStyleSheet("background-color: rgb(255,100,0)");
    sensorindicador=0;
    auxstart=1;
}
```

La señal de la variable “sensorindicador”, permitirá el paso al siguiente condicional, donde se captura el frame de la cámara, para luego ser mandada a analizar con la red neuronal, mediante el modelo previamente ingresado y la utilización del vector de parámetros de clasificación antes creado. El vector nos entregará 3 parámetros: clase, eficiencia y tiempo de análisis respectivamente como se muestra en el siguiente código, es importante transformar estos parámetros al tipo “String” para ser utilizados más adelante en la Interfaz Gráfica (GUI).

```
std::string valor1 = parametros[0];
std::string valor2 = parametros[1];
std::string valor3 = parametros[2];
```

Finalmente se compara dichos parámetros nuevamente con condicionales “if” para generar las diferentes acciones, por ejemplo:

- Si valor 1 es igual a “concodigo”, el programa no realizará ninguna acción, ya que la lata de atún se encuentra correctamente codificada y no es necesario realizar alguna acción, sino que el programa volverá a esperar la señal del sensor para volver a analizar la siguiente lata de atún.
- Si valor 1 es igual a “sincodigo”, el programa enviará una señal por el puerto serial para que la tarjeta de control active una baliza de alerta lumínica y sonora. Luego de ello el programa volverá a esperar la señal del sensor para volver a analizar la siguiente lata de atún, lo ideal sería detener la línea de producción para que el operario puede verificar el

estado de la lata, y evitar que más latas salgan sin el código correcto, pero debido a las limitaciones de la empresa, no se pudo manipular el sistema.

Es muy importante la creación de una variable del tipo “QTimer” para visualizar los frames captados por la cámara en el HMI, lo cual se consiguió mediante las siguientes líneas de código, teniendo en cuenta el menor tiempo de captura que es de 1 milisegundo.

```
timer = new QTimer(this);
connect(timer,SIGNAL(timeout()),this,SLOT(comunicacion()));
timer->start(1);
```

3.7. Programación de la tarjeta de control

Para comunicar la tarjeta de control con el HMI, (Human-Machine Interface) por sus siglas en inglés, se hizo uso de una variable tipo carácter “incomingByte”, y dependiendo del estado de la variable “lectura” se envían las señales de activación o inactivación del sensor infrarrojo, también mediante la tarjeta se envía una señal para activar el relé y que el modulo temporizador funcione dependiendo del tiempo establecido.

```
void loop() {
  int lectura=digitalRead(8);
  if(lectura==LOW){
    Serial.write('a');
  }
  delay(75);
  if(lectura==HIGH){
    Serial.write('b');
  }
  delay(75);
}
```

Se debe de tener en cuenta los tiempos de espera tanto para el sensor, como para la transmisión de los datos por el puerto serial, ya que si no se espera lo suficiente antes de volver a leer un dato, van a existir choques entre las señales de entrada-salida.

Para enviar la señal de activación de la baliza se utilizó un condicional “switch”

```

if (Serial.available() > 0) {
// Leer los datos de entrada "incoming byte"

incomingByte = Serial.read();

switch(incomingByte) {
case '1':
digitalWrite(pinBaliza,HIGH);
delay(200);
digitalWrite(pinBaliza,LOW);
}
}

```

3.8. GUI (Interfaz Gráfica de Usuario)

La interfaz gráfica (GUI) ayuda a la manipulación del sistema del usuario por medio de una pantalla de control, esta debe ser lo más amigable posible para facilitar el trabajo del usuario, para ello se hizo uso del software *Qt Creator*, que facilita el uso del lenguaje de programación C++ y a su vez la interacción con la red neuronal creada.

Para la creación del GUI, iniciamos creando un proyecto *Qt Widgets Application* y usando el Tool Kit Qt 5.9.1 MSVC2017 64bits como se observa en la Figura 3-15, solo se recomienda no usar el MinGW debido a incompatibilidades con las librerías de la red neuronal.

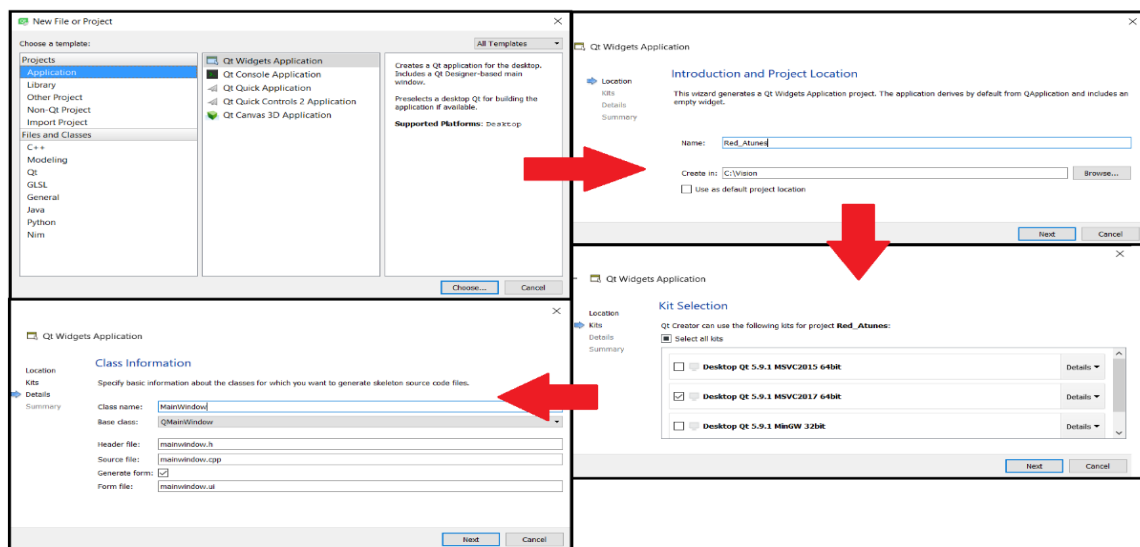


Figura 3-15. Creación del entorno de la GUI.

Elaborado por: (Palma Wilson ,2019)

Luego de tener el entorno donde se va a realizar el GUI, es necesario conocer donde se encuentran las herramientas con las que se van a diseñar la GUI, lo cual se detalla en la figura 3 – 16.

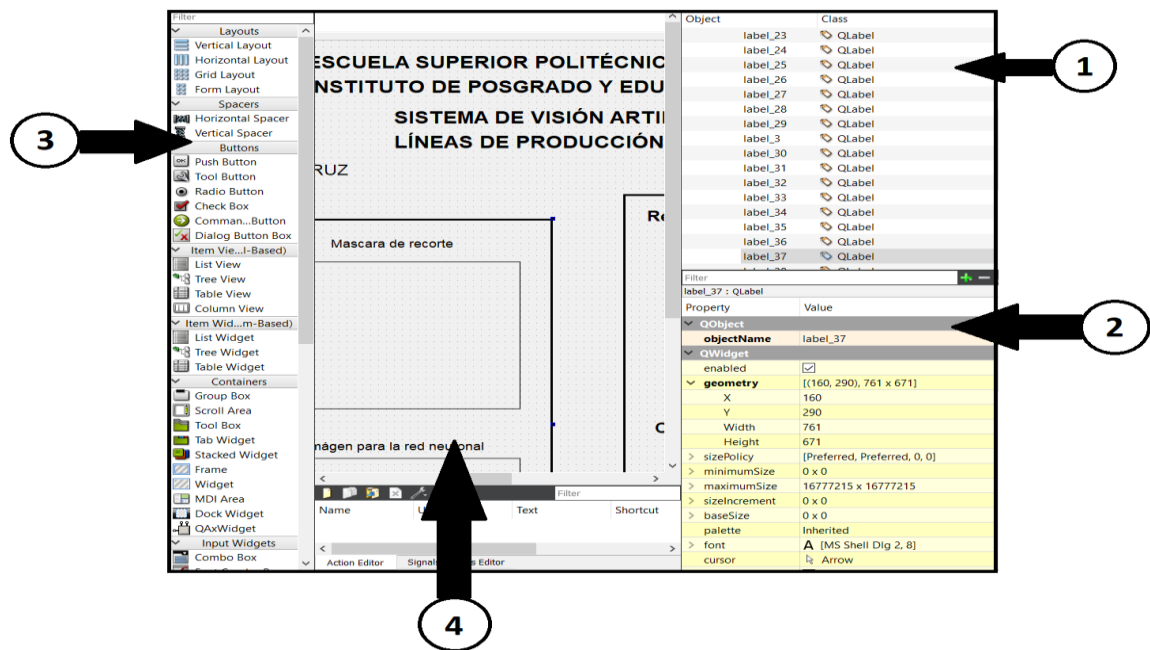


Figura 3-16. Estructura del panel de diseño del GUI de QtCreator.

Elaborado por: (Palma Wilson ,2019)

1. Estructura del panel de diseño del GUI de QtCreator Panel inspector aquí aparecen los objetos creados junto a ellos su nombre de identificación para la programación de la misma. Cabe mencionar que los objetos parecerán en forma de árbol dependiendo si los elementos estén unos dentro de otros.
2. Panel de propiedades: Si se hace click en algún elemento, aquí parecerán las diferentes propiedades del objeto.
3. Panel de componentes: Lista de objetos o elementos que se pueden usar en el GUI (botones, diálogos, paneles, combobox, etc)
4. Panel de edición: Lugar donde se colocan y distribuyen los elementos.

Luego direccionar y ubicar los objetos y programar cada uno de ellos, se obtiene la siguiente GUI, como se muestra en la Figura 3 -14 y se detallan a continuación.



Figura 3-17. Partes que conforma el GUI.

Elaborado por: (Palma Wilson ,2019)

A continuación, se detallan las acciones que realizan los diferentes *labels* utilizados en la GUI.

1. Video de las latas de atún que está observando la cámara en el sistema de visión artificial.
2. Máscara de recorte generada, cuando detecta un objeto circular dentro del video.
3. Imagen con el pre procesamiento detallada en el punto 3.5.1.
4. Imagen que se guarda para el análisis en la red neuronal.
5. Estado del análisis de la lata de atún. Puede ser CORRECTA para cuando existe un correcto etiquetado, mientras que se mostrará INCORRECTO cuando no exista un etiquetado en la lata de atún.
6. Muestra la eficiencia de la red neuronal dependiendo de la aceptación que muestra la red, con cada una de las clases que compara.
7. Muestra el tiempo de análisis de la red neuronal con cada imagen captada.
8. Indicador del sensor infrarrojo, mostrará el estado activo o inactivo del sensor, con color verde para activo y color rojo para inactivo.
9. Contador de latas de atún CORRECTAS analizadas.
10. Contador de latas de atún INCORRECTAS analizadas.
11. Contador total de latas de atún

Continuamente se detallan las acciones que realizan los botones generados en la GUI.

12. Botón INICIAR: Permite empezar el ciclo de análisis del sistema de control de calidad en las latas de atún.
13. Botón PARAR: Permite detener por un momento el sistema de visión artificial.
14. Botón SALIR: Permite salir del HMI.

3.9. Conexión del sistema de visión artificial.

El sistema cuenta con un sensor infrarrojo conectado al PIN digital 8 de la tarjeta de control y alimentado con 5V, también tiene un módulo relé temporizador, que ayuda al sistema a generar por un determinado tiempo la alerta sonora, dependiendo de las condiciones que se desee, el tiempo puede variar de 0.1 – 99.9 segundos; 1 – 999 segundos y 1 – 999 minutos. Tiene un margen de error de 0.01% por segundo.

Debido a las bajas condiciones de luz presentes en la empresa, se vio necesario implementar un sistema de iluminación de luz lateral, con el cual se resaltan los bordes y las rayas presentes en el codificado de los atunes, el sistema de iluminación cuenta con dos lámparas led como se muestra en la Figura 3-15.

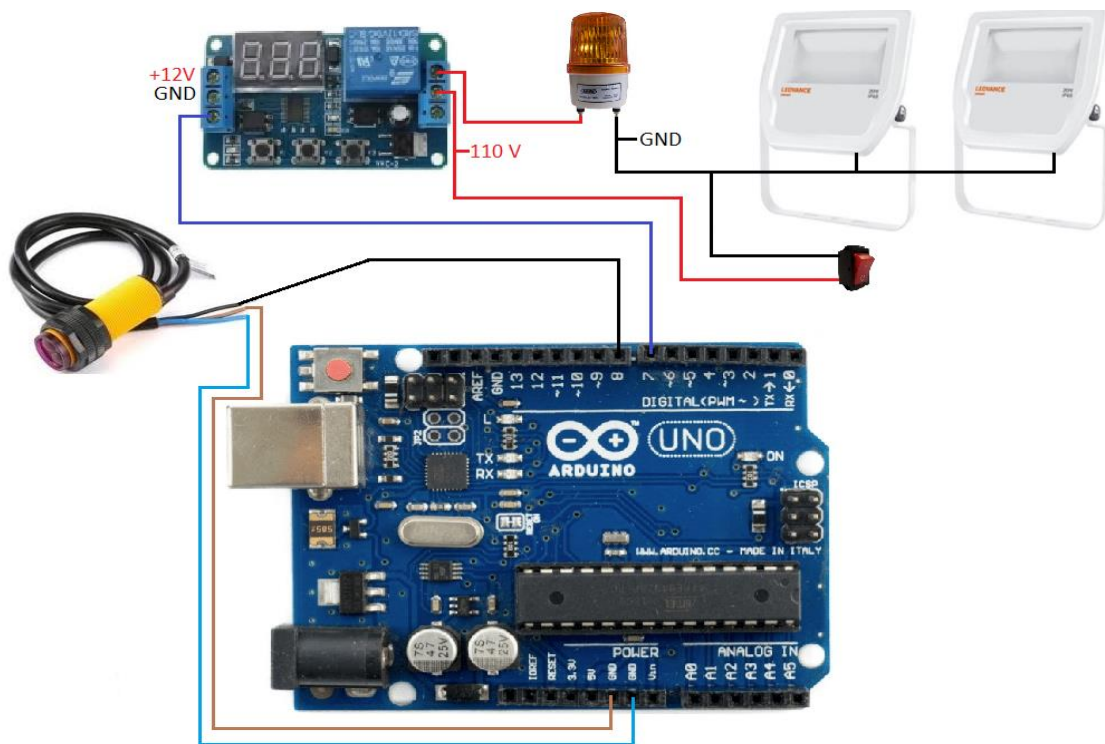


Figura 3-18. Conexión del sistema de visión artificial

Elaborado por: (Palma Wilson ,2019)

Implementar un sistema de para la detección de fallas mediante visión artificial en el codificado de las latas de atún , requiere de la integración de los sistemas de hardware y software.

En la parte el hardware encontramos la cámara, el procesador de datos (PC), la banda transportadora y el mecanismo de selección (ver figura 3-19), la parte de software constituida por un algoritmo de procesamiento de imágenes y una red neuronal convolucional que es la encargada

de la toma de decisión, en la cual se incluye los parámetros de selección de codificado en las latas de atún.

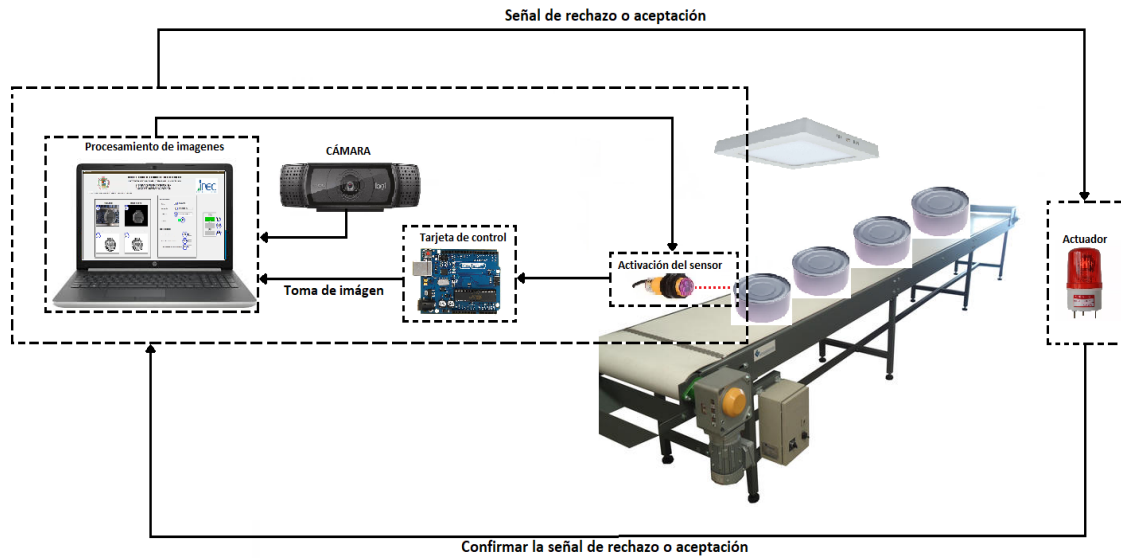


Figura 3-19. Descripción del proceso de visión artificial

Elaborado por: (Palma Wilson ,2019)

CAPÍTULO IV

4. RESULTADOS EXPERIMENTALES Y DISCUSIÓN.

4.1. Clasificación de la red neuronal.

Para análisis experimental se consideró en primera instancia del tamaño de la muestra, en la empresa Puerto Mar, para lo cual se toma en cuenta la siguiente ecuación.

$$n = \frac{Z_{\alpha}^2 * p * q}{d^2} \quad (4.1)$$

Donde:

n tamaño de la muestra que se tomara en cuenta

z Nivel de confianza de la distribución normal

p Proporción esperada

q 1-p



d precisión













$$n = \frac{1.96^2 * 0.01 * 0.99}{0.028^2}$$


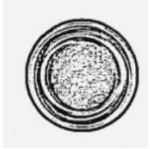
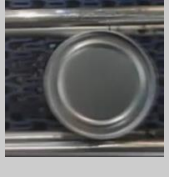













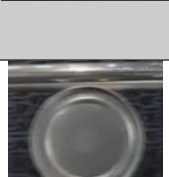
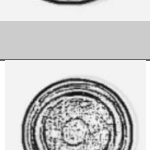
$$n = 48.5 \quad n = 49$$





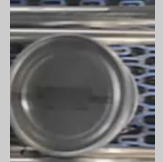





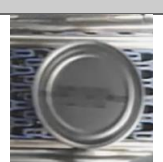







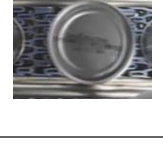

A continuación, se muestra la tabla 4-1 de pruebas en la cual se detalla la imagen obtenida por la cámara y la imagen procesada que es enviada hacia la red neuronal, además del resultado obtenido por la red neuronal, estos datos posteriormente serán usados para la obtención del porcentaje de efectividad de la red neuronal.

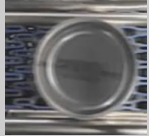












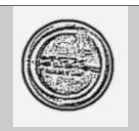








Tabla 4-1. Clasificación de la red neuronal de las latas de atún según la presencia o ausencia del codificado.













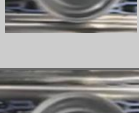



N° Ensayo	Imagen Cámara	Imagen Procesada	Acierto Positivo	Acierto Negativo
1			SI	

2			SI	
3			SI	
4			SI	
5			SI	
6			SI	
7				SI
8				SI
9			SI	
10			SI	
11			SI	

12			SI	
13			SI	
14			SI	
15			SI	
16			SI	
17			SI	
18			SI	
19			SI	
20			SI	

21			SI	
22			SI	
23			SI	
24			SI	
25			SI	
26			SI	
27			SI	
28			SI	
29			SI	
30			SI	

31			SI	
32			SI	
33				SI
34			SI	
35				SI
36				SI
37			SI	
38			SI	
39			SI	
40				SI
41			SI	

42			SI	
43			SI	
44			SI	
45				SI
46				SI
47			SI	
48			SI	
49			SI	

Elaborado por: (Palma Wilson ,2019)

De la tabla 4-1 se obtienen los siguientes resultados, mostrados en la tabla 4-2.

Tabla 4-2. Resultados de las pruebas realizadas

N° de muestras	Aciertos positivos	Aciertos negativos
49	41	8

Elaborado por: (Palma Wilson ,2019)

Con los datos obtenidos en la Tabla 4-2 se puede obtener el porcentaje de efectividad de la red neuronal, para lo cual se realiza el cálculo:

$$\% \text{ efectividad} = \frac{\text{Aciertos Positivos}}{\text{N}^\circ \text{ de muestras}} * 100\%$$

$$\% \text{ efectividad} = \frac{41}{49} * 100\%$$

$$\% \text{ efectividad} = 83.67\%$$

4.2. Validación de la hipótesis.

En la línea de producción de codificado de las latas de atún de la empresa PuertoMar S.A., ubicada en la provincia de Manabí se toma como dato principal la velocidad de la línea de codificado en cual arroja que se codifican alrededor de 200 latas por cada minuto, los contenedores de transporte de cajas de latas de atún son de 20 a 40 pies, la planta llena 1920 cajas para contenedores de 20 pies, dichas cantidades son transportadas vía marítima y también por vía terrestre alcanzando envíos a los diferentes lugares de exportación y distribución.

De acuerdo a la investigación de campo realizada se obtiene que el costo de cada lata es de 0.72 ctvs, entonces se puede calcular el costo generado cada minuto en la empresa.

$$\text{Ganancia por min} = \text{Costo de cada lata} * \text{Número de latas por min}$$

$$\text{Ganancia por min} = 0.72 \text{ ctvs} * 200$$

$$\text{Ganancia por min} = \$144$$

En la tabla 4-3 se muestran las pérdidas que se generarían sin un sistema de visión artificial que alerte al operario al momento de presentarse alguna falla en el codificado de las latas de atún.

Tabla 4-3. Pérdidas totales de la empresa por fallos en la codificación.

Tiempo de falla en la línea de producción	Total de latas con fallas	Pérdidas económicas (dólares)
1 min	200	144
5 min	1000	720
10 min	2000	1440

Elaborado por: (Palma Wilson ,2019)

En la Tabla 4-4 se muestran las pérdidas que se generarían luego de implementar el sistema de visión artificial en la línea de producción, para lo cual se calcula el número de latas mínimas que necesitaría la red neuronal creada para identificar con acertación el fallo dentro del sistema, mediante la efectividad de la red neuronal y un cálculo probabilístico del sistema.

Para calcular el número mínimo de latas de atún que necesita la red neuronal para acertar con eficiencia, se hace uso de la Función de Laplace para probabilidades.

$$P = \frac{\# \text{ Casos favorables}}{\# \text{ Casos totales}}$$

$$P = \frac{(100\% - 84\%)(200)}{200}$$

$$P = \frac{(16\%)(200)}{200}$$

$$P = \frac{32}{200}$$

$$P = 0.16$$

Tabla 4-4. Pérdidas totales de la empresa por fallos en la codificación luego de implementar el sistema de visión artificial

Tiempo de falla en la línea de producción	Total de latas con fallas	Red neuronal Eficiencia 84%	Pérdidas económicas (dólares)
1 min	200	0.16	23.04
5 min	1000	0.16	115.2
10 min	2000	0.16	230.4

Elaborado por: (Palma Wilson ,2019)

Debido a los diferentes factores en la planta que alteraron la eficiencia en la red neuronal no se pudo llegar a una eficiencia del 100%. Estos inconvenientes no permitieron la correcta toma de imágenes que serán enviadas a la red neuronal.

- Vibraciones en la estructura de la banda transportadora que alteraban el enfoque.
- Altas temperaturas y humedad en el ambiente que empañaban la lente de la cámara
- Condiciones de luz variables, las cuales no permitían tener un brillo uniforme en todo el sistema.

- El cambio manual en la estructura de la banda transportadora para cambiar la línea de producción, el cual limita el posicionamiento exacto del sistema, ocasionando problemas al momento de la captura de las imágenes.

Si se mejorase dichas condiciones se podría llegar a tener una eficiencia del 98%, dando como resultados los datos que se visualizan en la Tabla 4-5.

Tabla 4-5. Pérdidas totales de la empresa por fallos en la codificación luego de implementar el sistema de visión artificial mejorado

Tiempo de falla en la línea de producción	Total de latas con errores	Red neuronal Eficiencia 98%	Pérdidas económicas (dólares)
1 min	200	4	2.88
5 min	1000	20	14.4
10 min	2000	40	28.8

Elaborado por: (Palma Wilson ,2019)

Mediante las Tablas 3,4 y 5 se puede generar las siguientes curvas estadísticas de eficiencia vs pérdidas de la empresa. En la Figura 4-1 se puede observar que si no existiese el sistema de visión artificial la empresa pierde 144 dólares en 1 min, mientras que, con el sistema implementado, la empresa pierde como máximo 23,04 dólares cada 1 min, y si se mejora la red neuronal la pérdida fuera mínima de 2,88 dólares.

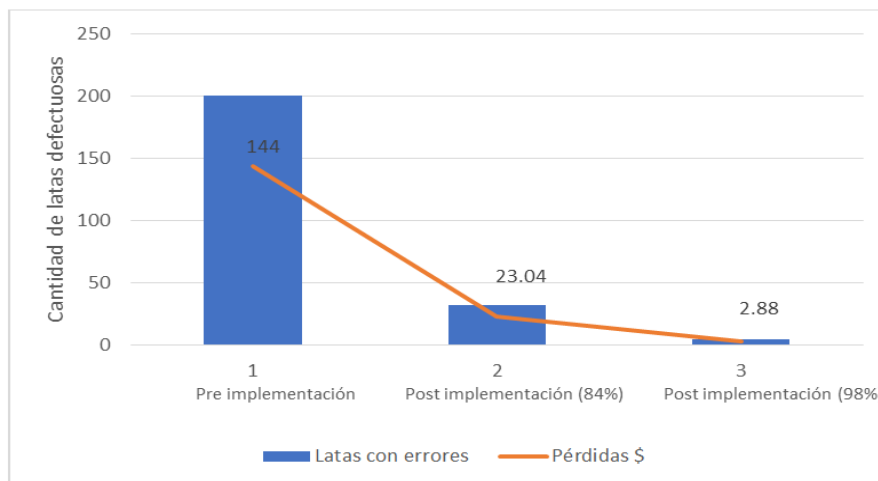


Gráfico 4-1. Latas con errores vs pérdidas

Elaborado por: (Palma Wilson ,2019)

4.2.1. Eficiencia de la red neuronal aplicada en Puerto Mar a 84%.

Para el caso práctico de la implementación la empresa PuertoMar S.A facilitó sus instalaciones y de igual manera sus líneas de producción de codificado de latas de atún.

Luego de la implementación se tomó como prueba un tiempo de 3 meses para verificar la efectividad del sistema de visión artificial en los días normales de trabajo para los cuales de acuerdo a los datos censados por el sistema se muestran un ahorro significativo para la empresa en tiempos cortos de falla.

Tomando como base el cálculo anterior en el cual mediante investigación de campo al departamento de contabilidad y producción, se tiene un valor de 0.72 ctvs cada lata de atún una vez codificada.

$$\text{Ganancia por min} = \text{Costo de cada lata} * \text{Número de latas por min}$$

$$\text{Ganancia por min} = 0.72 \text{ ctvs} * 200$$

$$\text{Ganancia por min} = \$144$$

Tabla 4-6 Datos adquiridos en PuertoMar S.A.

	Minutos de Falla	Pérdidas económicas pre diseño	Perdidas económicas pos diseño	Ganancias económicas
1 Mes	8	1152	184.32	967.68
2 mes	15	2160	345.6	1814.7
3 mes	10	1440	230.40	1209.6
			Total	3991.98

Elaborado por: (Palma Wilson ,2019)

De acuerdo a los datos obtenidos en primer mes de la implementación, con la empresa operando en su línea de producción a plena carga se obtuvieron 8 minutos de falla en el transcurso del mes. Con el sistema de visión artificial implementado y con una eficiencia de un 84% se obtuvieron los siguientes datos, mostrados en el siguiente gráfico.

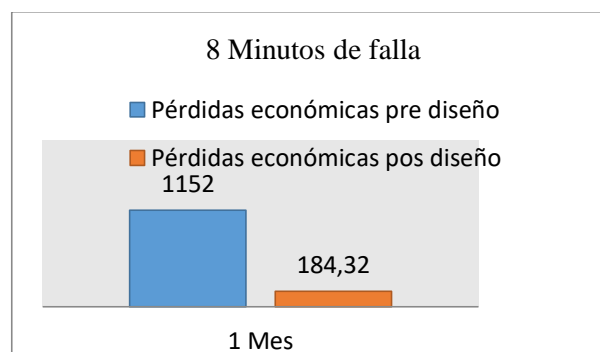


Gráfico 4-2. Pre Implementación vs Post Implementación con 8 minutos de falla

Elaborado por: (Wilson Palma, 2019)

La ganancia de la empresa en el primer mes fue de \$ 967.68 el cual considerando el tiempo de fallas durante todo el mes se considera significativo, en relación al costo de la implementación.

Para el segundo mes de la implementación en la empresa el número de fallas incremento a 15 minutos en todo el transcurso de mes la empresa operando en su línea de producción a plena carga lo cual representa un ahorro significativo.

Con el sistema de visión artificial implementado con una eficiencia de un 84% se obtuvieron los siguientes datos.

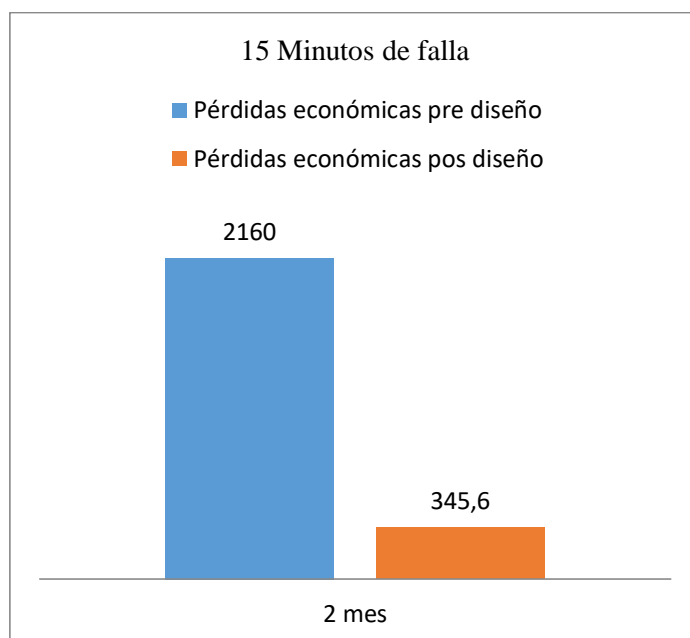


Gráfico 4-3. Pre Implementación Vs Post Implementación con 15 minutos de falla.

Elaborado por: (Wilson Palma, 2019)

Para el segundo mes la ganancia de la empresa fue de \$ 1814.7 el cual considerando el tiempo de fallas fue de 10 minutos durante todo el mes, se considera un ahorro significativo, en relación al costo de la implementación.

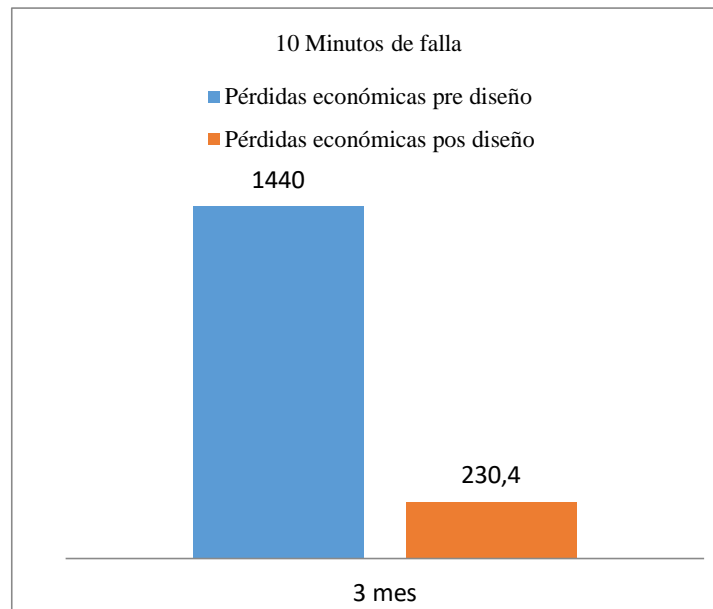


Gráfico 4-4. Pre Implementación Vs Post Implementación con 10 minutos de falla.

Elaborado por: (Wilson Palma, 2019)

En el tercer mes de la implementación la ganancia de la empresa fue de \$ 1209.6 el cual considerando el tiempo de fallas fue de 10 minutos durante todo el mes, se considera un ahorro significativo, en relación al costo de la implementación.

En total en los 3 meses de análisis la Empresa PuertoMar S.A se obtuvo un ahorro de \$ 3991.98 mejorando así la línea de producción y disminuyendo considerablemente las perdidas por reprocesamiento.

4.2.2. Validación de la hipótesis.

Para realizar la validación de la hipótesis se plantea la siguiente hipótesis nula (H0) e hipótesis alternativa (H1) respectivamente, siguiendo los siguientes pasos

Paso 1 Plantear las hipótesis.

- H0: El porcentaje de eficiencia de un sistema un sistema de visión artificial sustentado en redes neuronales del proceso de codificado de atún no es mayor a un 85%

$$p \leq 85\%$$

- H1: El porcentaje de eficiencia de un sistema un sistema de visión artificial sustentado en redes neuronales del proceso de codificado de atún es mayor a un 85%

$$p > 85\%$$

Paso 2 Seleccionar el nivel de significancia (alfa)

Se trabaja con un nivel de significancia del 5 %

Paso 3 Calcular el valor del estadístico de prueba.

Se va a utilizar como estadístico de prueba la distribución normal Z cuya ecuación es la siguiente para una proporción.

$$Z = \frac{\bar{p} - p}{\sqrt{\frac{p \cdot q}{n}}} \quad (4.2)$$

Dónde:

- \bar{p} = Porcentaje de eficiencia del sistema.
- p = Porcentaje de eficiencia esperado.
- q = Porcentaje de fallas esperadas.
- n = Número de muestras.

$$Z = \frac{0.8367 - 0.85}{\sqrt{\frac{0.85 \cdot 0.15}{49}}} \quad (4.3)$$

$$Z = 0.26$$

Paso 4. Establecer la regla de decisión.

Como se trabaja con un nivel de significancia del 5% y es una prueba de hipótesis de una sola cola tenemos un valor $Z_c = 1.65$.

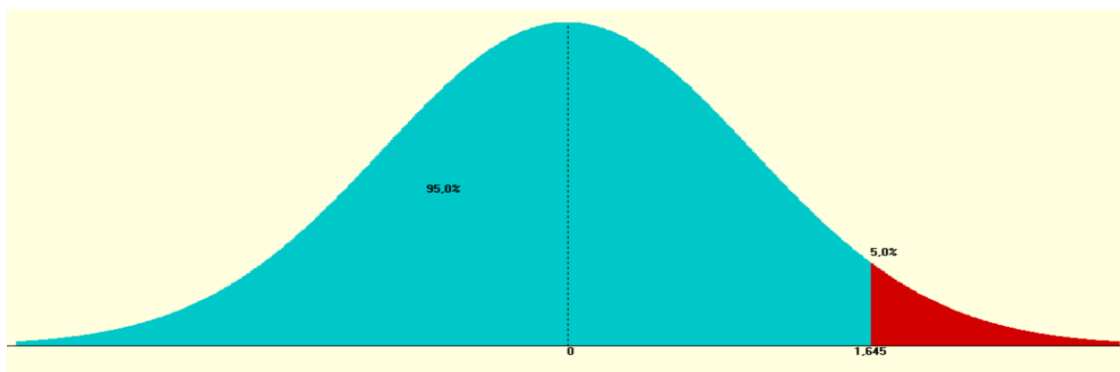


Figura 4-1. Distribución Normal

Elaborado por: (Palma Wilson ,2019)

La regla de decisión es se acepta H_0 si $Z < Z_c$, caso contrario se rechaza H_0

Paso. 5 Toma de decisión

Como $0.26 < 1,65$ se acepta H_0 , por lo tanto se concluye que el porcentaje de eficiencia de un sistema un sistema de visión artificial sustentado en redes neuronales del proceso de codificado de atún no es mayor a un 85%

4.3. Análisis financiero.

En la Tabla 4-6 se observan los detalles de los gastos generales realizados en el transcurso de la elaboración del sistema de visión artificial.

Tabla 4-7. Análisis económico de los elementos utilizados en la elaboración del proyecto

Cantidad	Detalle	Valor Unitario \$	Valor Total \$
Material del ensamble general			
1	Estructura Metálica	95	95
12	Tuerca, tornillos y arandelas	0.58	7
1	Llave hexagonal	5,5	5.5
1	Interruptor, cinta doble faz, lijas	5,34	5.34
1	Lima	4,75	4.75
2	Canaletas	1,85	3.71
1	Caja metálica	14	14
Subtotal			135.3
Material para el sistema de visión artificial			
2	Paneles leds	9.56	19.17
1	Cámara Logitec	90	90
1	Luz tipo sirena	35	35
5	Metro de cable gemelo #18	0.77	3.86
1	Módulo temporizador	18,65	18.65
1	Arduino UNO	12,5	12.5
1	Sensor infrarrojo	9,8	9.8
1	Fuente de 12v + cable	9,15	9.15
1	Cable extensor	2,9	2.9
Subtotal			201.03
Otros gastos			
	Papel		10
	Impresiones		25
	Materiales varios		14.11
	Viáticos		51.5
Subtotal			100.61
TOTAL			436.94

Elaborado por: (Palma Wilson ,2019)

El costo total para el sistema de visión artificial alcanza la suma de \$436.94 dólares sin considerar los costos de diseño e implementación realizada por el autor, comparando este monto con un sistema de visión artificial industrial para la clasificación de la compañía Festo (2018) que bordean los \$9000 dólares, se puede concluir que el sistema es factible económicamente.

CAPÍTULO V

5. PROPUESTA

5.1. Desarrollo de la propuesta alternativa

El desarrollo de la propuesta alternativa se plantea en función de la realización y cumplimiento de los objetivos planteados.

- ✓ Analizar las condiciones técnicas para el uso de sistema de visión artificial en el proceso de producción del atún.

Para alcanzar el presente objetivo se tomó como referencia las industrias atuneras en la provincia de Manabí, las que en su mayoría se encuentran en los cantones Manta, Montecristi y Jaramijó, y por ende tienen normas de control y seguridad interna que son muy estrictas por lo que exteriorizar sus procesos no se permite, ante estas condiciones de confidencialidad la empresa Puertomar S.A facilitó sus instalaciones para el desarrollo de este proyecto y se hará público el nombre de ella ya que será parte de este proceso de estudio

- ✓ Desarrollar un modelo de visión, sustentado en redes neuronales artificiales (neocognitron) para la detección y aviso emergente ante fallas de codificado mediante alarma sonora.

Dentro de este objetivo se desarrolló un modelo de visión artificial que actuó a tiempo real y para ello se usó como clasificador en la toma de decisiones una red neuronal artificial convolucional, debido a que era necesaria la realización del diagnóstico de las fallas en tiempo real, y para lo cual se utilizaron imágenes captadas por una cámara de esta manera no solo se llegó a su detección y aislamiento sino también a la estimación de su magnitud, con vistas a interrumpir la propagación de la falla en el codificado de las latas de atún en la línea de producción.

Debido a que se tuvo que detectar fallos en un proceso complejo, donde existe incertidumbre paramétrica y estructural como es el caso de estudio que ocupa (línea de producción de atún), no existen modelos matemáticos del mismo, por lo que se requieren técnicas de inteligencia artificial para el reconocimiento de los patrones creados (en el caso de análisis, las imágenes de codificación de las latas) con herramientas computacionales como las redes neuronales artificiales convolucional estimada en el modelo de neocognitron.

- ✓ Obtener el algoritmo de adquisición y el respectivo procesamiento de las imágenes.

Para la consecución de este objetivo se llevó a cabo un proceso de adquisición y procesamiento de imágenes para la adquisición de las imágenes se utilizó una cámara Logitech C920 y las librerías “QImage” y “QPixmap” del programa Qt Creator, en el HMI se presenta el video del proceso y con el comando “QImage” se genera una captura del fotograma cuando el sensor infrarrojo indique que existe la presencia de una lata de atún en la banda transportadora. Esta imagen es guardada en una dirección que se le especifique en el computador, para luego ser utilizada en el pre procesamiento de la misma.

Para la visualización de los videos e imágenes de análisis se utilizó el comando “QPixmap” que permite la visualización de las mismas en el HMI. Es necesario tener en cuenta las siguientes consideraciones al momento de adquirir las imágenes, también cabe recalcar que el modelo usado es una red llamada “Mobilenet_1.0_224”, es decir un modelo convolucional de 224 pixeles de análisis en su primera versión.

Una vez que generados los 4000 pasos, se obtuvieron los siguientes resultados:

- Eficiencia general de la red neuronal 97.2%
- Tiempo promedio de análisis 0.3 ms

Para la implementación del sistema de visión artificial se utilizó el Software Qt Creator, el cual permite realizar toda la etapa de visión artificial desde la adquisición de la imagen, pre procesamiento, comparación mediante la red neuronal, comunicación con los sensores y actuadores y manipulación del sistema con el HMI, además de ser un software de baja complejidad y altas prestaciones al momento de editar el código tales como:

- Editor de código con soporte para C++, QML y ECMAScript
- Herramientas para la rápida navegación del código
- Resaltado de sintaxis y auto-completado de código
- Control estático de código y estilo a medida que se escribe
- Soporte para refactoring de código.
- Ayuda sensitiva al contexto.
- Plegado de código (code folding).
- Paréntesis coincidentes y modos de selección.

Para el entrenamiento de la red neuronal se utilizó la librería liberada por Google llamada TensorFlow, la cual comparada con otras librerías brinda mayor facilidad en su utilización, mejores

tiempos de procesamiento y los mejores porcentajes de efectividad en la red neuronal, lo cual se detalla en el estado del arte anteriormente.

- ✓ Analizar mediante una matriz de comparación la mejora del proceso con la implementación del sistema de visión artificial.

Para el desarrollo de este objetivo se realizó un análisis económico pre-implementación y post-implementación, lo que significaba en realidad las pérdidas económicas que la empresa estaba teniendo por cada minuto de falla en su línea de producción de codificado de latas de atún, para lo cual se estableció un antes y un después

Tabla 5-1 Pérdidas totales de la empresa por fallos en la codificación luego de implementar el sistema de visión artificial

Tiempo de falla en la línea de producción	Total de latas con fallas	Red neuronal Eficiencia 84%	Pérdidas económicas (dólares)
1 min	200	0.16	23.04
5 min	1000	0.16	115.2
10 min	2000	0.16	230.4

Elaborado por: (Wilson Palma, 2019)

Debido a los diferentes factores en la planta que alteraron la eficiencia en la red neuronal no se pudo llegar a una eficiencia del 100%. Estos inconvenientes no permitieron la correcta toma de imágenes que serán enviadas a la red neuronal.

- Vibraciones en la estructura de la banda transportadora que alteraban el enfoque.
- Altas temperaturas y humedad en el ambiente que empañaban la lente de la cámara
- Condiciones de luz variables, las cuales no permitían tener un brillo uniforme en todo el sistema.
- El cambio manual en la estructura de la banda transportadora para cambiar la línea de producción, el cual limita el posicionamiento exacto del sistema, ocasionando problemas al momento de la captura de las imágenes.

5.2. Prueba de clasificación de latas de atún

El objetivo de esta prueba es determinar si se realiza la clasificación exitosa de las latas de atún codificada de una manera adecuada o errónea, mediante la toma de decisión de la red neuronal y el accionamiento de la baliza lumínica señalizadora.

Para validar el sistema de clasificación se tomó como punto de partida una línea de producción de 200 latas por minuto, sobre la cual se realizó la implementación.



Figura 5-1 Resultado del análisis a las latas de atún con codificación correcta.

Elaborado por: (Palma Wilson ,2019)

En la figura 5-1 se puede observar el análisis de una lata de atún correctamente codificada, la cual pasa por todas las etapas de procesamiento digital de imágenes, se le aplican los respectivos filtros y máscaras de recortes, para luego ser comparada con el data set ingresado en la base de datos de la red neuronal, comprara mediante convoluciones.

En conclusión el porcentaje de aceptabilidad de 97.1754% y un tiempo de respuesta de 0.018 segundos y toma la decisión en estado correcto o incorrecto



Figura 5-2 Actuador activado

Elaborado por: (Palma Wilson ,2019)

CONCLUSIONES.

- ✓ Se elaboró una red neuronal convolucional con una eficiencia del 84% mediante las librerías de TensorFlow y con ayuda del modelo Convolucional (Neocognitron) Mobilenet_v1_1.0_224, para ello se utilizó un data set de 417 imágenes, y se entrenó mediante 4000 pasos de iteración.
- ✓ La eficiencia de la red neuronal depende de varios factores ambientales como: temperatura, vibración, ruido, humedad entre otros factores ambientales, así como también de la cantidad de imágenes con las que se entrene el modelo.
- ✓ A través del uso de funciones y algoritmos de procesamiento de imágenes (GaussianBlur, HoughCircles, adaptiveThreshold, entre otras) se puede llevar a cabo mejoras de imágenes con el fin de incrementar o resaltar aspectos relevantes para su posterior análisis.
- ✓ De tal modo que este proceso permite mejorar el rendimiento de la red neuronal debido a que la información que recibe la red es menor pero más relevante, como en el caso de las latas de atún que únicamente se requiere su código, más no su fondo ni algún atributo adicional.
- ✓ Sin el sistema de visión artificial la empresa pierde 144 dólares cada minuto, mientras que, con la red neuronal implementada, se disminuyó la pérdida a 23 dólares, generando así, una ganancia de 121 dólares cuando existan fallos en el sistema.
- ✓ Con la implementación del sistema la empresa mejoro su eficiencia en un 84 % dentro de la línea de codificación de latas de atún generando así una ganancia sustancial como lo demuestran los cálculos obtenidos.

RECOMENDACIONES.

- ✓ Se recomienda usar una cámara con un número de frames por segundo superior a 90, ya que la velocidad de las bandas transportadoras es de 3.33 latas por segundo, el cual, si no se estudia con detenimiento dichas velocidades, puede generar distorsiones en el momento de la toma de imágenes.
- ✓ Es recomendable la utilización de una tarjeta de control de mayor robustez que tiene Arduino Uno, puesto que la temperatura dentro de la planta atunera es bastante elevada, además de esto conseguir mejorar los tiempos de envío y recepción de los datos a procesar.
- ✓ Se recomienda entrenar el modelo con un data set más grande con el fin de mejorar la eficiencia de la red neuronal, y lograr una efectividad superior al 95%.
- ✓ Se recomienda el uso de librerías de visión artificial actualizadas, y evitar el uso de librerías antiguas debido a que las nuevas librerías siempre presentan mejoras respecto a sus algoritmos, funciones o tiempo de procesamiento, volviéndose cada vez más eficientes.
- ✓ Investigar acerca de métodos más eficientes para el procesamiento de imágenes que permitan mejorar las características necesarias para el análisis, mejorando de forma significativa el tiempo de procesamiento en el computador y a su vez disminuyendo el tiempo de acción de la red neuronal y de todo el programa.
- ✓ Utilizar un computador con una tarjeta gráfica lo suficientemente potente para poder entrenar la red neuronal, y que el HMI pueda visualizarse lo más fluido posible

BIBLIOGRAFÍA

- © 2020 Arduino. (2020). Machine vision with low-cost camera modules. Retrieved from www.arduino.cc
- Alonso Marco. (2009). Espacios de Color RGB , HSI y sus Generalizaciones a n -Dimensiones., 118.
- BcnVision. (2017). Sistemas de iluminación para aplicaciones de visión artificial. Retrieved from <http://www.bcnvision.es/blog-vision-artificial/iluminacion-vision-artificial2/>
- Berzal, F. (2018). Redes Neuronales & Deep Learning, 803. Retrieved from <https://www.amazon.es/Redes-Neuronales-Deep-Learning-Edición/dp/1731314337>
- Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2016). *Diagnosis and Fault-Tolerant Control*.
- Bravo, C., Aguilar, J., & Rivas, F. (2004). Diseño de una arquitectura de automatización industrial basada en sistemas multi-agentes Industrial automation architecture based on multi-agents systems, 25(2).
- Constante, P., Chang, O., Pruna, E., & Escobar, I. (2016). Artificial Vision Techniques for Strawberry ' s Industrial Classification, 14(6), 2576–2581.
- FAO. (2018). Organización de las Naciones Unidas para la Alimentación y la Agricultura. Retrieved from <http://www.fao.org/home/es/>
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <https://doi.org/10.1007/BF00344251>
- Galushkin Alexander. (2007). *Neural Networks Theory. Journal of Chemical Information and Modeling* (Vol. 53).
- García Flores De Valga. (2018). “Integración De Codificadores Industriales En Sistemas Automatas De Producción Y Su Influencia En La Eficiencia Del Proceso Industrializado Del Atún En La Provincia De Manabí, República Del Ecuador.”

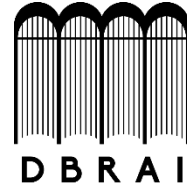
- Garzón Juan. (2013). *Escuela politécnica nacional. Automatización del sistema de refrigeración de las cámaras de congelación de eurofish S.A.*
- Gonzales, A., Martínez, F., Alba, F., Pernía, A., Castejón, M., Ordieres, J., & Vergara, E. (2006). *Técnicas Y Algoritmos Básicos De Visión Artificial* (Vol. 53).
<https://doi.org/10.1017/CBO9781107415324.004>
- González José. (1973). Los sistemas de codificación de artículos, 1–9.
- Gorokhovatskyi, O. (2016). Neocognitron as a tool for optical marks recognition. *Proceedings of the 2016 IEEE 1st International Conference on Data Stream Mining and Processing, DSMP 2016*, (August), 169–172. <https://doi.org/10.1109/DSMP.2016.7583533>
- Infaimon. (2019). Visión Artificial. Retrieved from <https://www.infaimon.com/>
- Kim, S. (2013). [CNN] Applications of Convolution in Image Processing with MATLAB, 385, 19. Retrieved from www.math.washington.edu/~wcasper/math326/projects/sung_kim.pdf
- Lab-Volt (Quebec) Ltda. (2001). *Fluidos y Sensores*.
- Logitech. (2018). Cámaras Web. Retrieved from <https://www.logitech.com/es-mx>
- Matich Damián. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia*, 55. Retrieved from <ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf>
- Mendiburu, H. (2013). *Instrumentación Virtual Industrial*.
- OpenCV. (2019). OpenCV. Retrieved from <https://opencv.org>
- Outeiriño, M. B. P. (2016). Germán Corona Ramírez, Leonel, Abarca Jiménez, Griselda S., and Mares Carreño, Jesús. *Sensores y actuadores: aplicaciones con Arduino*. Distrito Federal, MÉXICO: Grupo Editorial Patria, 2014. ProQuest ebrary. Web. 12 October 2016. Copyright © 2014. Grupo E, (October).
- Platero, C. (2009). *Introducción a la Visión Artificial*.

- Poli, G., & Saito, J. (2006). Parallel Face Recognition Processing using Neocognitron Neural Network and GPU with CUDA High Performance Architecture. *Milos Oravec, ISBN*. Retrieved from http://cdn.intechopen.com/pdfs/10202/InTech-Parallel_face_recognition_processing_using_neocognitron_neural_network_and_gpu_with_cuda_high_performance_architecture.pdf
- Prateek, J. (2017). *Artificial Intelligence with Python*. <https://doi.org/10.1002/ejoc.201200111>
- Puig, V., Quevedo, J., Escobet, T., Morcego, B., & Ocampo, C. (2004). Control Tolerante a Fallos (parte I): Fundamentos y Diagnóstico de Fallos, *1(Parte I)*, 15–31.
- Rechner. (2016). Sensores Magneto-Resistivos, 26. Retrieved from <https://www.rechner-sensors.com/wp-content/uploads/2016/12/kat-mrs-es.pdf>
- Rechner. (2019a). Sensores Capacitivos S26, 119. Retrieved from <https://www.rechner-sensors.com/dokumentation/kataloge>
- Rechner. (2019b). Sensores Inductivos S26, *1*, 100. Retrieved from <https://www.rechner-sensors.com/dokumentation/kataloge>
- Rungta Krishna. (2018). TensorFlow in 1 Day : Make your Own Neural Network, 449.
- Sanchis Roberto, Romero Julio, A. C. (2018). *Automatización industrial*.
- Serra José. (2018). *Informatica a Bordo*.
- Simon, H. (1999). Neural Networks A Comprehensive Foundation. *International Journal of Neural Systems*. <https://doi.org/10.1142/s0129065794000372>
- Technologies, V. (2018). Errores de codificación y embalaje.
- Tecnología, D. (2010). Sistemas electrónicos, 1–12. Retrieved from http://iesodrapisuerga.centros.educa.jcyl.es/sitio/upload/SISTEMAS_ELECTRONICOS.pdf
- Vargas Víctor. (2010). *Sistema de Visión Artificial para el Control De Calidad en Piezas Cromadas*.

Velez Serrano. (2017). Visión por Computador.



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO



DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN

UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 02 / 07 /2020

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Wilson Sigifredo Palma Santacruz
INFORMACIÓN INSTITUCIONAL
Facultad: Instituto de Posgrado y Educación Continua
Carrera:
Título a optar: Magister en Sistemas de Control y Automatización Industrial
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.



02-07-2020
0102-DBRAI-UPT-2020