



“UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA”

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE ELECTRONICA Y AUTOMATIZACION

TRABAJO DE INTEGRACIÓN CURRICULAR

previo a la obtención del título de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

“Diseño y simulación de sistema automatizado para el control de calidad en la producción de enlatado de atún mediante visión artificial.”

AUTOR:

WILSON STALIN GALDEA GONZÁLEZ

TUTOR:

ING. SENDEY VERA GONZÁLEZ, Mgt.

LA LIBERTAD - ECUADOR

2023

AGRADECIMIENTO

Un agradecimiento especial a **Dios** por sus bendiciones, por los instrumentos proporcionados que necesitaba día a día para aprobar cada semestre y culminar con éxito mis estudios.

A mis padres **Anselmo** y **Anita**, por ser mis guías y formarme como una persona con valores, gracias por sus consejos y apoyo, a mis hermanos por su ayuda brindada durante mis estudios y esas palabras de motivación de que los logros se pueden cumplir con la confianza depositada en uno mismo.

A mi esposa **Verónica Montenegro**, a mis hijas **Allison** y **Tiffany**, por su comprensión , siendo mis fuentes de inspiración para seguir estudiando, gracias esposa por brindarme siempre tu apoyo incondicional para cumplir este logro y convertirme en un profesional, estaré agradecido toda la vida y con mi trabajo les compensare todo este esfuerzo que han realizado.

Al Ing. Sendey Vera González, mi tutor guía, por el aporte de ideas y sus conocimientos, siempre motivándome a desarrollar la propuesta de manera correcta para terminar mi tesis.

A los docentes de la facultad de Electrónica y Automatización que aportaron sus conocimientos para el fortalecimiento de mi aprendizaje y su aporte al proyecto de titulación.

DEDICATORIA

Este trabajo de titulación se la dedico a mi esposa Verónica Montenegro, por recibir siempre tu ayuda, por tus palabras de motivación para seguir cuando quería renunciar. Gracias esposa por ser ese motor en mi vida, fuente de inspiración de verte ejerciendo en tu vida profesional.

A mis padres que siempre se sacrificaron para brindarnos un estudio y convertirnos en profesionales. Siempre estaré agradecido por sus consejos y todo lo que hicieron por mí.

A mis hijas Allison y Tiffany que fueron ese pilar y fuente de inspiración para cada día de estudio, por su comprensión, permitiéndome culminar con éxito mis estudios universitarios.

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado **“Diseño y simulación de sistema automatizado para el control de calidad en la producción de enlatado de atún mediante visión artificial”**, elaborado por el egresado Galdea González Wilson Stalin, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo a el estudiante para que inicie los trámites legales correspondientes.

La libertad, 7 de Diciembre del 2023



Ing. Sendey Vera González, MSc

TRIBUNAL DE SUSTENTACIÓN



Ing. Ronald Rovira Jurado, Ph.D
DIRECTOR DE LA CARRERA DE
ELECTRONICA Y AUTOMATIZACION



Ing. Luis Chuquimarca Jimenez, Mgt
DOCENTE ESPECIALISTA
DOCENTE GUIA UIC



Ing. Sendey Vera González, Mgt.
DOCENTE TUTOR



Ing. Corina Gonzabay D., Mgt
SECRETARIA

RESUMEN

El proyecto consiste en la implementación de un sistema de control automatizado, que realiza la detección y clasificación de las fallas presentadas en las latas de atun aplicando técnicas de visión artificial. Los dispositivos utilizados en el sistema están compuestos por una banda transportadora que traslada las latas hacia la cámara para el reconocimiento , el sistema de visión consiste en reconocer el contorno, etiqueta y ubicación del abrefácil.

El proceso empieza mediante la detección de la lata por un sensor fotoeléctrico en ese instante se activa el Relay para detener la banda transportadora y que el sistema de visión proceda con el reconocimiento de la lata de atun, después de unos milisegundos se desactiva el Relay, activando la banda transportadora que llevara la lata hasta el segundo sensor, cuando se activa el sensor se inicia la secuencia de programación del brazo robot y según el estado de la lata , los movimientos de secuencia que realiza son, hacia derecha para las latas en buen estado y hacia la izquierda para las latas que se encuentran con las fallas , cuando finaliza el proceso de clasificación el brazo robot se dirige a su punto de inicio y se activa la banda para empezar otro proceso de inspección.

Se aplica la industria 4.0 con el sistema de monitoreo IOT, el bróker mosquitto permite publicar y suscribirse para el envío y recepción de datos en tiempo real, node red es para la programación de flujo, utilizamos InfluxDB para crear nuestra base de datos y grafana donde se crea el dashboards , para el monitoreo del sistema de visión hacia cualquier parte del mundo.

Palabras Claves: Visión Artificial, Yolov5, Reconocimiento OCR, Detección de Objeto, Comunicación IOT.

ABSTRACT

The project consists in the implementation of an automated control system, which it detects and classify failures in tuna cans by applying artificial vision techniques. The devices used in the system. They are composed by a conveyor belt, it allows to move the cans to the camera for recognition, the vision system recognize the contour, label and ring pull location

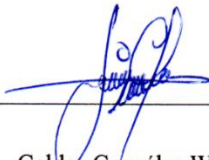
The process begins by detecting the can by a photoelectric sensor. At the same time, the Relay is activated to stop the conveyor belt and the vision system starts with the recognition of the tuna can. After that, a few milliseconds, the Relay is deactivated, and activating the conveyor belt that it will take the can to the second sensor, when the sensor is activated to begin the programming sequence of the robot arm and depending on the state of the can, the sequence of movements robot arm are: the right for cans in good condition and to the left for the cans are found with error, when the classification process ends, the robot arm goes from start point and the belt is activated to begin another inspection process.

In this project is applied the industry 4.0 with the IOT monitoring system, the mosquitto broker allows publishing and subscribing for sending and receiving data in real time, node red is for stream programming, in its use InfluxDB to create database and grafana in where create the dashboards to monitor the vision system anywhere in the world.

Keywords: Artificial Vision, Yolov5, OCR Recognition, Object Detection, IOT Communication.

DECLARACIÓN

El contenido del presente Trabajo de Titulación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.

A handwritten signature in blue ink, consisting of stylized, overlapping loops and lines, positioned above a horizontal line.

Galdea González Wilson Stalin

ÍNDICE GENERAL

1.	CAPÍTULO I.....	3
1.1.	ANTECEDENTES.....	3
1.2.	JUSTIFICACIÓN	4
1.3.	OBJETIVOS.....	7
1.3.1.	<i>Objetivo General:</i>	7
1.3.2.	<i>Objetivos Específicos:</i>	7
1.4.	DESCRIPCIÓN DEL PROYECTO.....	8
1.5.	METODOLOGÍA DE LA INVESTIGACIÓN.....	9
1.5.1.	<i>Investigación Exploratoria</i>	10
1.5.2.	<i>Investigación Descriptiva</i>	10
1.5.3.	<i>Investigación Experimental</i>	11
2.	CAPÍTULO II	12
2.1.	MARCO CONTEXTUAL.....	12
2.2.	MARCO CONCEPTUAL	13

2.2.1.	<i>Materia Prima</i>	13
2.2.2.	<i>Envase Lata de Atun</i>	13
2.2.3.	<i>Fallas en las latas de atún</i>	16
2.2.3.1.	<i>Sellado afilado</i>	16
2.2.4.	<i>Control de cierres</i>	17
2.3.	SISTEMA DE AUTOMATIZACIÓN	18
2.4.	SISTEMA DE TRANSPORTE.....	19
2.4.1.	<i>CINTA TRANSPORTADORA INDUSTRIAL</i>	19
2.5.	SISTEMA ELECTRÓNICO.....	19
2.5.1.	<i>SENSOR DE PROXIMIDAD</i>	20
2.6.	SISTEMA DE CONTROL	20
2.7.	SISTEMA DE VISIÓN ARTIFICIAL	20
2.7.1.	<i>Fases del sistema de visión artificial</i>	21
2.8.	PROCESAMIENTO DIGITAL DE IMÁGENES	21
2.8.1.	<i>Morfología matemática</i>	21
2.9.	SEGMENTACIÓN DE IMÁGENES.....	23
2.9.1.	<i>Técnica de umbralización</i>	23
2.10.	DETECTAR BORDES DE IMÁGENES EN ESCALA DE GRISES	24

2.10.1.	<i>Detectar bordes de imágenes en escala de grises: Operadores gradiente</i>	24
2.11.	TÉCNICAS MATEMÁTICAS DE COMPRESIÓN DE VIDEO PARA MEJORAR LA IMAGEN.....	26
2.11.1.	<i>Suavizado</i>	26
2.11.2.	<i>Expansión del histograma</i>	26
2.11.3.	<i>Manipulación del histograma</i>	27
2.11.4.	<i>Ecuación del Histograma</i>	28
2.12.	DETECCIÓN DE LOS REMACHES.....	28
2.13.	RECONOCIMIENTO DE TEXTO OCR.....	29
2.14.	DESARROLLO DEL SISTEMA DE VISIÓN ARTIFICIAL.....	30
2.14.1.	<i>SELECCIÓN DE LA CÁMARA</i>	31
2.15.	TIPOS DE APRENDIZAJE.....	33
2.15.1.	<i>Aprendizaje Supervisado</i>	33
2.15.2.	<i>Método de Aprendizaje</i>	34
2.15.3.	<i>Redes Neuronales Convolucionales</i>	35
2.15.4.	<i>Segmentación Semántica</i>	36
2.16.	MARCO TEORICO	37
3.	CAPITULO III.....	39
3.1.	COMPONENTES DE LA PROPUESTA.....	39

3.1.1.	<i>Componentes Físicos</i>	39
3.1.2.	<i>Componentes Lógicos</i>	54
3.2.	DISEÑO DE LA PROPUESTA.....	60
3.2.1.	<i>Topología de comunicación</i>	60
3.2.2.	<i>Diagrama Red de Comunicación del Sistema</i>	61
3.2.3.	<i>Diseño del Sistema de Control de Visión Artificial</i>	61
3.2.4.	<i>Sistema de control para productos de enlatados con sensor fotoeléctrico</i>	62
3.3.	DISEÑO ELÉCTRICO Y ELECTRÓNICO DEL PROTOTIPO	63
3.3.1.	<i>Conexión de los servomotores a la tarjeta Sensor Shield V5.0</i>	63
3.3.2.	<i>Conexión de los sensores fotoeléctricos a la placa Arduino Uno</i>	65
3.3.3.	<i>Conexión control automático de banda transportadora</i>	66
3.3.4.	<i>Diagrama de conexión motorreductor a la placa Arduino</i>	67
3.4.	INSTALACIÓN DEL SISTEMA OPERATIVO PI OS A LA RASPBERRY PI 4 B	67
3.5.	INSTALACIÓN DE ADVANCES IP SCANNER	69
3.6.	HABILITAR VNC VIEWER MEDIANTE COMANDO EN EL CMD.....	70
3.7.	INSTALACIÓN E INGRESO AL VNC VIEWER EN LA PC.	71
3.8.	CONFIGURACIÓN DE LA MEMORIA GPU EN LA RASPBERRY	71
3.9.	INSTALAR OPENCV-BULLSEYE EN RASPBERRY PI 4	72

3.10.	INSTALACIÓN DE TENSOR FLOW LITE.....	73
3.11.	INSTALACIÓN DE NUMPY	76
3.12.	INSTALACIÓN DE TESSERACT OCR.....	77
3.13.	HABILITAR 1-WIRE E I2C EN LA RASPBERRY PI 4	78
3.14.	INSTALACIÓN DE NODE-RED EN RASPBERRY PI 4B.....	79
3.14.1.	<i>Configuración de Seguridad para NodeRed</i>	80
3.15.	INSTALACIÓN DE MOSQUITTO	81
3.15.1.	<i>Configuración de mosquito</i>	82
3.16.	INSTALACIÓN DE INFLUXDB	84
3.16.1.	<i>Configuración de seguridad de InfluxDB</i>	85
3.16.2.	<i>Configuración de InfluxDB en node red</i>	86
3.17.	INSTALACIÓN DE GRAFANA.....	87
3.17.1.	<i>Conexión fuente de datos InfluxDB a Grafana</i>	88
3.18.	DESARROLLO DEL SISTEMA DE VISIÓN ARTIFICIAL	89
3.18.1.	<i>Construcción del dataset del modelo de Visión Artificial.....</i>	90
3.19.	ENTRENAMIENTO DEL MODELO DE OBJECT DETECTION EN GOOGLE COLAB.....	95
3.19.1.	<i>Descarga del modelo de entrenamiento para raspberry pi 4</i>	100
3.19.2.	<i>Algoritmo de Detección de Objetos en Tiempo Real.....</i>	100

3.19.3.	<i>Entrenamiento del modelo reconocimiento de texto en Google Colab y aplicación de EasyOCR.</i>	103
3.20.	PRUEBAS Y RESULTADOS.....	106
3.20.1.	<i>Pruebas de funcionamiento.....</i>	106
3.20.2.	<i>Resultados en tiempo real del sistema de visión con cámara Raspberry HQ.....</i>	107
3.20.3.	<i>División de las clases etiquetadas.....</i>	110
3.20.4.	<i>Análisis de métricas de precisión con 200 épocas.....</i>	111
3.20.5.	<i>Matriz de confusión.....</i>	112
3.20.6.	<i>Graficas con curvas de confianza y de precisión.....</i>	113
3.20.7.	<i>Resultados de métricas con entrenamiento de 200 épocas.....</i>	114
3.20.8.	<i>Resultados de las inferencias en Google Colab.....</i>	115
3.20.9.	<i>Sistema de Clasificación con el Brazo Robótico.....</i>	116
3.20.10.	<i>Secuencias del funcionamiento del brazo robótico.....</i>	117
3.20.11.	<i>Análisis de resultados del sistema de control con visión artificial.....</i>	119
3.20.12.	<i>Comunicación del Sistema IoT.....</i>	121
3.20.13.	<i>Inicialización de Node-Red en la Raspberry pi 4.....</i>	121
3.20.14.	<i>Integración de la cámara raspberry HQ.....</i>	123
3.20.15.	<i>Transmisión de frame en tiempo real.....</i>	123

3.20.16.	<i>Inicialización de Mosquitto en la Raspberry pi 4</i>	123
3.20.17.	<i>Integración de Node-Red con Mosquitto</i>	124
3.20.1.	<i>Comunicación remota con mosquitto</i>	125
3.20.2.	<i>Inicialización de InfluxDB en la Raspberry pi 4</i>	126
3.20.3.	<i>Creación de Database InfluxDB</i>	126
3.20.4.	<i>Creación del dashboards en Grafana</i>	127
3.21.	ESTUDIO DE FACTIBILIDAD.....	128
3.21.1.	<i>Factibilidad Técnica</i>	128
3.22.	ANALISIS DE COSTOS DEL PROYECTO.	129
3.22.1.	<i>COSTOS DE EQUIPOS</i>	129
4.	CAPITULO IV	131
4.1.	CONCLUSIONES.	131
4.2.	RECOMENDACIONES.	132
5.	ANEXOS	133
6.	BIBLIOGRAFÍA.....	136

ÍNDICE DE FIGURAS

Figura 1: Atun Bonito.....	13
Figura 2: Lata de Atun.....	14
Figura 3: Doble Cierre del Sellado de Lata	15
Figura 4: Borde de Sellado Afilado.....	16
Figura 5: Anilla de Abrefácil mal posicionado	17
Figura 6: Etiqueta de Codificación no impresa	17
Figura 7: Sistema de Visualización Código OCR	30
Figura 8: Tamaño de Detección de objeto.....	31
Figura 9: Tipos de Aprendizaje Supervisado y no Supervisado	33
Figura 10: Entrenamiento de Sistema Autónomo y Eficiente	34
Figura 11: Métodos de Aprendizaje	34
Figura 12: Versiones de YOLOv5 para el entrenamiento	35
Figura 13: Red Neuronal Convolutacional.....	36
Figura 14: Etiqueta segmentación de Imagen.....	37

Figura 15: Cámara Raspberry PI High Quality	40
Figura 16: Banda Transportadora Vevor	41
Figura 17: Motorreductor	42
Figura 18: Speed Control Motor.....	43
Figura 19: Brazo Robótico ARM-N2	43
Figura 20: Servomotor MG996R.....	44
Figura 21: Mini Bomba de vacío.....	45
Figura 22: Caja de Control Eléctrico.....	46
Figura 23: Sensor de proximidad E18-D80NK.....	46
Figura 24: Conductor Eléctrico	48
Figura 25: Rele Arduino 110VAC a 5V DC	49
Figura 26: Tarjeta Raspberry PI 4 4GB.....	49
Figura 27: Sensor Shield V5.0.....	51
Figura 28: Placa Arduino Uno R3	52
Figura 29: Fuente de Alimentación AC/DC	53

Figura 30: Interruptor Termomagnético	54
Figura 31: Software Raspberry Pi Imager	54
Figura 32: Lenguaje de Programación Python	55
Figura 33: Librería OpenCV.....	56
Figura 34: Librería Tensor Flow	56
Figura 35: Librería Pandas	57
Figura 36: Librería Ultralytics.....	57
Figura 37: Node Red.....	58
Figura 38: Protocolo de Red Mosquitto	58
Figura 39: Base de Datos InfluxDB	59
Figura 40: Grafana.....	59
Figura 41: Diagrama de Comunicación del sistema	61
Figura 42: Sistema de control de Visión Artificial.....	62
Figura 43: Conexión del motor al Relay	63
Figura 44: Esquema conexión de servomotor brazo robótico	64

Figura 45: <i>Configuración de pines del sensor fotoeléctrico E18-D80NK</i>	65
Figura 46 : Conexión de sensor fotoeléctrico	66
Figura 47: Diagrama de conexión del motorreductor a Arduino.....	67
Figura 48: Sistema Operativo PI OS 64 Bits.....	68
Figura 49: Verificación de Instalación del sistema operativo	69
Figura 50: Instalación de Advances IP Scanner	69
Figura 51: Comandos para ingreso a la Raspberry	70
Figura 52: Habilitar el VNC Viewer por CMD.....	70
Figura 53: Autenticación de ingreso a VNC Viewer	71
Figura 54: Instalación de Opencv en la Raspberry Pi 4	73
Figura 55: Comandos y versión de Opencv.....	73
Figura 56: Comandos Instalación de Librerías Tensor Flow Lite	74
Figura 57: Ubicación de carpeta tensorflow lite.....	75
Figura 58 : Ubicación de carpeta image-segmentation	75
Figura 59: Versión de tensorflow lite	76

Figura 60 : Instalación de librerías Numpy	76
Figura 61: Instalación de librerías para Tesseract OCR	77
Figura 62 : Paquetes para Tesseract OCR	78
Figura 63: Habilitar 1-Wire e I2C en la Raspberry pi 4	78
Figura 64: Comandos de Instalación Node-Red.....	79
Figura 65: Node-Red como Servicio	79
Figura 66: Conexión en línea con Node-Red	80
Figura 67: Configuración de acceso a Node-Red.....	81
Figura 68: Líneas de seguridad Node-Red	81
Figura 69: Línea de comando instalación de mosquito	82
Figura 70 : Configuración de password.....	83
Figura 71: Configuración de seguridad	84
Figura 72 : Instalación de InfluxDB.....	85
Figura 73: Configuración de seguridad InfluxDB.....	86
Figura 74: Configuración de propiedades de InfluxDB	87

Figura 75: Instalación de Grafana	87
Figura 76: Comando ejecución de grafana	88
Figura 77: Carpetas para Imágenes y Anotaciones	90
Figura 78: Importacion de Imágenes al software Makesense.....	91
Figura 79: Creación de labels de las imágenes	92
Figura 80: Etiqueta Falla Abrefácil	93
Figura 81: Etiqueta Falla Contorno	93
Figura 82 : Etiqueta en buen estado.....	94
Figura 83: Falla en Etiqueta.....	94
Figura 84: Exportación de las anotaciones de imágenes	95
Figura 85 : Selecccion de GPU en Google Colab	95
Figura 86 : Clonación del repositorio Ultralytics yolov5	96
Figura 87 : Carpeta data images y labels.....	97
Figura 88 : Modificación de archivo coco128.yaml.....	98
Figura 89: Entrenamiento con YOLOV5	99

Figura 90 : Descarga de archivo de entrenamiento best.pt	100
Figura 91 : Etiqueta de codificación OCR.....	104
Figura 92 : Cuadros delimitadores de texto	105
Figura 93 : Cuadro delimitador del texto.....	105
Figura 94 : Reconocimiento de etiqueta	106
Figura 95 : Resultado de inspección falla en etiqueta	108
Figura 96 : Resultado de inspección falla en contorno.....	108
Figura 97 : Resultado de inspección falla en abrefácil.....	109
Figura 98 : Resultado de inspección lata en buen estado de calidad.....	110
Figura 99 : División de las clases etiquetadas	110
Figura 100 : Graficas de metric/precision-recall-mAP0.5-mAP0.5:0.95	112
Figura 101 :Matriz de confusión	113
Figura 102 : Graficas curvas de precisión	113
Figura 103 : Graficas de curva de confianza y precisión	114
Figura 104 : Imágenes de las inferencias durante el entrenamiento	115

Figura 105 : Inferencias del entrenamiento en Google Colab	116
Figura 106: Desplzamiento hacia la derecha lata con fallas	118
Figura 107: Desplazamiento del brazo hacia la izquierda	118
Figura 108: Inicialización de Node-Red en la Raspberry pi 4	122
Figura 109: Pantalla de programación nodos de Node-Red	122
Figura 110: Comando inicialización de mosquitto	124
Figura 111: Configuración node red -mosquitto	124
Figura 112: Configuración del Server Mosquitto	125
Figura 113: Comunicación remota de Mqtt.....	126
Figura 114: Comando creación de data base influxdb	126
Figura 115: Comando verificación de base de datos influxdb	127
Figura 116: Ingreso de datos al data sources	128

ÍNDICE DE TABLAS

Tabla 1: Datos de Resolución de Megapíxeles.....	32
Tabla 2: Características de banda transportadora	41
Tabla 3: Especificaciones Técnicas de motor.....	42
Tabla 4 : Especificaciones Técnicas Servomotor MG996R.....	44
Tabla 5: Especificaciones Técnicas de Bomba de vacío	45
Tabla 6: Ficha Técnica Sensor Proximidad E18-D80NK.....	47
Tabla 7: Característica de sensor inductivo y sensor de proximidad.....	47
Tabla 8: Ficha Técnica Raspberry Pi 4 B	50
Tabla 9: Especificaciones Técnicas Sensor Shield V5.0	51
Tabla 10: Especificaciones Técnicas Arduino Uno.....	52
Tabla 11: Especificaciones Técnicas de Fuente de Alimentación AC/DC.....	53
Tabla 12: Direcciones IP de Dispositivos	60
Tabla 13: Pines de conexión Sensor Fotoeléctrico	66
Tabla 14 : Comandos Actualización del Firmware	72

Tabla 15: Comandos Instalación de Tensorflow Lite.....	74
Tabla 16 Pruebas del modelo de IA con 200 épocas.	107
Tabla 17: Métricas del entrenamiento en Google Colab con 200 épocas.....	115
Tabla 18: Pines PWM para el sistema de control del brazo robótico	117
Tabla 19 : Métricas del entrenamiento con 29 épocas.....	119
Tabla 20 : Métricas del entrenamiento con 200 épocas.....	120
Tabla 21 : Configuración de Grafana.....	127
Tabla 22 <i>Costo de Equipos</i>	130

ÍNDICE DE ANEXOS

Anexo A: Primera parte de programación en Arduino del sistema de clasificación del brazo robótico.	133
Anexo B : Primera parte de programación en Python del sistema visión artificial.	134
Anexo C: Ensamble y diseño final del sistema de control con visión artificial	135

INTRODUCCIÓN

En la actualidad las empresas dedicadas a la producción de atún buscan optimizar sus procesos y disminuir los tiempos de producción para alcanzar sus metas. Las productoras de enlatados tienen que cumplir un estricto control calidad de sus productos para exportación, donde es indispensable la verificación de cada lata de atún, en el proyecto se desarrollara un sistema para la detección de fallas en cada lata de atun mediante visión artificial, empleando algoritmos que permitan procesar información de las imágenes. Este tipo de sistemas de visión son necesarios en buscar una solución en las industrias para tener un mejor control de calidad en cada proceso.

En el sistema automatizado se empleará una banda transportadora que traslada las latas de atún hasta la cámara , al ser detectados por el sensor de proximidad, se procede a la captura de la imagen, en ese instante se realiza el procesamiento con el modelo de aprendizaje mediante el algoritmo programado en Python en la Raspberry Pi 4 , permitiendo identificar y clasificarla de acuerdo al tipo de fallas donde se verificará que el contorno del sellado de las latas de atún sea de forma óptima. El modelo de aprendizaje permite revisar el sellado hermético de cada lata, y el texto de etiqueta para obtener información relevante de cada producto.

Para obtener estos resultados se aplicará redes neuronales convolucionales y procesamiento digital de imágenes que permite la búsqueda de información para que pueda ser procesada por la cámara de visión inteligente. Luego pasará por el sistema de clasificación donde el brazo robótico mediante el método de succión extraerá las latas de atun, las que se encuentran en buen estado serán colocadas hacia el lado derecho y al lado izquierdo las latas con fallas con etiqueta impresa ilegible , sellado con falla en los contornos y ubicación inusual del abrefácil.

El capítulo I describe los antecedentes de la propuesta y justificación de la solución que se desarrollará en cada uno de los objetivos mediante la implementación del sistema de control.

El capítulo II detalla el desarrollo de la propuesta tecnológica y el análisis de las técnicas matemáticas mediante fórmulas que permiten tener resultados de lo que se requiere para el análisis del sellado hermético de las latas de atún empleando visión artificial.

El capítulo III detalla los componentes a emplear en el prototipo sean estos físicos y lógicos, también se refleja los resultados y presupuesto del prototipo.

En el capítulo IV se describe las conclusiones, recomendaciones y la bibliografía que se empleó para investigar y realizar la propuesta del proyecto.

CAPÍTULO I

1.1. ANTECEDENTES

El atún aumentó su producción en toneladas para la exportación en los periodos 2017 al 2018, según los datos que proporciona la (Cámara Ecuatoriana de Industriales y Procesadores Atuneros). El Ecuador tiene ese privilegio de estar ubicado en esa zona geográfica que permiten a las embarcaciones asegurar en un 60% de aprovisionamiento de estos productos como los peces bonitos, aleta amarilla y ojo grande, que son procesados en las provincias de Manta, Santa Elena y Posorja [1].

Las empresas del sector atunero que se dedican a este tipo de producción de enlatados de atún en la actualidad han ganado un espacio importante en sus exportaciones por la gran cantidad de volumen de su producción y por esto se ven forzados a modernizar sus industrias que se encuentran ligados a los avances tecnológicos y esto exige que se tenga niveles altos de continuidad en su producción y para cumplir con los procesos de calidad se aplican sistemas de visión artificial que puedan detectar fallas en el contorno de doble cierre, etiqueta y ubicación del abrefácil de la lata, si se detectan a tiempo el sellado de doble cierre esto evita una contaminación por infiltración, es decir, en el interior de las latas selladas se encuentren burbujas de aire.

La inteligencia artificial es un complemento a nuestra inteligencia y el motor de toda innovación debido a que ya no se trata de una tecnología del futuro sino de lo que actualmente se emplea para diferentes proyectos en la industria. Estos sistemas inteligentes pueden imitar

a la inteligencia humana donde por medio de cámaras pueden reconocer imágenes o algún patrón a partir de algún algoritmo [2].

La inteligencia artificial es una tecnología que se vuelve indispensable en la industria debido a que sus procesos se desarrollan a gran velocidad y sus controles de calidad se tienen que regir bajo la NORMA INTERNACIONAL DE ALIMENTOS CXS 70-1981 para las conservas de atún en las que se establecen los aditivos alimentarios de las conservas, el peso neto de cada producto, la existencia de materias extrañas que se encuentren en el pescado, su envase no debe tener defectos y que este se pueda cerrar herméticamente, con el sistema de visión se puedan detectar ciertos factores de calidad [3].

Las empresas en la provincia de Santa Elena que se dedican a la producción de enlatados de atún para tener un mejor control automatizado de sus procesos deberían emplear sistema de visión artificial que detecten fallas en sus líneas de producción. La inteligencia artificial juega un papel importante en las empresas porque por medio de aplicaciones o de sus algoritmos se puede evitar que una persona realice tareas repetitivas, buscando además que los empleados se les asigne otros tipos de tareas que podrían ser de mayor beneficio en el área de producción u otra.

1.2. JUSTIFICACIÓN

Datos proporcionados por el Ministerio de Comercio Exterior da a conocer que en la zona costera del Ecuador se desarrollan actividades de capturas, procesamiento y comercialización de atún siendo una principal fuente de ingresos que tiene el país aportando con un 65,48% de

divisas de este sector de la industria atunera, datos considerados entre el 2010 al 2016, representando un 80,02% en exportación de conservas de atún y el 19,98% en lomos de atún [4].

En la Provincia de Santa Elena se ubican fábricas dedicadas a la comercialización de atún, debido al incremento de exportación de este producto se considera la realización de un estudio para la factibilidad de que estos procesos de verificación sean de forma automatizada por los diferentes estándares de calidad que exigen los países que consumen este producto y esto tendría grandes beneficios a la empresa que adopte este tipo de procesos.

En la actualidad las industrias trabajan en procesos a gran velocidad por las necesidades que tiene el mercado y una forma de verificar las fallas de las latas que se pueden presentar en la producción de atún es la implementación de visión artificial para tener un mejor control del sellado hermético de cada lata.

La aplicación de este sistema también permite realizar el análisis con otros tipos de producción de enlatados como las conservas, pulpas de frutas o latas de frejoles. Se tiene conocimiento que este proceso de verificación de cierre de latas la realiza una persona en un determinado tiempo estableciendo bajo su punto de vista si la lata de atún cumple con los estándares de calidad que se exige para su comercialización. Por los problemas que se pueden presentar se plantea realizar un diseño de control que por medio de visión artificial pueda verificar mediante cámaras inteligentes el sellado de forma óptima y tener un mayor control del producto cuando sea distribuido a nivel local o de exportación a diferentes comercios.

El diseño consiste en que se analice cada lata de atún que pasa por la línea de producción y poder desviar las unidades que no se encuentren con los valores que previamente se predeterminaron. Considerando las mejoras que se plantean en la línea de producción se tendrían grandes beneficios para la empresa y que sus productos no sean devueltos y se generen pérdidas.

Los procesos en la industria al ser evaluados dependen mucho de la calidad del producto final y su etapa de control es de mucha importancia en cada uno de sus procesos. Para poder establecer la calidad de los productos dependerá de varios parámetros como su forma o la geometría del mismo y que estos mediante tareas de inspección con visión artificial se puede verificar el control de calidad.

En la actualidad, con los sistemas de visión artificial se puede realizar una inspección de toda la línea de producción, donde se puede hacer un análisis de 50 piezas por segundo con una precisión en sus mediciones. Las personas pueden entender más rápido lo que se desea realizar o comprender de algún defecto, pero los sistemas de visión artificial trabajan a largo plazo, cumpliendo procesos sin distracción y sin cansancio durante las 24 horas del día, dándonos una excelente alternativa la utilización de visión inteligente para poder realizar la detección de aquellos errores que un trabajador puede pasar por alto.

Los diferentes softwares que existen ayudan a la visión artificial a ser capaces de resolver diferentes problemas a la vez, y con su implementación de protocolos industriales permiten realizar un cambio de información o interconexión con otras tecnologías. Este proyecto surge ante la necesidad de los cambios o evoluciones que se tiene en el sector industrial, donde se

puede realizar una inspección visual artificial y que esta sea objetiva, rápida y confiable ante las fallas que se puedan presentar.

1.3. OBJETIVOS

1.3.1. Objetivo General:

- Implementar sistema de control y monitoreo en la producción final de latas de atún para la detección de defectos en el sellado de la lata utilizando métodos de visión artificial.

1.3.2. Objetivos Específicos:

- Seleccionar el algoritmo del modelo de aprendizaje para determinar fallas de borde afilado en el contorno de lata de atún.
- Adquirir datos de la etiqueta de codificación de la lata utilizando reconocimiento de texto OCR con un modelo entrenado para almacenar información de la producción de lote de atún.
- Determinar la ubicación de la anilla abrefácil mediante un modelo preentrenado inteligente para la detección de mal posicionamiento de acuerdo a las normas de productos enlatados.
- Analizar los resultados obtenidos del sistema de control con visión artificial para determinar criterios de mejoras en la implementación de nuevos procesos del sistema.

1.4. DESCRIPCIÓN DEL PROYECTO

Dentro de sus componentes importantes del sistema de visión artificial tenemos el lente que realizara la captura de la imagen para enviarla en forma de luz al sensor detector de imagen, a su vez esa luz es procesada convirtiéndola en una imagen digital para su analisis, procesador de visión mediante sus algoritmos es el que realiza el control de la imagen para extraer su información y el software para la comunicación.

El proyecto se inicia con la banda transportadora por donde pasan las latas de atún, y estas son detectadas por un sensor que está conectado al software, el sensor fotoeléctrico se encarga de enviar una señal que dará las órdenes para la detección de una lata sobre la banda, este a su vez activa la cámara de visión artificial para que detecte las fallas que se pueden presentar en la línea de producción.

Este algoritmo reconoce a través de la cámara y retorna información del reconocimiento hacia la etapa de control y si la lata procesada cumple con los parámetros establecidos, el software envía la señal a la etapa de control y con otro sensor deja pasar la lata de atún por la banda transportadora. Pero, si este producto no cumple con los estándares que se ha establecido entonces el software envía una señal al actuador externo del proceso (brazo robótico) que se encuentra al final de la banda transportadora y este al recibir una señal ejecuta el proceso de clasificación para el cual fue programado. La rutina del brazo se encuentra establecida de la siguiente manera: la banda se detiene, el efector final del brazo succiona la lata y la lleva al contenedor 1 o 2 dependiendo del resultado determinado por la inspección, luego el brazo

robótico regresa a su estado de reposo y se procede a dar marcha a la banda y se reinicia el ciclo de inspección de visión artificial.

El sistema de visión artificial permite separar los productos que tienen algún golpe y los que cumplen los estándares de producción. El diseño de inspección visual del OCR consiste en identificar la letra de la etiqueta, como primer paso realizará la captura de la imagen luego el algoritmo garantiza que se efectuó la toma de decisiones del sistema esto significa obtener información de la producción de la remesa de la lata de atun. La interconexión entre dispositivos ayudara a recoger la información como:

Lote: que permite la identificación de la fabricación del producto, su codificación varía si empieza con LA es lomo en aceite seguida de una numeración.

Hora: el tiempo de elaboración esta expresado mediante 4 dígitos, los 2 primeros números indican la hora y los dos restantes los minutos.

Línea: es la identificación del lugar de la planta de fabricación del producto, que está representada mediante números y letras.

1.5. METODOLOGÍA DE LA INVESTIGACIÓN

La metodología de la investigación reúne los procesos o métodos que se deben seguir para realizar una investigación, además se deben establecer estrategias que permitan justificar el estudio. Se debe tener información existente del tema sobre en cual se hará la investigación [5].

Cuando el investigador realiza un estudio, generalmente intenta hacer comparaciones, por lo tanto, el diseño de la investigación implica especificar la naturaleza de las comparaciones a realizar y lograr la solución del problema.

1.5.1. Investigación Exploratoria

La investigación exploratoria generalmente se lleva a cabo cuando el objetivo a explorar es un tema o problema poco estudiado, donde se tienen dudas o no se han resuelto antes o si deseamos realizar indagación sobre temas desde nuevas perspectivas o de hacer una ampliación a las existentes. La investigación exploratoria ayuda a descubrir un fenómeno relativamente desconocido [6].

Se aplica la investigación exploratoria en el proyecto porque mediante la observación se establece el estudio dentro de la planta de producción para obtener datos que permitan realizar mejoras en cada uno de los procesos investigados, para ser aplicados al sistema de visión y también establecer el control de cada lata detectada mediante el brazo robot. Con las características se obtiene una descripción de los parámetros para identificar como se encuentra la calidad de cada lata de atún.

1.5.2. Investigación Descriptiva

Este tipo de investigación describe las situaciones, recolectando datos para efectuar mediciones sobre ellas, buscando características importantes sobre algún fenómeno que se desea analizar [6].

1.5.3. Investigación Experimental

La investigación experimental define las variables que se emplearan en la investigación, como controlarlas y medirlas y cuál sería la mejor forma de definir las técnicas de analisis en función de las características de los datos recopilados [7].

El proyecto consiste en la implementación de un sistema de visión y control automatizado que permite tener una visualización de los resultados después de entrenar un modelo y determinar la detección de las fallas por medio del reconocimiento, aplicando técnicas de visión por computadora, con estos resultados cuantificables se pueden realizar correcciones en cada etapa de los procesos de producción.

CAPÍTULO II

2.1. MARCO CONTEXTUAL

Las exportaciones de las industrias procesadoras de atún del país se han triplicado en los últimos 10 años, según FedExport. En 2010, el país exportó 318 millones de dólares, alcanzando los 1033 millones de dólares a fines del 2020 [8]. En la provincia de Santa Elena se encuentran 3 empresas que se dedican al procesamiento de conservas de latas de atún, su capacidad de producción comprende las 500.000 toneladas de atún no procesado. Estas industrias en la provincia de Santa Elena mantienen parte de sus procesos de forma manual, donde estas actividades de control de calidad las realizan el personal que labora en estas empresas [4].

La propuesta de implementar un sistema automatizado para realizar el control del sellado de las latas de atún, busca obtener datos en cada una de las inspecciones que se realizan en las líneas de producción, esto brinda a las empresas controlar los lotes de sus productos en casos de que se presenten fallas de sellado hermético de las latas de atún, evitando que esto pueda disminuir costos relacionados a la devolución de estos productos, tener automatizados sus procesos generarían más producción, representando un crecimiento económico en el sector atunero del país.

Se emplea un sistema de visión automático que determinara las fallas que se presenten en la línea de producción separando las latas que no se encuentren debidamente selladas o que

carecen del texto y poder obtener estos datos para realizar las correcciones en la etapa del proceso donde ocurrió.

2.2. MARCO CONCEPTUAL

2.2.1. Materia Prima

2.2.1.1. El Atun

El atún tiene un cuerpo de cabeza piramidal y una boca muy pequeña, esto le permite ser excelentes nadadores, su dorso es de color azul, miden alrededor de 1 metro y su peso puede alcanzar hasta los 35 kg , su alimentación es a base de peces, calamares y crustáceos. Dentro de sus especies más importantes para el enlatado tenemos el atún albacora, atun bonito (Ver Figura 1), atún aleta amarilla y el atún ojo grande [9].



Figura 1: Atun Bonito

2.2.2. Envase Lata de Atun

El tipo de envases de hojalata es el más utilizado en la industria de conservas por poseer características de rigidez, resistencia al choque térmico, buena conductividad térmica, son de bajo costo y poseen ausencia de toxicidad [10].

Características de las latas de atún.

- Herméticas: están protegidas del aire, oxígeno y bacterias que contaminen el contenido, al tener esta característica permite conservar sus valores nutricionales.
- Protección del contenido: están protegidos contra la luz y poseen una alta resistencia a la rotura.
- Poseen una tapa de fácil apertura, estas se embalan en las talas, por medio de un remache(Ver Figura 2).



Figura 2: Lata de Atun

2.2.2.1. Partes de la tapa del envase

Es la parte que tiene el envase para permitir el cierre hermético de la lata, este compuesto por:

Curva de la tapa: Es el extremo de la tapa doblado hacia afuera que permite la unión con el gancho cuerpo.

Doble cierre o sello: Es la unión entre la pestaña del cuerpo y el gancho tapa de la lata, este doble cierre posee tres espesores del componente de tapa y dos espesores del cuerpo del envase, que se aseguran con un sellador en todo el contorno de la pestaña doblada para poder asegurar la hermeticidad de la tapa(Ver Figura 3).

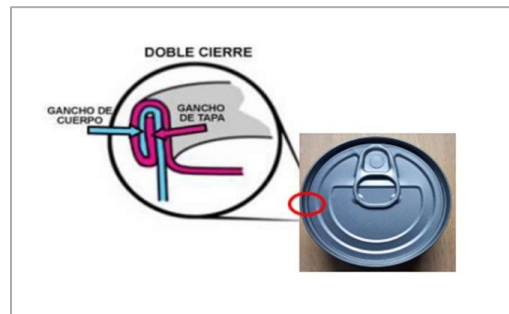


Figura 3: Doble Cierre del Sellado de Lata

Compuesto por dos operaciones:

Primera Operación: La tapa se entrelaza con la pestaña del cuerpo de la lata, donde hay que considerar que no se encuentre apretado o suelto y el gancho del cuerpo debe quedar paralelo al gancho de la tapa.

Segunda Operación: En esta operación se consigue aplanar el primer sello para que se puede adherir bien el compuesto sellador, donde posterior se comprimen los enganches preformados y estiran los pliegues del gancho tapa [11].

2.2.3. Fallas en las latas de atún

Cuando existen fallas afectan directamente a la calidad de un producto, generando a las empresas costos por la reprocesamiento, donde también se generan multas por los entes reguladores y un daño a la marca que se distribuye.

Hay que considerar que las operaciones de la planta en la producción trabajan 24 horas, los 7 días de la semana y esto puede provocar que no se pueda realizar el reprocesamiento.

2.2.3.1. Sellado afilado

En la parte superior interna del sello, se tiene una orilla afilada(Ver Figura 4), esto se produce al forzar la tapa cuando se realiza el doble sello [11].

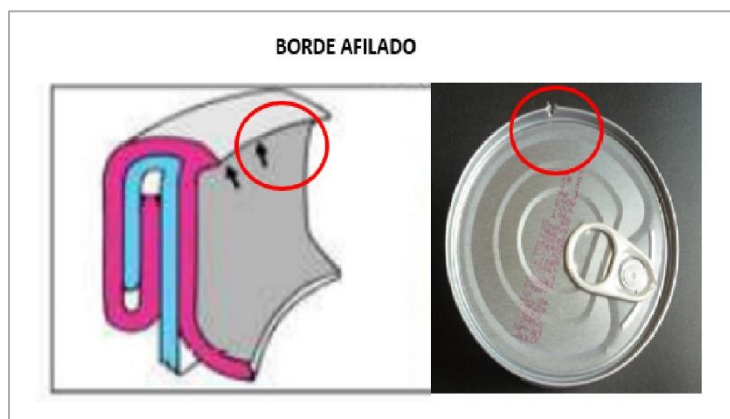


Figura 4: Borde de Sellado Afilado

2.2.3.2. Anilla del abrefácil mal posicionado

El metal sufre una ruptura en la parte superior interna del sello justo donde se produce la orilla afilada(Ver Figura 5).



Figura 5: Anilla de Abrefácil mal posicionado

2.2.3.3. Etiqueta de codificación no impresa

No se encuentra la etiqueta codificada que contiene los siguientes datos como el lote, fecha de elaboración, caducidad y fábrica de producción(Ver Figura 6).



Figura 6: Etiqueta de Codificación no impresa

2.2.4. Control de cierres

Se verifica el cierre hermético del envase, considerando los siguientes parámetros:

2.2.4.1. Espacio libre

Esta comprendido entre los dos espesores del gancho tapa y los tres espesores del gancho cuerpo, si se tiene un espacio muy grande existe un vacío y esto provoca un peligro para mantener la hermeticidad del cierre de la lata.

2.2.4.2. Traslape

Los bordes del gancho cuerpo y el gancho tapa debe estar junto, de esta manera se asegura que el sellador se comprima entre ellos y tener un espesor de cierre correcto [11].

2.3. SISTEMA DE AUTOMATIZACIÓN

La automatización es un sistema que realiza tareas de producción generalmente realizada por humanos mediante dispositivos tecnológicos. El sistema de automatización consta de dos partes como el control y la parte de operación, que trabaja directamente en la máquina, permitiendo que se mueva o haga otro tipo de operación que se dese gracias a los actuadores y sensores. La parte de control la compone un autómata que tiene la capacidad de comunicación con otros componentes del sistema [12].

El ser humano ha tratado de crear maquinas que ayuden en las tareas más peligrosas, pesadas y repetitivas. En la actualidad existen maquinas sofisticadas que son aplicadas a los procesos de automatización y de control en tareas de procesamiento diario en las empresas, consiguiendo liberar al ser humano de ciertas tareas.

2.4. SISTEMA DE TRANSPORTE

Los sistemas de transporte son esenciales en las plantas de fabricación, permitiendo el movimiento automatizado de manera eficiente de los productos, aumentan la productividad en las empresas, reduciendo la siniestralidad o rotura de los productos. Todos estos beneficios se tendrán instalando en la planta sistemas modernizados de transporte [13].

2.4.1. CINTA TRANSPORTADORA INDUSTRIAL

Las bandas transportadoras industriales son mecanismos utilizados para transportar productos, consiste en una banda continua que se mueve entre dos rodillos. Posee una base móvil que gira constantemente mientras encima contine productos que son movilizadods de un lugar a otro.

Están diseñadas para el manejo y transporte de productos en agroindustria, farmacéutica, automotriz u otro sector industrial que desea movilizar grandes volúmenes. Las bandas transportadoras son versátiles y se pueden utilizar en una variedad de aplicaciones, en las que se pueden realizar trabajos con movimiento inclinados, verticales u horizontales dependiendo del material que se transporte [14].

2.5. SISTEMA ELECTRÓNICO

Un sistema electrónico es un grupo de circuitos que operan con señales eléctricas, posee una etapa de entrada donde se recogen los datos del exterior (luz, humedad, movimiento,

temperatura) y la etapa de control en el que se interpretan, controlan e indican el resultado permitiendo activar o desactivar los dispositivos de salida [15].

2.5.1. SENSOR DE PROXIMIDAD

Los sensores de proximidad son ampliamente utilizados en aplicaciones industriales y de fabricación. Se utilizan para detectar objetos o materiales que luego inician una acción o simplemente notan su presencia o ausencia. Una de sus características es que no se requiere un contacto físico con el objeto que se desea detectar, por lo que reciben el nombre de sensores sin contacto [16].

2.6. SISTEMA DE CONTROL

Garantiza la calidad de los productos, su fabricación o diseño requiere de este tipo de procesos para lograr ciertos objetivos, posee un efectivo control de los envases de la línea de producción. Su funcionamiento se basa en controlar o regular otro tipo de sistema para evitar el mal funcionamiento y que esta sea precisa y fiable en las líneas de producción [17].

2.7. SISTEMA DE VISIÓN ARTIFICIAL

La Visión Artificial se emplea en diferentes sectores de las industrias, emplean sistemas de imágenes digitales para determinar el control de calidad, mediante su sistema de inspección y poder alcanzar los requisitos mínimos de calidad y obtener un informe de posibles errores del control total de la producción. La Visión Artificial se basa en imágenes que mediante capturas de la cámara de visión industrial son procesadas por un software específico de visión

artificial, siendo estas capaces de verificar, contar, medir, seleccionar e identificar fallos, también cuentan con la capacidad de tomar ciertas decisiones como expulsar un producto de la línea de producción cuando esta no cumpla con los estándares de calidad que se requiere [18].

2.7.1. Fases del sistema de visión artificial

FASE 1: en esta fase por medio del sensor se capta la imagen digital del objeto.

FASE 2: se separa las partes que no son útiles, y quedamos con las partes que se requiere de la imagen.

FASE 3: se realiza el procesamiento de la imagen mediante un algoritmo

FASE 4: Se realiza la segmentación de los productos dependiendo de las características que se desea analizar.

2.8. PROCESAMIENTO DIGITAL DE IMÁGENES

2.8.1. Morfología matemática

Para procesar las imágenes de forma digital se las relaciona con principios algebraicos y geométricos. Poseen dos operaciones la de dilatación y de erosión

La erosión $\varepsilon_B(f)$ la obtenemos del menor valor del elemento estructurante B, ecuación (1).

$$\varepsilon_B(f)(r, c) = \{d, l) \in B\} \quad (1)$$

En la dilatación δ_B la obtenemos con el elemento de mayor valor del dominio del elemento estructurante B, ecuación (2).

$$\delta_B(f)(r, c) = \{d, l) \in B\} \quad (2)$$

Dentro de la morfología matemática también se tiene la apertura y clausura estas se las obtiene de combinar la erosión y la dilatación.

La apertura morfológica de una imagen f por el elemento estructurante B , ecuación (3), definiéndose como la erosión de f por B , seguida de la dilatación por el mismo elemento estructurante.

$$f \circ B = \delta(\varepsilon(f, B), B) \quad (3)$$

La clausura de una imagen f por un elemento estructurante B , ecuación (4), se la define como la dilatación de f por B , seguida de la erosión por el mismo elemento estructurante [19].

$$f \cdot B = \varepsilon(\delta(f, B), B) \quad (4)$$

El efecto de realizar la dilatación en grises está dado por:

- Al ser positivos todos los números del elemento estructural, proporciona una imagen con más brillo que la original es decir es más clara.
- Pero si varían los valores como la forma del elemento estructural, los detalles que son oscuros pueden reducirse o eliminarse.

El efecto de emplear erosión en grises tiene como características:

- Al tener una imagen más oscura que la original al ser positivos los elementos del valor estructural.
- El detalle claro depende tanto del valor como la forma del elemento estructural para reducirse o eliminarse.

2.9. SEGMENTACIÓN DE IMÁGENES

Para procesar las imágenes digitales se debe emplear la segmentación con técnicas basados en los valores de pixel.

2.9.1. Técnica de umbralización

Es una técnica que emplea valores por pixel y dependiendo de la información de un pixel local pueden tomar decisiones. Si la intensidad de un objeto cae fuera del ángulo recto del fondo de la imagen estas son más eficientes [20].

En esta parte lo que se busca es aislar los puntos de la imagen del objeto que se desea evaluar, es decir detectar los bordes de forma automática de las latas de atún. Se realiza la

segmentación por ser la parte del algoritmo en la que se puede separar, las formas que puede tener el objeto y esto ayudaría a la identificación de las latas, para luego realizar la extracción y obtener el objeto que se desea evaluar.

Se realiza la segmentación por umbralización que es la empleada en aplicaciones industriales, por ser rápida en su ejecución, cuando se aplica un umbral dicha imagen queda binarizada, esto da como resultado que los valores de los objetos que se representan con 1 sean diferentes a los pixeles de fondo que se los representaría con 0.

2.10. DETECTAR BORDES DE IMÁGENES EN ESCALA DE GRISES

Cuando se realiza detección de bordes se debe cumplir lo siguiente:

1. Para reducir el ruido se realiza un suavizado de la imagen que se quiere analizar.
2. Se detectan los posibles puntos del borde.
3. Solo los puntos que representan el borde se deben seleccionar.

2.10.1. Detectar bordes de imágenes en escala de grises: Operadores gradiente

Para una función discreta la derivada en un punto u puede ser calculada a partir de la diferencia existente entre los puntos vecinos de u dividido por el valor de muestreo entre ambos puntos. La derivada se aproxima con la siguiente ecuación (5), empleándola también para el sentido vertical v , a lo largo de la columna de la imagen [21].

$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} \quad (5)$$

Entonces, si $\frac{\partial I}{\partial x}(x, y)$ y $\frac{\partial I}{\partial y}(x, y)$ expresan la derivada parcial de la función de la imagen $I(u, v)$ con respecto a la variable u o v . Entonces el vector del gradiente de la función I en un punto (x, y) dado por la ecuación (6).

$$\nabla I(x, y) = \left[\frac{\partial I}{\partial x}(x, y) \quad \frac{\partial I}{\partial y}(x, y) \right] \quad (6)$$

Mediante la siguiente ecuación (7) se define la magnitud del gradiente.

$$|\nabla I| = \sqrt{(\nabla_x)^2 + (\nabla_y)^2} \quad (7)$$

Para realizar el cálculo de la dirección del gradiente en cada pixel se emplea la siguiente ecuación (8).

$$(\nabla_x, \nabla_y) \quad (8)$$

Se emplea el operador de Sobel para el cálculo de la primera derivada dando como resultado la detección de bordes mediante las siguientes matrices, ecuación (9).

$$\begin{aligned}
 H_x^S &= [-1 \ 0 \ 1 \ -2 \ 0 \ 2 \ -1 \ 0 \ 1] \text{ y } H_y^S \\
 &= [-1 \ -2 \ -1 \ 0 \ 0 \ 0 \ 1 \ 2 \ 1]
 \end{aligned}
 \tag{9}$$

El filtro de Sobel tiene un mayor peso en la columna central del filtro, este operador da buenos resultados y son fácil de implementar en la mayoría de software para el procesamiento de imágenes digitales [21].

2.11. Técnicas matemáticas de comprensión de video para mejorar la imagen

2.11.1. Suavizado

Es una técnica matemática aplicada a las matrices, permite reducir el ruido, mejorando la claridad del contorno de la imagen.

2.11.2. Expansión del histograma

Los valores mayores suben al máximo, los intermedios permanecen y los menores llegan al mínimo, mediante la siguiente ecuación (10) para mejorar la calidad de la imagen.

$$I_{T(i,j)} = \left(\frac{I_{o(i,j)} - I_{o(i,j)min}}{I_{o(i,j)max} - I_{o(i,j)min}} \right) 2^{N^{\circ}bpp} - 1 - \frac{I_{o(i,j)min}}{I_{o(i,j)max}}
 \tag{10}$$

Sin el factor adicional:

Si consideramos $I_o=15$, da un $I_t= 10.44$ y con $I_o= 250$, resulta $I_t = 255$

Ejemplo:

$$\begin{aligned} [15 \ 15 \ 120 \ 15 \ 10 \ 120 \ 4 \ 250 \ 4 \ 4 \ 4 \ 4] 2^{N^{\circ}bpp} &= 2^8 \\ &= 256 \text{ nivel de gris} \end{aligned} \quad (11)$$

0= negro, 128=gris,255=blanco

Con el factor adicional:

Considerando $I_o=15$, da un $I_t=10.38$, ecuación (12), tenemos un mejor resultado al ser de un valor menor que el pixel anterior donde se acerca a 0 que representa al negro, y tenemos una mejor calidad, ecuación (11). Con $I_o=250$, resulta $I_t=254.9$, ecuación (13) donde se aproxima a 255 que representaría el color blanco [22].

$$\left(\frac{15 - 4}{250 - 4} \right) 256 - 1 - \frac{15}{250} = 10.38 \quad (12)$$

$$\left(\frac{250 - 4}{250 - 4} \right) 256 - 1 - \frac{15}{250} = 254.94 \quad (13)$$

2.11.3. Manipulación del histograma

Se suma un valor positivo ($D>0$) a una matriz para incrementar los valores de los pixeles para aclarar o un negativo ($D<0$) para oscurecer la imagen, pixel por pixel.

La DCT.

Mejora la comprensión, y la secuencia de imágenes mediante un algoritmo matemático.

2.11.4. Ecuación del Histograma

Si aplicamos la ecuación del histograma conseguimos mejorar el contraste de la imagen, es decir se reparte de una forma uniforme los valores del histograma.

Si se tiene una imagen $M \times N$, con un n_k pixeles por cada nivel r_k : $H(r_k) = n_k$

Al aplicar la ecuación del histograma realizamos la transformación sobre los niveles de intensidad de la imagen ($k=0,1,2, \dots, L-1$) mediante la ecuación (14).

$$S_k = (L - 1) \sum_{j=0}^k \frac{n_j}{n} \quad (14)$$

Mapear cada pixel con intensidad r_k . de la imagen de entrada con su valor transformado s_k en la imagen de salida [23].

2.12. DETECCIÓN DE LOS REMACHES.

Primero se realiza una eliminación de los puntos de alta reflectancia, esto se logra por medio de secuencia de procesos morfológicos de apertura y cierre. Luego cuando ya se tiene identificado la agrupación de puntos de lo referente a la imagen este se identifica mediante un algoritmo de componente conectado, siendo estos descriptores de área y circularidad, refiriéndose a los descriptores como el número de puntos que se conectan al componente y la circularidad que se puede calcular mediante la siguiente fórmula, ecuación (15).

$$C_s = \frac{4 * \pi * A_s}{P_s^2} \quad (15)$$

En donde A_s es el área de la forma del componente y P_s sería su perímetro, y mediante este resultado se puede tener una referencia de tener un círculo perfecto.

Se elige este tipo de descriptores debido a que su traslación y rotación son invariantes siendo esta una característica importante, esto da como resultado de que tanto su posición como su orientación sean las mismas cuando se realice la inspección en cada uno de los externos de las latas.

Para el descriptor de circularidad se lo selecciona debido a que se pueden presentar defectos como fractura o abolladuras de los remaches, y estos factores provocan que se tenga cambios en su forma original del círculo de la lata.

2.13. RECONOCIMIENTO DE TEXTO OCR

La tecnología OCR utiliza algoritmos para el análisis de imágenes y reconocer patrones de caracteres que representan letras, números u otros símbolos donde son comparados con una plantilla asignada en una base de datos. Su aprendizaje lo realizan por medio de una red neuronal siendo capaces del reconocimiento de caracteres en diferentes posiciones, claridad de la imagen o ángulo [24]. Para tener una identificación de la producción se establecen aspectos como la serialización y trazabilidad de la industria alimentaria, en la gráfica

podemos visualizar un reconocimiento de texto OCR (Ver Figura 7), para tener una lectura clara de la etiqueta de cada lata de atún.

Para la detección del contenido del envasado de atún, también se puede realizar por medio de la litografía estableciendo un control en el envase o en el sellado, en la que podemos detectar fallas como manchas o que la lata tenga cambio de color, rayones, abolladuras o que se presente ausencia de la colocación del anillo de fácil apertura.

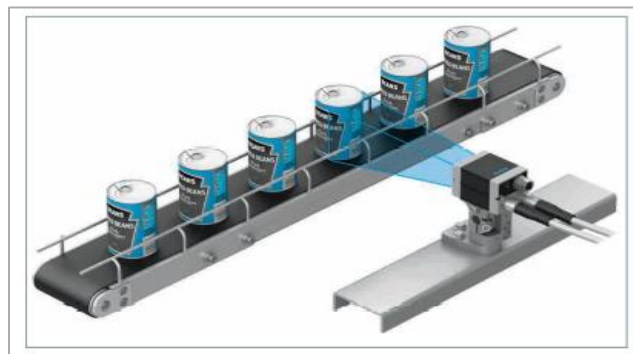


Figura 7: Sistema de Visualización Código OCR

2.14. DESARROLLO DEL SISTEMA DE VISIÓN ARTIFICIAL

Para realizar el desarrollo de un sistema de visión artificial se deben considerar aspectos de cómo está constituido: el prototipo posee un sistema de control para clasificar a las latas dependiendo de las fallas de calidad y con el software de entrenamiento se procesarán las imágenes y sensores para detectar la presencia de latas de atun, además debe contar con iluminación que es empleada para corregir el brillo del objeto que se detectara y tener una inspección optima.

2.14.1. SELECCIÓN DE LA CÁMARA

2.14.1.1. Tamaño de detección de objeto

Por la capacidad de detección de defectos pequeños se requiere una cámara de 2 megapíxeles, ya que estas pueden detectar con una mayor estabilidad. Para realizar el cálculo mínimo de detección se debe considerar el número de píxeles, su campo de visión y el tamaño mínimo de detección de píxeles (Ver Figura 8).

A=número de píxeles, rango de captura de la cámara (eje Y)

B=campo de visión (eje Y) 20mm

C= tamaño mínimo de detección de píxeles para elementos de sensor de imagen

Tamaño mínimo de detección = $B / A \times C$

Tamaño mínimo de detección = $20 \text{ mm} / 1200 \text{ píxeles} \times 4 \text{ píxeles} = 0.066 \text{ mm}$

Si el objetivo de inspección a detectar es de 0.1 mm se debe seleccionar una cámara de 2 megapíxeles o superior.

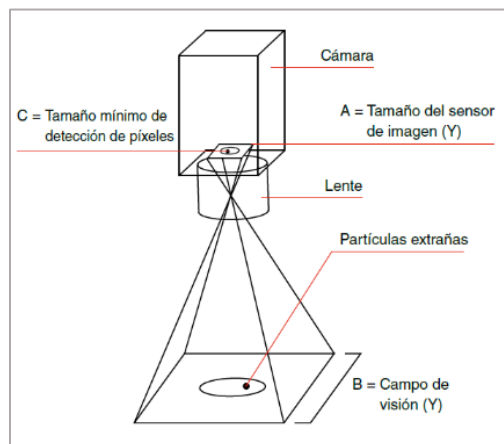


Figura 8: *Tamaño de Detección de objeto*

2.14.1.2. Selección de cámara según el área de detección

Si el número de píxeles es alto se pueden detectar objetos más pequeños, pero estas tienen un costo alto, se debe considerar elegir una cámara que como requisito tenga el tamaño mínimo de detección para tener una buena inspección (Ver Tabla 1).

Tabla 1:Datos de Resolución de Megapíxeles

	Resolución	Rango de captura	
Alta Precisión	21 megapíxeles	5104 x 4092 píxeles	Color/Monocromática
	5 megapíxeles	2432 x 2040 píxeles	Color/Monocromática
	2 megapíxeles	1600 x 1200 píxeles	Color/Monocromática
	0.47 megapíxeles	784 x 596 píxeles	Color/Monocromática
Bajo costo	0.31 megapíxeles	640 x 480 píxeles	Color/Monocromática

2.14.1.3. Tiempo de inspección

Como se realizará inspección de imágenes la velocidad de procesamiento debe ser rápida, pero esto depende lo que se desea inspeccionar, para que se puedan realizar más inspecciones en menos segundos, esto también influye en la resolución de la cámara, dependiendo de su procesamiento y el número de objetos a detectar.

2.15. TIPOS DE APRENDIZAJE

2.15.1. Aprendizaje Supervisado

En los algoritmos de aprendizaje supervisado, se genera un modelo predictivo, cuyos datos están basados en entradas y salidas. Entrenamos nuestro modelo con cientos de imágenes clasificadas, al generarse una nueva imagen, nuestro modelo clasificara de acuerdo a los datos preetiquetados. Para este modelo de aprendizaje los resultados del algoritmo deben ser conocidos y los datos utilizados con los que se entrenó el algoritmo se encuentren etiquetados [25].

2.15.1.1. Aprendizaje No supervisado

Estos algoritmos solo adaptan sus modelos predictivos en función de los datos de entrada, independientemente de los datos de salida, en este tipo de aprendizaje los datos de entrada no se clasifican (Ver Figura 9), no se consideran estas características para entrenar el modelo. El principal objetivo del aprendizaje no supervisado (Ver Figura 10), es comprender y abstraer patrones directamente de la información [25].

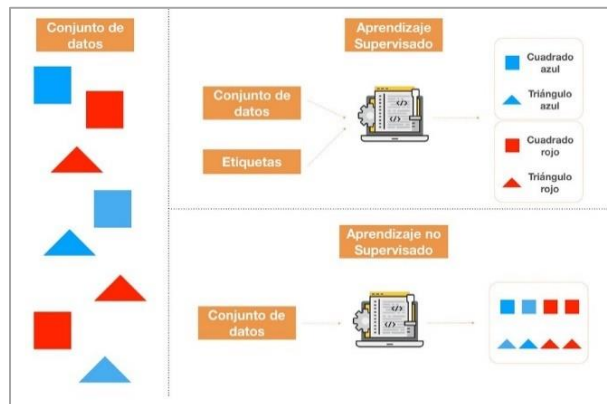


Figura 9: Tipos de Aprendizaje Supervisado y no Supervisado

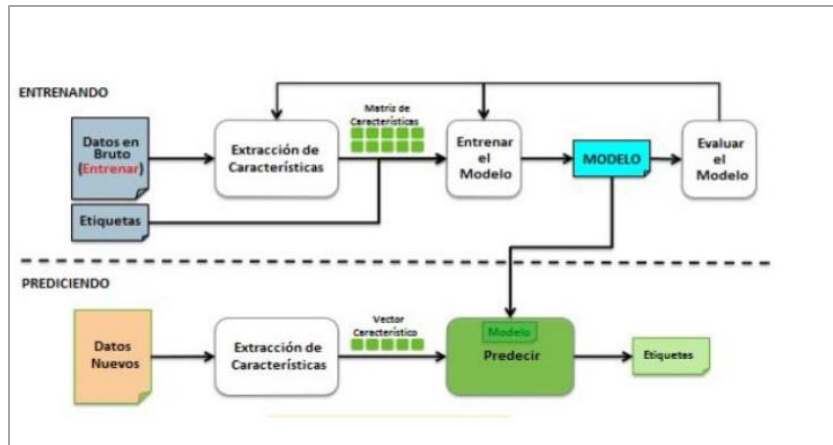


Figura 10: Entrenamiento de Sistema Autónomo y Eficiente

2.15.2. Método de Aprendizaje

2.15.2.1. Deep Learning

El Deep learning es un aprendizaje profundo no supervisado del aprendizaje automático (Ver Figura 11), donde se solicitan los datos de entrenamiento. Basadas en unas redes neuronales jerárquicas que posee capas especializadas de unidades de procesamiento, para identificar características específicas. Hay conjuntos de neuronas que están especializadas para realizar ciertas tareas como los bordes de un objeto [26].

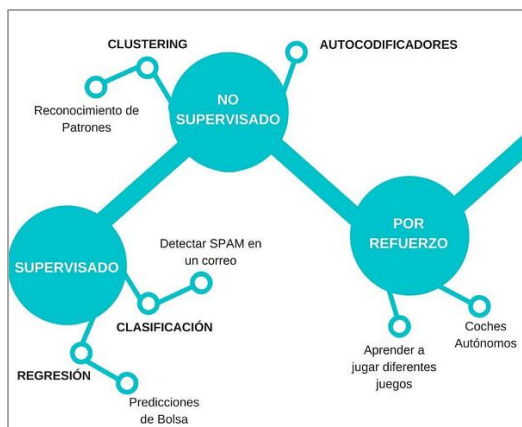


Figura 11: Métodos de Aprendizaje

2.15.2.2. YOLOv5

Este es un modelo de visión por computadora empleado para realizar detección de objetos, dependiendo de su versión estas ofrecen una precisión de la más pequeña(s) hasta la extra grande(x). Su entrenamiento puede variar en la cantidad de tiempo empleado, esta depende de la versión que se utilice. En la imagen se visualiza las variantes que tiene YOLOv5, su entrenamiento es más rápido que el de EfficientDet [27]. El YOLOv5x es el más preciso donde al procesar varias imágenes lo realiza de forma rápida (Ver Figura 12).

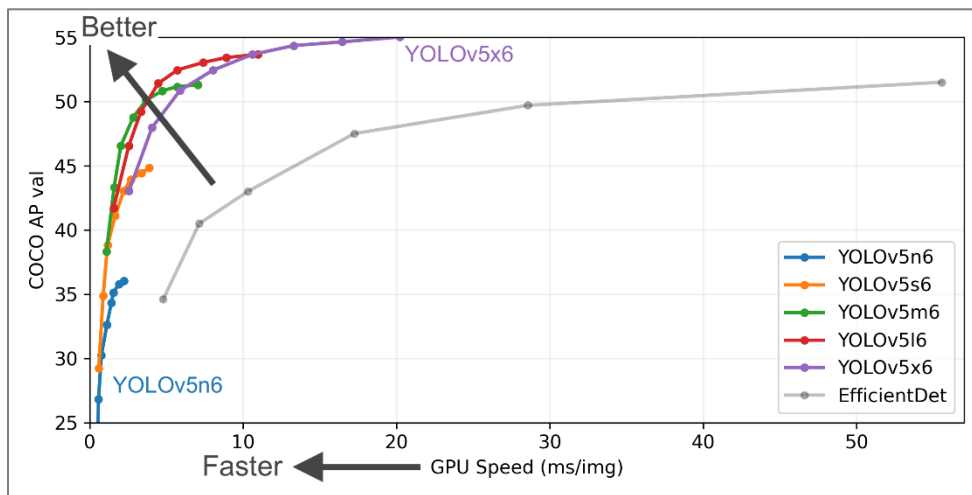


Figura 12: Versiones de YOLOv5 para el entrenamiento

2.15.3. Redes Neuronales Convolucionales

Una red neuronal se define como un modelo computacional formado por un conjunto de unidades básicas denominadas neuronas artificiales, normalmente dispuestas en capas. Sus neuronas están interconectadas entre sí, estas pueden enviar una señal que consta de valores numéricos [28].

Usando una cadena, una neurona puede multiplicar la salida de la neurona anterior por un valor de peso o suprimir el estado de activación de una neurona adyacente. La organización jerárquica se divide en tres partes: una capa de entrada que detecta y recibe datos, una o más capas ocultas y una capa de salida que representa la activación de clases [28]. Los sistemas de aprendizaje automático no son fáciles de programar, pero deben entrenarse en un conjunto de muestras etiquetables, por lo que las redes neuronales apuntan a resolver problemas como lo haría el cerebro humano (Ver Figura 13).

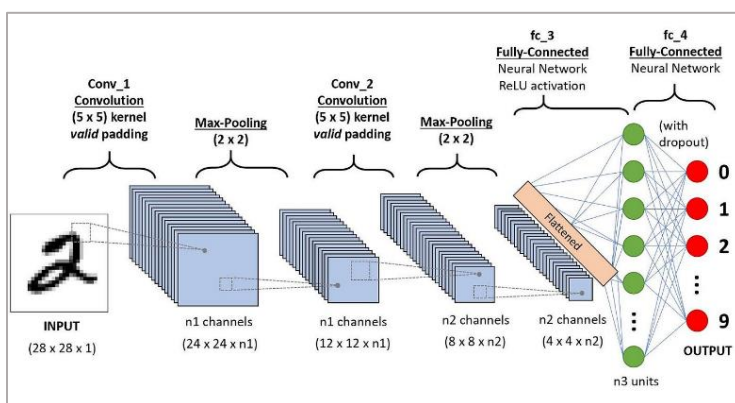


Figura 13: Red Neuronal Convolutiva

2.15.4. Segmentación Semántica

La segmentación semántica es una técnica de aprendizaje profundo que posee una ventaja sobre las otras técnicas de aprendizaje, como la capacidad de aprender mediante representaciones de características. Este tipo de técnicas consiste en otorgarle una etiqueta a cada pixel de la imagen, el objetivo principal es delimitar los objetos de las diferentes categorías a nivel de pixel [29]. Estas anotaciones se las puede realizar en VGG Image Annotator , makesense.ai, Roboflow o LabelImg (Ver Figura 14).

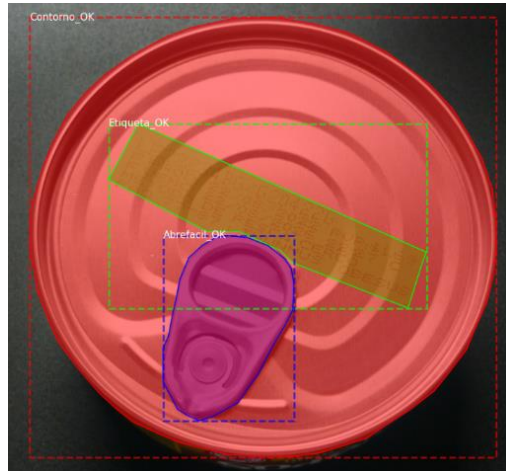


Figura 14: Etiqueta segmentación de Imagen

2.16. MARCO TEORICO

A continuación, Se detallan las tesis de titulación, que se encuentran relacionadas con la presente propuesta, cuyo objetivo es de obtener información para el análisis y realizar la propuesta tecnológica.

En el año 2019, Icaza Paredes Washington Ángel, de la Universidad Estatal de Milagro, mediante su tema de titulación, “Desarrollo de sistema de visión artificial con LabVIEW, que permita detectar imperfecciones en las latas de atún”, realizo sobre una maqueta el prototipo de un sistema de visión artificial funcional que permite la detección de fallas en productos enlatados mediante la adquisición de imágenes y su procesamiento en tiempo real [30].

El presente trabajo de titulación “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” realizado en la Escuela Superior Politécnica de Chimborazo, en

el año 2020, por Wilson Sigifredo Palma Santacruz, diseñó un sistema de visión artificial capaz de determinar las fallas de codificado de las latas de atún a tiempo real tomando como base el procesamiento digital de imágenes empleando filtros y algoritmos mediante un modelo de red neuronal artificial [31].

En el año 2017, Tierra Gusqui Juan José y Galarza Duchi Marco Vinicio, de la Escuela Superior Politécnica de Chimborazo, mediante su propuesta de titulación, “Implementación de un sistema de control y monitoreo en base al procesamiento de imágenes digitales en los sistemas de visión artificial aplicado al reconocimiento de la maquina selectora de botellas en el laboratorio de la Escuela de Ingeniería Industrial de la ESPOCH”, realizo un sistema de visión artificial que se aplicó a una maquina selectora de botellas, que se encontraba en el laboratorio de Automatización, con la finalidad de aportar nueva tecnología y proporcionar la posibilidad de manipulación de este tipo de sistemas fortaleciendo el nivel de aprendizaje de futuras generaciones [32].

En el 2022, Nick Bryan Branque Rosales, de la Universidad Estatal Península de Santa Elena, con su propuesta de titulación, “Implementación de un sistema automatizado, mediante el uso de visión artificial para la clasificación del maracuyá, según su color de madurez y el uso de un sistema scada para el monitoreo de la productividad”, implemento un sistema automatizado, para detectar el estado de madurez de la maracuyá, mediante visión artificial, programado en una Raspberry pi 4, con un sistema de transporte para llevar la maracuyá hacia el sistema de visión y pasar por el de clasificación y un sistema de monitoreo de la productividad [33].

CAPITULO III

3.1. COMPONENTES DE LA PROPUESTA

Para la elaboración de la propuesta tecnológica se necesita componentes físicos y componentes lógicos, que se describen en esta sección.

3.1.1. Componentes Físicos

Para realizar la implementación del sistema de control, se utilizará dispositivos eléctricos y electrónicos que permiten realizar el proceso de detección de calidad de la lata mediante el sistema de visión artificial detallados a continuación.

3.1.1.1. Cámara Raspberry Pi HQ

La cámara Raspberry Pi HQ (Ver Figura 15) posee un sensor Sony IMX477 que proporciona una resolución de 12,3 MPx, dándonos una mayor sensibilidad a la luz. Son versátiles que permiten realizar trabajos con lentes intercambiables que poseen montura CS. Es una cámara de alta calidad por lo que son perfectas para trabajos en el área industrial, compatibles con cualquier modelo Raspberry Pi [34].

Características.

- ✓ Sensor Sony IMX477, retroiluminado, 12,3 megapíxeles.
- ✓ Salida: RAW 12/10/8, COMP8.
- ✓ Estándares de lentes: montura CS y C.
- ✓ Longitud del cable plano: 200 mm.

- ✓ Enfoque posterior: ajustable (12,5 mm – 22,4 mm).
- ✓ Filtro de corte IR: integrado.



Figura 15: Cámara Raspberry PI High Quality

3.1.1.2. Cinta transportadora

Es un sistema para el manejo de materiales y componentes mediante un proceso eficiente y sin esfuerzo, permitiendo el ahorro de tiempo, energía y costo. Las medidas de esta cinta transportadora son 150 cm de largo, 74 cm de alto y 20 cm de ancho (Ver Figura 16). En el proyecto se empleará este sistema para transportar las latas de atún, su velocidad estará controlada por un motor drive DC, luego la lata pasara al sistema de visión artificial para ser inspeccionado hasta llegar al brazo robótico donde se clasificará las fallas que se presenten en el proceso de producción cumpliendo así el control sobre la lata.

La cinta industrial de PVC es antiestática y resistente al desgaste (Ver Tabla 2), la tensión de la correa se puede ajustar dependiendo de las necesidades del proceso, este transportador es de dos vías donde se puede ajustar su dirección y velocidad, es ampliamente utilizada en el transporte de los alimentos, botellas de plástico, de vidrio, metal y cartones [35].



Figura 16: Banda Transportadora Vevor

Tabla 2: Características de banda transportadora

Características	
Material	Acero inoxidable
Tamaño del cinturón (largo x ancho)	59 x 7,8 pulgadas/150 x 20cm
Grosor de la correa	0.1 pulg. /2.5 mm
Longitud del transportador	46 in / 117 cm
Ancho del transportador	9.8 in / 25 cm
Altura del transportador	29 in / 74 cm
Altura ajustable de la baranda	0-6 in / 0-15 cm
Peso del producto	42 libras / 19 kg

3.1.1.3. Motorreductor

Es un motor de cobre (Ver Figura 17), proporciona una gran potencia confiable, permitiendo un transporte más estable, tienen un ahorro de energía reduciendo su consumo. La velocidad se puede ajustar dentro de los 30-120 r/min (Ver Tabla 3), esto da como resultado mejorar su eficiencia de trabajo [36].



Figura 17: Motorreductor

Tabla 3: Especificaciones Técnicas de motor

Especificaciones del Motorreductor	
Voltaje nominal	110 V
Frecuencia	60 Hz
Potencia	60 W
Motor	Motorreductor
Velocidad ajustable	30 – 120 r/min

3.1.1.4. Speed Control US-52

Es un controlador para motor monofásico de corriente alterna (Ver Figura 18), es de alta precisión, su circuito electrónico permite lograr una velocidad constante de retroalimentación con la función de variar su velocidad, es utilizado en diferentes procesos de industrias [37].



Figura 18: Speed Control Motor

3.1.1.5. Brazo Robótico ARM2

La estructura del brazo robótico es metálica y para que su operación tenga mayor estabilidad las articulaciones están fijadas con cojinetes de alta precisión. El brazo robótico de 6 grados de libertad posee una bomba de succión, usa 4 servos mg996r de direcciones digitales y 2 servos mg90s para garantizar una mayor potencia (Ver Figura 19), también cuenta con una placa UNO R3 y la placa de controlador de motor [38].

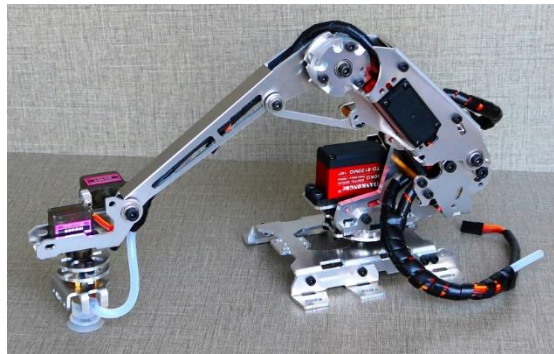


Figura 19: Brazo Robótico ARM-N2

3.1.1.6. Servomotor

El servo MG996R es un motor Dc (Ver Figura 20), que tiene la capacidad de ubicar el eje en un ángulo determinado, posee una caja reductora la cual le aumenta el torque y reduce la

velocidad, con la tarjeta electrónica y el potenciómetro forman un control de lazo cerrado, se puede trabajar con Arduino, Raspberry pi y otros microcontroladores. Su rotación es de 180° grados (Ver Tabla 4), su posición de 0° necesita un pulso de 0.6 ms, para el de 90° necesita 1.5 ms y para el de 180° requiere de 2.4 ms [39]. Los cables para su conexión se detallan de la siguiente forma:

Café=tierra(GND), Rojo=VCC(5V), Naranja=señal de control(PWM)



Figura 20: Servomotor MG996R

Tabla 4 : Especificaciones Técnicas Servomotor MG996R

Especificaciones Técnicas	
Modelo	MG996R
Voltaje de alimentacion	6.0 V – 7.2 V DC
Stall Torque	9.4kg/cm(4.8V)
Angulo de rotación	0-180°
Rango de temperatura	-30 a +60 °C
Peso	55 gramos
Ancho de pulso	500us y 2400us
Periodo de pulso	20ms

3.1.1.7. Mini Bomba de Vacío

Es una bomba de vacío eléctrica con alto flujo de aire (Ver Figura 21), trabaja con un voltaje de 12 V DC (Ver Tabla 5),sirve para suministrar aire suficiente produciendo el succionamiento por medio de una ventosa [40].



Figura 21: Mini Bomba de vacío

Tabla 5: Especificaciones Técnicas de Bomba de vacío

Especificaciones Técnicas	
Modelo	YYP370-12E2
Voltaje Nominal	12 V DC
Sin corriente de carga	$\leq 250\text{mA}$
Flujo de aire	$\geq 2,5 \text{ l/min}$
Vacío Máximo	$\leq -60\text{Kpa}$
Nivel de Ruido	$\leq 55\text{dB}$
Prueba de vida estándar	$\geq 500\text{H}$

3.1.1.8. Caja de control eléctrico

Es una moldura de plástico que posee componentes eléctricos para controlar y supervisar procesos (Ver Figura 22). Los operadores pueden interactuar con los controles del panel que les facilitara muchas tareas, brindándoles seguridad al momento de utilizar los mandos de

accionamiento. En el proyecto se utilizará para instalar las protecciones, pulsadores de accionamiento y el sistema de fuerza. Los tableros eléctricos poseen diferentes funciones como control y de protección, siendo un elemento de vital importancia para la seguridad del operador y la protección de los equipos instalados [41].



Figura 22: Caja de Control Eléctrico

3.1.1.9. Sensor de proximidad fotoeléctrico infrarrojo E18-D80NK

Este sensor permite realizar detecciones sin tener contacto con el objeto en un rango de 80 cm como máximo (Ver Figura 23), se puede integrar de forma fácil con tarjetas de control. Su principio básico está basado en la detección del cambio de intensidad de luz, no necesitan de un espejo adicional por ser un sensor difuso su emisor como el receptor se encuentran integrados (Ver Tabla 6). Para el proyecto se emplea este sensor que permite detectar el metal que está compuesto la lata de atún que se desplazará por la banda transportadora [42].



Figura 23: Sensor de proximidad E18-D80NK

Tabla 6: Ficha Técnica Sensor Proximidad E18-D80NK

FICHA TECNICA	
MODELO E18-D80NK	
Voltaje de operación	5V DC
Corriente de trabajo	20mA máx.
Corriente de salida(carga)	100mA máx.
Rango de detección	3cm a 80cm
Emisor de luz	Led infrarrojo
Salida	Tipo NPN normalmente abierto (NO) (ON: GND/OFF: VCC)
Temperatura de trabajo	-25 a 70° C

Criterios selección de sensores.

De acuerdo a las definiciones de cada sensor y conociendo sus datos técnicos, características, área de aplicación se establece una comparativa entre sus ventajas y desventajas para poder elegir el sensor para su implementación en la detección de las latas de atún. Se empleará el sensor fotoeléctrico E18-D80NK que permite una detección a mayor distancia y que se puede ajustar por medio del potenciómetro (Ver Tabla 7).

Tabla 7: Característica de sensor inductivo y sensor de proximidad

Dispositivo	Aplicaciones	Ventajas	Desventajas
Sensor Inductivo LJ18A-8-/BX	Detección de metales como acero, bronce, aluminio	<ul style="list-style-type: none">• Son sensores sin contacto.• Trabajan bajo condiciones ambientales duras.• Poseen resultados y rendimientos muy predecibles.• Resistentes a cortocircuitos.• Ignoran materiales cercanos que no sean metales.	<p>Pueden detectar solo objetos metálicos.</p> <p>Su rango de detección es corto desde 0.4 a 0.8 mm, es decir tiene limitaciones de distancia.</p> <p>Por los campos magnéticos pueden ocurrir errores con la exactitud de su medición.</p>

Sensor Fotoeléctrico E18-D80NK	<ul style="list-style-type: none"> • Detección de metal, cartón, vidrio, madera 	<ul style="list-style-type: none"> • Son sensores sin contacto. • El rango de detección esta entre los 3 cm a 80 cm. • Su ajuste de detección se la realiza por medio de un potenciómetro 	Propiedades del objeto puede incidir en la detección.
--------------------------------	--	--	---

3.1.1.10. Conductor Eléctrico

El conductor eléctrico está formado por un solo hilo o varios hilos que pueden estar entrelazados entre sí (Ver Figura 24), compuesto de material como el cobre y aluminio, poseen una cubierta protectora que sirve de aislamiento para evitar que la energía eléctrica que por ella circula entre en contacto con personas u objetos. Para el proyecto se empleará el de cobre por las ventajas de conductividad eléctrica, de acuerdo a su calibre se las puede utilizar para que soporte el tipo de corriente que se necesita, es un componente necesario para dotar de energía a cada dispositivo eléctrico [43].



Figura 24: Conductor Eléctrico

3.1.1.11. Rele

Los relés son interruptores eléctricos, utilizan su principio del electromagnetismo para la conversión de pequeños estímulos eléctricos del microcontrolador en corrientes más altas. Las pequeñas señales del Arduino activan el electroimán del relé, al abrir o cerrar el circuito

cuando se alimenta con una potencia de 5 V, puede trabajar con voltajes alternos o directos (Ver Figura 25). Los relés tienen dos modalidades, normalmente abiertos (NO) y normalmente cerrado (NC) [44].

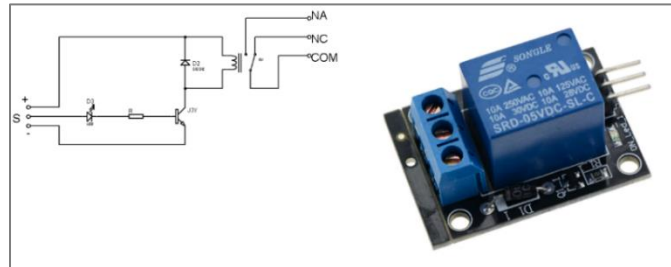


Figura 25: *Relé Arduino 110VAC a 5V DC*

3.1.1.12. Raspberry PI 4

El procesador del raspberry pi 4 (Ver Figura 26), es un mini ordenador posee un disipador de calor (Ver Tabla 8), que permite el control térmico por esto es el más indicado en trabajos de larga duración, en comparación con otros procesadores que poseen un núcleo de rango medio. Este modelo puede reproducir videos 4k a 60 fotogramas por segundo [45].



Figura 26: *Tarjeta Raspberry PI 4 4GB*

Tabla 8: Ficha Técnica Raspberry Pi 4 B

FICHA TECNICA MODELO RASPBERRY PI 4 B	
Procesador	Broadcom BCM2711, quad-core Cortex-A72(ARM V8)
Memoria	4GB LPDDR4
Conectividad	2.4 GHz y 5.0 GHz IEE 802.11b/g/n/ac Wireless LAN.
Puerto de pantalla	MIPI DSI de 2 carriles
Puerto de cámara	MIPI CSI de 2 carriles
Puertos audio estéreo y video	4 polos
Puertos	2 puertos USB 3.0
Voltaje de entrada	5V DC vía conector USB-C(mínimo 3 A), vía header GPIO 5V DC(mínimo 3 A)
Expansión	Encabezado GPIO de 40 pines estándar
Temperatura funcionamiento	0 – 50 ° C ambiente
Ranura tarjeta	Micro- SD cargar sistema operativo y almacenamiento de datos

3.1.1.13. Tarjeta Arduino Sensor Shield v5.0

El sensor shield permite realizar directamente la conexión de sensores a la tarjeta Arduino (Ver Figura 27), con conexión plug and play a diferentes módulos como servos y sensores. Brinda la facilidad de realizar conexiones de sensores analógicos.

Pines de conexión.

Pines digitales: están compuestos por 3 pines secuenciados de derecha a izquierda: el superior es Gnd= 0V, medio=Vcc(+5V) y el inferior= Señal(S)

Pines analógicos: compuestos por 3 pines estos se encuentran secuenciados de izquierda a derecha: el superior=Gnd(0V), medio=Vcc(+5V) y el inferior= Señal(S)

Conexión de sensores y dispositivos de salida.

Los pines de alimentación deben de conectarse en el sentido correcto: G conectado a G, Gnd a Gnd o 0V en el sensor, V conectado a Vcc o +5V en el sensor, S conectado al pin de señal OUT o INT.

Conexión de servos: los servos tienen 3 líneas, naranja, roja y café que van conectadas directamente al sensor shield (Ver Tabla 9), G se conecta al cable marrón, V conectamos al cable rojo y el cable naranja que es de la señal conectamos a S [46].

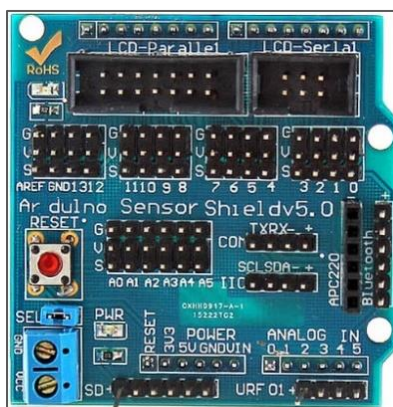


Figura 27: Sensor Shield V5.0

Tabla 9: Especificaciones Técnicas Sensor Shield V5.0

Especificaciones Técnicas Sensor Shield V5.0	
Voltaje de operación	5V DC
Compatibilidad	Arduino Uno, Mega, Leonardo
Puertos I/O	14
Puertos de entrada analógica	6
Puerto modulo inalámbrico	APC220
Puerto modulo Bluetooth	HC06 o HC05
Puerto modulo	SD card
Puerto	LCD paralelo
Puerto	LCD serial
Pulsador	Reset
Bornera	Alimentación externa(servos)
Led Power y Led L	Pin 13

3.1.1.14. Arduino UNO R3

Es una placa de microcontrolador que está basada en ATmega 328P (Ver Figura 28), permitiendo cargar nuevos códigos porque viene preprogramado con un cargador de arranque, sin que se tenga la necesidad de un programa de hardware externo. Se alimenta a través de una USB o con fuente de alimentación externa (Ver Tabla 10), esta puede ser una batería o un adaptador de CA a CC [47].

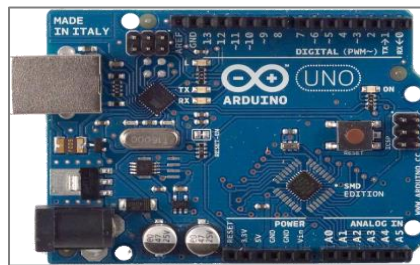


Figura 28: Placa Arduino Uno R3

Tabla 10: Especificaciones Técnicas Arduino Uno

Especificaciones Técnicas Arduino Uno	
Microcontrolador	ATMega328P
Velocidad de reloj	16MHZ
Voltaje de trabajo	5V
Voltaje de entrada	7,5 a 12 Voltios
Pinout	14 pines digitales(6PWM)
Puerto modulo	Puerto serie por hardware
Memoria	32 KB Flash (0,5 para bootloader)
Memoria	2KB RAM y 1KB Eprom

3.1.1.15. Fuente de Alimentación.

La fuente conmutada es un dispositivo electrónico que se encarga de transformar de corriente alterna a corriente continua (Ver Figura 29) que pueden tener una o varias salidas. Con esta

transformación de energía se puede alimentar diferentes dispositivos electrónicos como motor de CD (Ver Tabla 11), tarjetas, sensores [48]



Figura 29: Fuente de Alimentación AC/DC

Tabla 11: Especificaciones Técnicas de Fuente de Alimentación AC/DC

Especificaciones Técnicas Fuente de Alimentación 5V	
Tipo	Fuente conmutada de 5VCD a 5A
Voltaje de entrada	AC110-220V 50/60Hz
Potencia máxima de salida	50W
Protección	Cortocircuito/ sobrecarga/ sobretensión
Temperatura de trabajo	-10 ~ + 50 y el grado; c
Temperatura de almacenamiento	-20 ~ + 85 y el grado; c
Humedad ambiente	20% ~ 95% sin condensación
Material de Shell	Metal / base de aluminio

3.1.1.16. Interruptor termomagnético

El interruptor termomagnético (Ver Figura 30), es un dispositivo que combina elementos térmicos y magnéticos para la interrupción del flujo de corriente cuando detecta un valor de voltaje que excede ciertos parámetros. El funcionamiento se basa en que su sistema hay una dilatación del metal por el calor y la fuerza de atracción que se generan por los campos

magnéticos. Este interruptor posee un metal por el cual fluye la corriente, a medida que hay un aumento de la intensidad, el metal comienza a expandirse bajo la influencia del calor, creando un circuito abierto [49].



Figura 30: Interruptor Termomagnético

3.1.2. Componentes Lógicos.

3.1.2.1. Software Raspberry Pi Imager

Es la forma rápida y fácil de instalar un sistema operativo en una tarjeta microSD lista para usarla con la raspberry pi (Ver Figura 31), para el proyecto se empleará el sistema operativo de 64 bits [50].



Figura 31: Software Raspberry Pi Imager

3.1.2.2. Python 3.9.2

Es un lenguaje de programación orientado a objetos de alto nivel (Ver Figura 32), utilizado en la codificación, es multiplataforma es decir se puede ejecutar en las plataformas

principales como Microsoft Windows, Linux y macOS, es un software de código abierto, brindando la capacidad de desarrollar proyectos de forma rápida, usa menos líneas de código y posee una sintaxis simple y fácil al momento de programar [51].



Figura 32: Lenguaje de Programación Python

3.1.2.3. Open CV

OpenCv es una biblioteca de código abierto (Ver Figura 33), que posee infinidad de algoritmos que sirven para la visión por computadora. Se caracteriza por tener una estructura modular, es decir que sus paquetes contienen varias bibliotecas compartidas o estáticas. Posee módulos como:

Funcionalidad: define estructuras de datos básicas.

Procesamiento de imágenes: dentro del módulo incluye filtrado de imágenes lineales y no lineales, transformaciones de imágenes.

Análisis de video: incluye estimación de movimiento, sustracción de fondo y algoritmos de seguimiento.

Calibración de cámara y reconstrucción 3D: algoritmos básicos de geometría de vista múltiple, calibración de cámara única y estéreo.

Detección de objetos: detección de objetos e instancias de las clases predefinidas.

E/S de video: interfaz fácil de usar para la captura de video [52].



Figura 33: Librería OpenCV

3.1.2.4. Tensor Flow

Es una plataforma integral para el aprendizaje automático. Tensor Flow implementa matemáticas estándar en tensores (Ver Figura 34), así como operaciones especializadas para el aprendizaje. Cuando se configura de manera correcta se puede usar hardware acelerador como GPU para realizar operaciones de forma rápida. Posee compatibilidad para el cálculo numérico basado en matrices multidimensionales, GPU y procesamiento distribuido, diferenciación automática, construcción, capacitación y exportación de modelos. Tensor Flow proporciona herramientas para:

Optimización del rendimiento: acelerar el entrenamiento y la inferencia.

Exportar: se puede guardar el modelo cuando se termine de entrenar [53].



Figura 34: Librería Tensor Flow

3.1.2.5. Pandas

Pandas posee una herramienta que permite manipular y analizar los datos con características de ser rápida (Ver Figura 35), potente, flexible, es un paquete de Python con estructura de datos que fueron diseñadas para trabajar con etiquetados y que su uso sea fácil e intuitivo [54].



Figura 35: Librería Pandas

3.1.2.6. Ultralytics YOLOV5

Permite segmentar y detectar objetos en tiempo real, el YOLOv5 está basado en un aprendizaje profundo permitiendo un rendimiento a gran velocidad y precisión (Ver Figura 36). Está diseñado para el trabajo con diferentes aplicaciones con adaptaciones a diferentes dispositivos [55].



Figura 36: Librería Ultralytics

3.1.2.7. Node Red

Node-RED es la herramienta de desarrollo open-source que se basa en una programación visual, que puede conectarse a dispositivos de hardware y servicios de internet (Ver Figura

37). Es de fácil manejo, considerada como una aplicación de gestión y transformación de datos en tiempo real para dar soluciones IoT o Industria 4.0. Su entorno está compuesto de bloques de forma gráfica, permitiendo conectarlos por medio de nodos en la que se puede desarrollar tareas concretas [56]. Es un sistema que permite simplificar la conectividad y los servicios, para distribuirlos de forma eficiente, generando reducción del tiempo en el registro y gestión de la información.

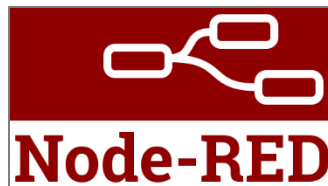


Figura 37: Node Red

3.1.2.8. Mosquitto

El MQTT es un protocolo de red empleado para mensajería (Ver Figura 38), con el objetivo de minimizar el ancho de banda, siendo utilizadas para conexiones con internet de las cosas. Usa el TCP/IP para su ejecución y funcionamiento, con topología PUSH/SUBSCRIBE, este tipo de sistema posee un cliente y un bróker.

Los clientes son los dispositivos que necesitan del bróker para tener una comunicación bidireccional. El bróker es el servidor para la comunicación con los clientes, donde reciben los datos para posterior enviarlos a los demás clientes que establecen la comunicación [57].



Figura 38: Protocolo de Red Mosquitto

3.1.2.9. InfluxDB

InfluxDB es un sistema de gestión de base de datos optimizada (Ver Figura 39), su mayor ventaja es la de mantener alta velocidad de escritura por un largo periodo de tiempo, cuando almacena y procesa datos de medición con marcas de tiempo. Se ejecuta como servidor en la nube que se puede personalizar, posee una interfaz de usuario basada en la web para ingresar o visualizar los datos [58].



Figura 39: Base de Datos InfluxDB

3.1.2.10. Grafana

La plataforma de grafana permite el monitoreo y visualización de datos por medio de tablas o gráficos en un panel de control permitiendo la interpretación o comprensión (Ver Figura 40), otra característica es que se pueden realizar consultas o configurar avisos sobre los indicadores para identificar las inconsistencias con mayor facilidad, brindando una mayor eficiencia en los procesos [59].



Figura 40: Grafana

3.2. DISEÑO DE LA PROPUESTA

3.2.1. Topología de comunicación

Para comunicar los diferentes dispositivos en la nube como la Raspberry Pi 4 y el computador de manera inalámbrica, se empleará el protocolo de comunicación TCP/IP para todo el sistema, los dispositivos se encuentran en comunicación con la red local 192.168.0.1/24, su mascara de subred 255.255.255.0 y la puerta de enlace 192.168.0.1, se asignan las direcciones IP a cada dispositivo. Se muestran las direcciones IP con el puerto TCP respectivo que permite establecer comunicación con el servicio IoT (Ver Tabla 12).

Tabla 12: Direcciones IP de Dispositivos

Direcciones IP de Dispositivos		
Dispositivo	Direcciones IP	
Router	192.168.1.2	
Tarjeta Raspberry Pi4	192.168.1.8	
Laptop	192.168.1.7	
Servicio IoT	Dirección IP	Puerto TCP
Mosquitto	192.168.1.8	1883
InfluxDB	192.168.1.8	8086
Grafana	192.168.1.8	3000
Node-RED	192.168.1.8	1880

La topología de comunicación utilizada en este proyecto es estrella (Ver Figura 41), se puede emplear conectividad inalámbrica o cableada entre la Raspberry Pi 4 y el Router. Para la red de comunicación empleamos Mosquitto que es el bróker MQTT que permite receptor mensajes para enviarlos a sus destinatarios, para esto utilizamos Node-RED que emplea bloques permitiendo realizar la programación y mediante la utilización de InfluxDB creamos

la base de datos, cada vez que ingresa un dato tendrá una fecha y hora y esta comunicación culmina con grafana que permite mostrar mediante graficas los datos en un dashboards.

3.2.2. Diagrama Red de Comunicación del Sistema

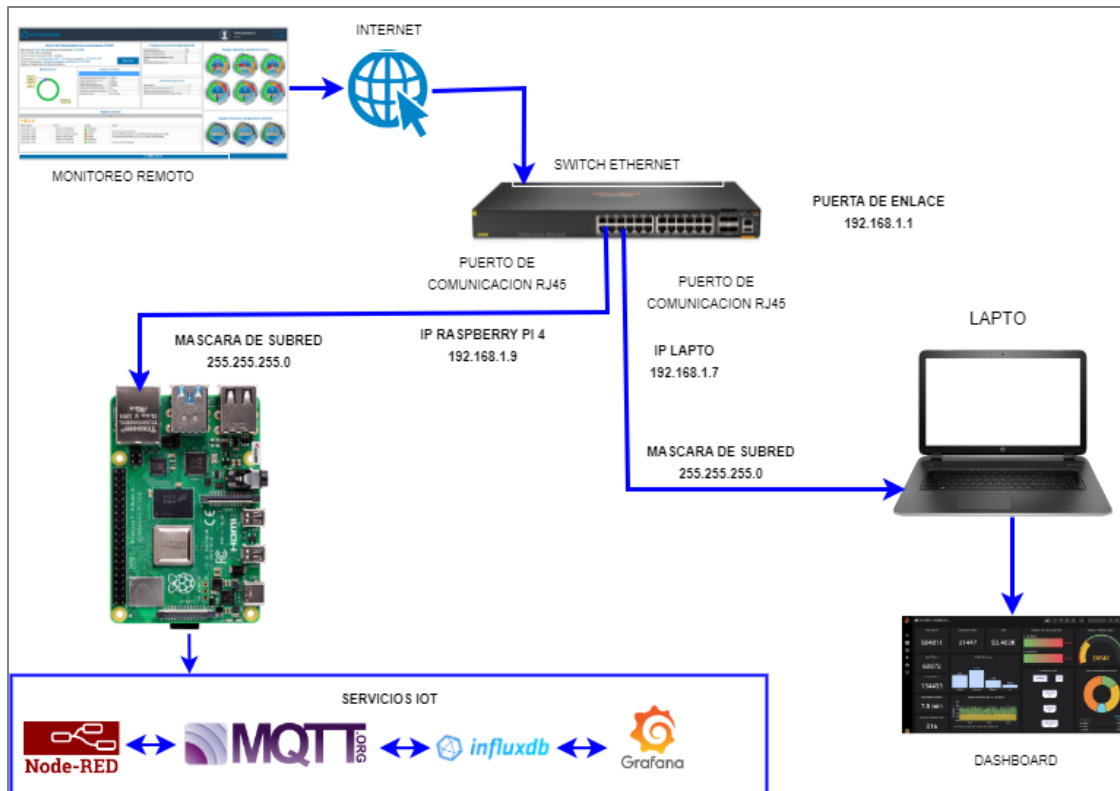


Figura 41: Diagrama de Comunicación del sistema

3.2.3. Diseño del Sistema de Control de Visión Artificial

En esta sección se detalla el proceso de ensamble de la banda transportadora y del brazo robótico, con sus respectivos sensores, tarjeta de control y cámara con sus respectivas conexiones para realizar el sistema de control de visión artificial (Ver Figura 42).



Figura 42: Sistema de control de Visión Artificial

3.2.4. Sistema de control para productos de enlatados con sensor fotoeléctrico

En los procesos de producción se necesita tener una contabilización de cada producto que se desplaza por la banda transportadora y que por medio de los sensores fotoeléctricos permitirá la detección de las latas hasta 80 cm de distancia, esto nos permite controlar cuantas se encuentran con los estándares de calidad o verificar los problemas de sellado y de falta de texto informativo de la producción de lata , antes de ser distribuidos.

El sistema se activa por medio de programación en Arduino donde se accionará la banda transportadora. Cada vez que el sensor detecte una lata emitirá una señal hacia la cámara para que el sistema inicie con la visión artificial verificando el estado de calidad de la lata y este a su vez emite una señal al sistema de control del brazo robótico para separar el producto de acuerdo a las fallas que presente cada lata.

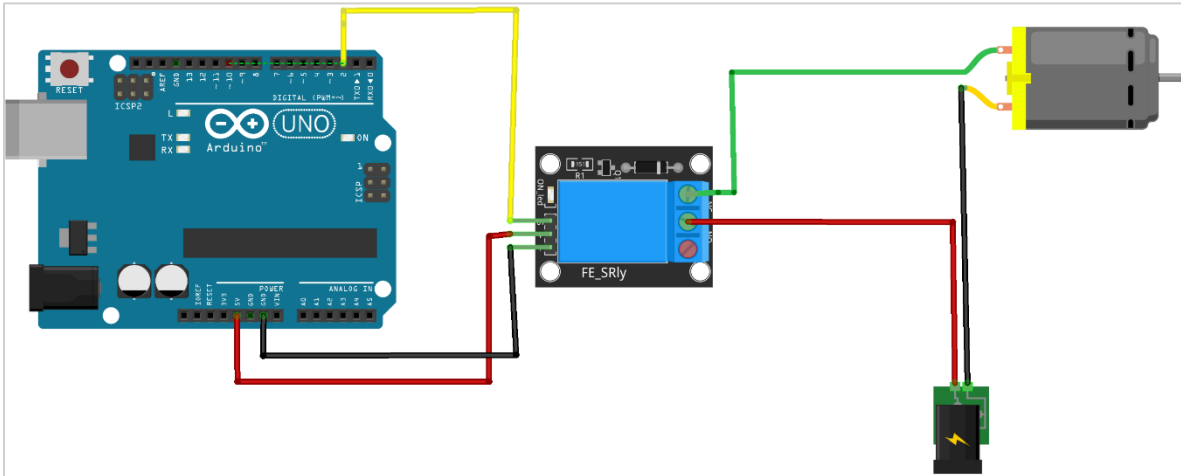


Figura 43: Conexión del motor al Relay

3.3. Diseño Eléctrico y Electrónico del prototipo

Se realiza las conexiones eléctricas para verificar el funcionamiento de la banda transportadora y de los servomotores del brazo robótico. Se realizarán las pruebas de funcionamiento de los servomotores mediante líneas de código en Arduino, para determinar el posicionamiento de cada servo que será utilizado en el proceso de clasificación.

3.3.1. Conexión de los servomotores a la tarjeta Sensor Shield V5.0

Procedemos a realizar las conexiones de los servomotores a la placa sensor shield V5.0, a continuación, se detallan los pines de conexión de los servos del brazo robótico: el pin digital 3 se conecta al servo de la base, el pin 5 conectamos al servo del antebrazo, el pin 6 conectado al antebrazo esta articulación posee 2 servos que giran en sentido contrario, el pin 9 se conecta al servo del brazo, el pin 10 para la muñeca que ejerce una rotación vertical y el pin 11 para el servo de la mano, la conexión de las terminales para cada servo seria para ‘G’ tierra, ‘V’

es para el voltaje y ‘S’ para la señal respectivamente. Para la alimentación de la tarjeta sensor shield utilizamos una fuente de alimentación de 110 AC de entrada y de salida los 5 VDC con corriente de 5 A que es requerida para el buen funcionamiento del brazo robótico (Ver Figura 44).

Para iniciar el control de los servomotores utilizamos la librería “Servo” que facilita la comprensión de las acciones, seguido establecemos las siguientes líneas de código para que el brazo tenga una posición inicial, donde se le determinara un tiempo para que ejecute la siguiente acción del otro servo. Los servos tienen una apertura de 0° a 180°, donde hay que establecer donde se inicia el servo y el ángulo hasta donde se desplazara en un tiempo determinado (Ver Figura 46).

3.3.1.1. Esquema de conexión de los servos a la placa Arduino

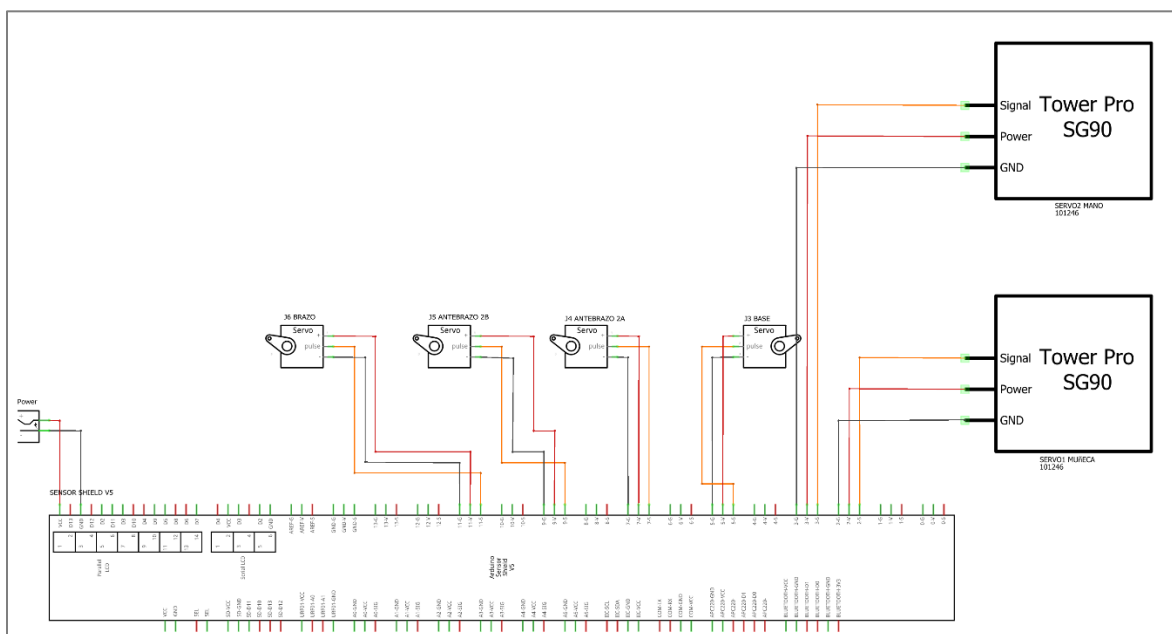


Figura 44: Esquema conexión de servomotor brazo robótico

3.3.2. Conexión de los sensores fotoeléctricos a la placa Arduino Uno

El sensor E18-D80NK es un sensor a distancia, NPN, en su parte interna posee un transistor y este tiene la función de cuando realiza una detección se cierra, dando un valor de 0V, pero cuando este no detecta en la salida tenemos un voltaje de 5V. en la figura se observa cómo queda establecido la conexión de los 3 cables que posee hacia la placa de control. Se emplea para este proyecto con la finalidad de la detección de las latas y su respectivo conteo por la banda transportadora (Ver Figura 45).

Cable Café: Positivo (Fase)

Cable Azul: Negativo (Tierra)

Cable Negro: Señal (NPN salida de colector abierto)

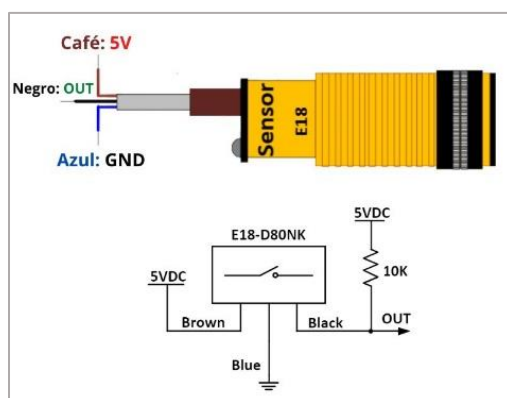


Figura 45: Configuración de pines del sensor fotoeléctrico E18-D80NK

La conexión del sensor fotoeléctrico E18-D80NK, se puede conectar a la tarjeta Arduino Uno, su conexión eléctrica quedaría conectando como se indica (Ver Tabla 13).

Tabla 13: Pines de conexión Sensor Fotoeléctrico

Pines de Conexión	
Pin Modulo	Pin Arduino Uno
Cable marrón	Voltaje 5 V DC
Cable azul	Tierra GND
Cable negro	Salida D2

3.3.2.1. Esquema de conexión sensor fotoeléctrico a la placa Arduino Uno.

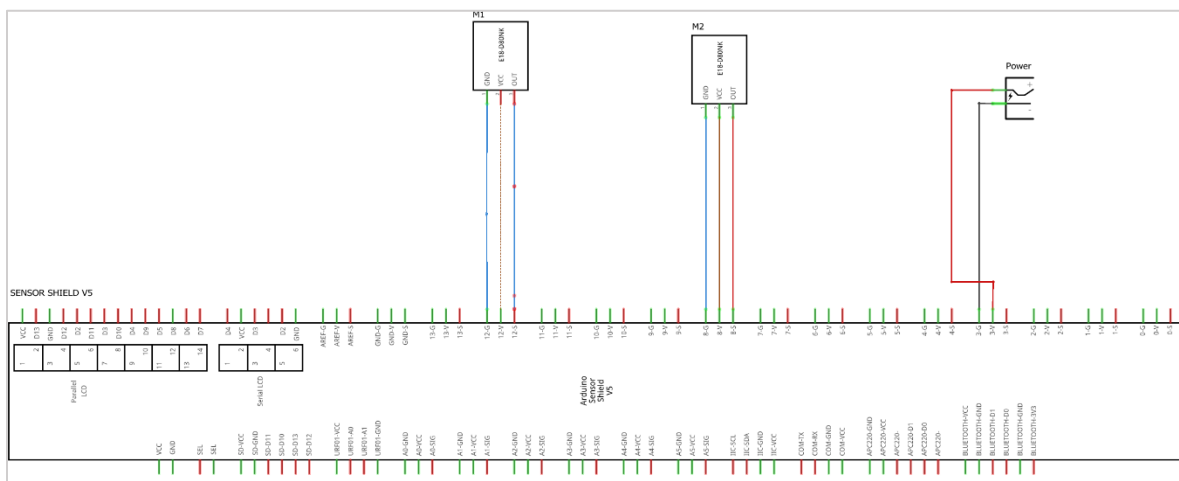


Figura 46 : Conexión de sensor fotoeléctrico

3.3.3. Conexión control automático de banda transportadora

Para el funcionamiento del motorreductor de la banda, este se encuentra conectado a un regulador de velocidad , para su alimentación utilizamos una fuente de 110V, conectamos los cables del switch a la entrada del Relay, se realiza la conexión al pin NC y al COM , en la salida del Relay conectamos el VCC al pin de 5v, y el IN va conectado al pin de señal del

arduino y el GND se conecta a uno de los pines GND del microcontrolador Arduino. (Ver Figura 47).

El motorreductor se activa durante un tiempo establecido por el microcontrolador este tiempo es utilizado para realizar la inspección con el sistema de visión, luego el motor gira para que la banda se mueva hasta que la posición de la lata llega al sistema de clasificación donde la banda se detiene por otro determinado tiempo para que el brazo robótico seleccione la lata de acuerdo los resultados de la visión artificial.

3.3.4. Diagrama de conexión motorreductor a la placa Arduino

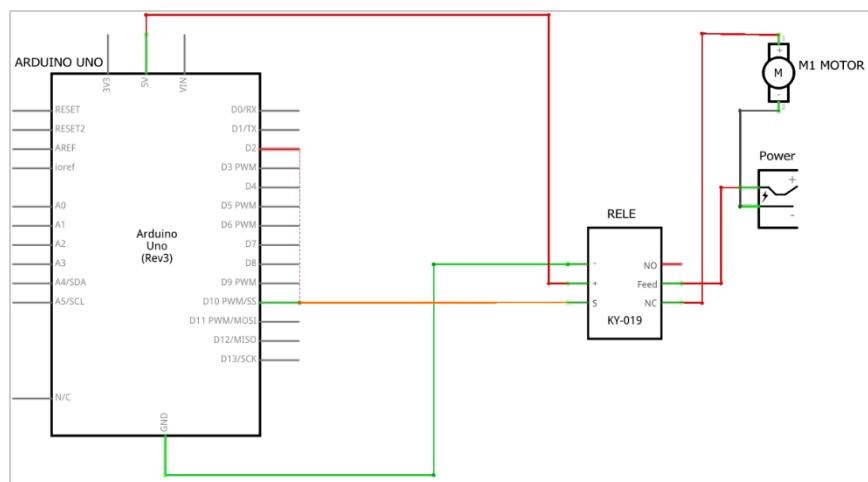


Figura 47: Diagrama de conexión del motorreductor a Arduino

3.4. Instalación del sistema operativo Pi OS a la Raspberry Pi 4 B

Para usar la tarjeta Raspberry Pi 4 B previamente se instala en la laptop la herramienta Raspberry Pi Imager V1.7.3 que permite realizar la instalación del sistema operativo Pi OS. Insertamos nuestra tarjeta SD de 32 GB a un adaptador SD y luego a la computadora y cuando

ejecutamos Raspberry Pi Imager, damos un clic en CHOOSE OS que despliega opciones de sistemas operativos existentes, seleccionamos Raspberry Pi OS 64-bit (Ver Figura 48), que permite instalar todas las aplicaciones para utilizarlas en el sistema de visión artificial.

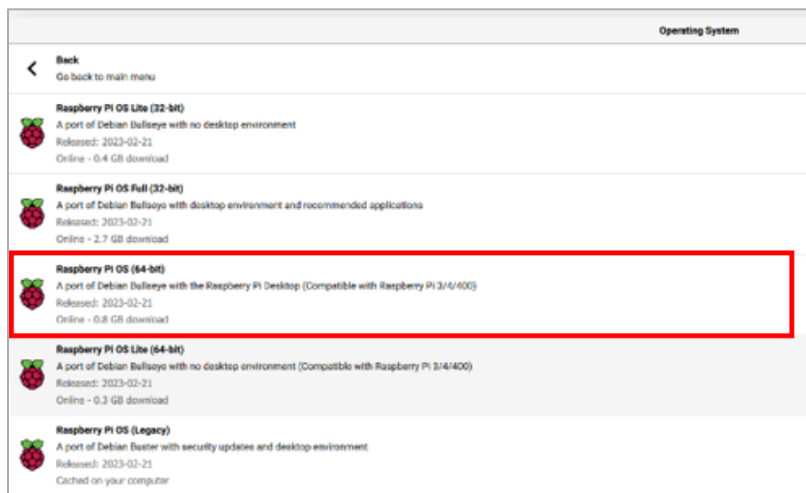


Figura 48: Sistema Operativo PI OS 64 Bits

Seleccionamos la tarjeta SD de 32 GB, al dar clic se procede a escribir el sistema operativo, donde visualizamos el Slider que contiene un mensaje “Writing” donde debemos de esperar hasta que cargue el 100%, este nos indica que la escritura está cargada en el sistema operativo, luego el mismo sistema verifica que todos los archivos se escriban de forma correcta, donde se visualiza por el Slider “Verifying” que debe estar al 100 % (Ver Figura 49), al concluir visualizamos un mensaje que podemos retirar la tarjeta SD que contiene el sistema operativo Pi OS.

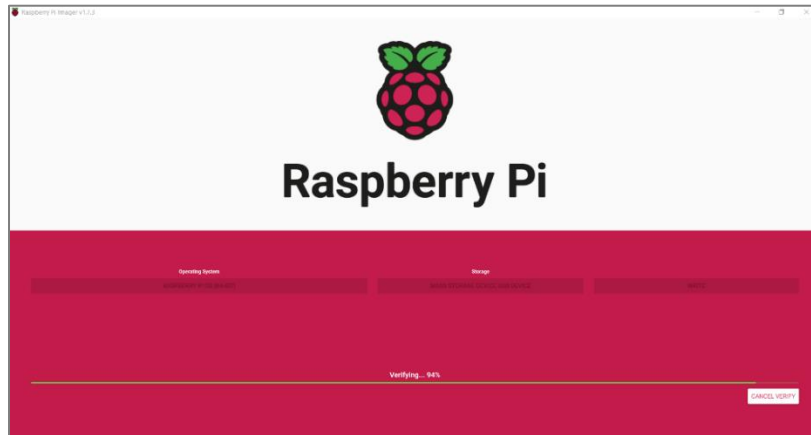


Figura 49: Verificación de Instalación del sistema operativo

3.5. Instalación de Advances IP Scanner

Instalamos Advances IP Scanner en nuestro ordenador para que proporcione la IP que fue asignada a la raspberry (Ver Figura 50), le damos un clic en explorar para el respectivo escaneó de las IP que se encuentran en nuestra red, y podemos visualizar que la IP asignada a la Raspberry Pi 4 es la 192.168.1.35

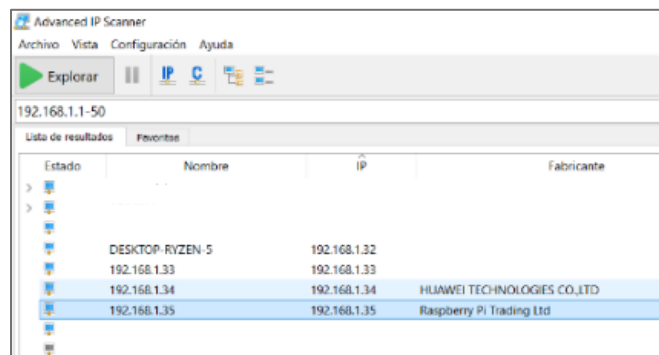


Figura 50: Instalación de Advances IP Scanner

En el cmd de la computadora digitamos el siguiente comando ssh WG@192.168.1.8, para acceder al host de autenticidad y le damos enter, solicitara si deseamos continuar donde tenemos que darle clic en yes para continuar, luego tenemos que colocar nuestra clave para ingresar a la raspberry Pi 4 y establecer la conexión (Ver Figura 51).

```
WG@raspberrypi -
Microsoft Windows [Versión 10.0.19044.2486]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\VERONICA MONTENEGRO>ssh WG@192.168.1.35
The authenticity of host '192.168.1.35 (192.168.1.35)' can't be established.
ECDSA key fingerprint is SHA256:10poXSF8D8yU0qxp3ClAoQv7g@4dzdZl#H1RCKomuzS0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.35' (ECDSA) to the list of known hosts.
WG@192.168.1.35's password:
Linux raspberrypi 5.15.84-v8+ #1613 SMP PREEMPT Thu Jan 5 12:03:08 GMT 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 27 15:18:21 2023
WG@raspberrypi:~$
```

Figura 51: Comandos para ingreso a la Raspberry

3.6. Habilitar VNC Viewer mediante comando en el CMD

Ahora procedemos habilitar el VNC Viewer esto permite administrar de forma gráfica nuestra raspberry y digitamos en el Cmd el siguiente comando sudo raspi-config y le damos enter para entrar en la herramienta de configuración del software (Ver Figura 52). Seleccionamos Interface Options para realizar la configuración de habilitación del VNC y nos pregunta si queremos habilitar y damos clic en yes el cual nos indicara que se encuentra habilitado y le damos ok para finalizar.

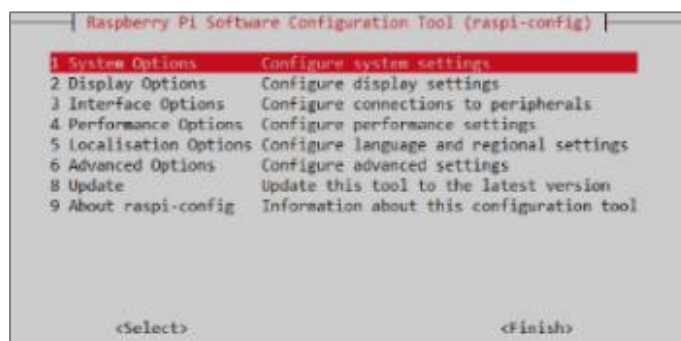


Figura 52: Habilitar el VNC Viewer por CMD

3.7. Instalación e Ingreso al VNC Viewer en la PC.

Instalamos el VNC en nuestra computadora, luego al abrir el programa solicita que ingrese la IP que se le asignó a la Raspberry Pi 4, para esto ingresamos 192.168.1.35, también solicita el nombre de usuario y escribimos WG (Ver Figura 53). Para ingresar al entorno de la Raspberry Pi 4, el VNC Viewer solicita la autenticación con el nombre de usuario y contraseña que debemos de escribir y le damos en aceptar para estar dentro del entorno de la raspberry.

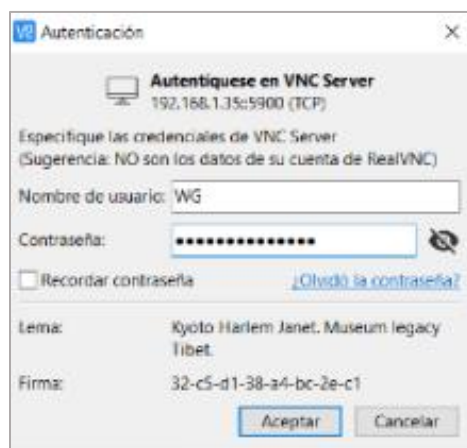


Figura 53: Autenticación de ingreso a VNC Viewer

3.8. Configuración de la memoria GPU en la raspberry

La memoria GPU de la raspberry Pi4 viene por defecto con 76 Mbytes y para nuestro proyecto de visión artificial tenemos que dejarlo en 256Mbyte aumentando de esta manera la memoria de la GPU. La tarjeta Raspberry Pi 4 se inicia desde dos EEPROM, se emplea dos comandos para mejorar la disipación del calor de la CPU que desde su inicio alcanza temperaturas de 65°C, pero con la actualización baja su temperatura a 48°C y para conseguir

esto debemos de ejecutar los siguientes comandos, para un rendimiento térmico y la optimización de la Raspberry Pi 4 (Ver Tabla 14).

Tabla 14 : Comandos Actualización del Firmware

Actualización de Firmware	
Comando	Detalle
<code>sudo rpi-eeeprom-update</code>	se ejecuta para obtener el estado actual
<code>sudo rpi-eeeprom-update -a</code>	se ejecuta para actualizar el firmware

3.9. Instalar Opencv-Bullseye en Raspberry Pi 4

Antes de instalar Opencv debemos de actualizar el firmware para seguir con la instalación del programa. Para conocer la arquitectura escribimos el siguiente comando en la consola: **uname -m** y para conocer la versión escribimos el comando: **cat /etc/os release**

Para instalar el opencv clonamos el repositorio con el siguiente comando: **sudo git clone https://github.com/freedomwebtech/rpi-bullseye-opencv4.5.5.git**, para seguir con la instalación digitamos en el terminal la siguiente línea de comandos: **cd rpi-bullseye-opencv4.5.5/** y le damos enter para ubicarnos en el comando interno, luego escribimos: **ls**, donde verificamos que tenemos `install.sh` que instalara el opencv con la versión de 64 bits.

El comando: **sudo chmod 775 install.sh**, nos proporciona el permiso para el script, digitamos **ls** para confirmar que `install.sh` se encuentre en verde y esto indica que tenemos otorgado los permisos. Escribimos el comando: **sudo ./install.sh**, en la consola y presionamos enter, donde debemos de esperar un determinado tiempo para instalar el opencv donde se

compilarán e instalarán automáticamente todos los paquetes de la última versión de opencv-Bullseye con una versión de 64 bits (Ver Figura 54).

```
WG@raspberrypi:~$ sudo git clone https://github.com/traadnwebtech/rpi-bullseye-opencv4.5.5.git
Cloning into 'rpi-bullseye-opencv4.5.5'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (0/0), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (12/12), 4.07 KiB | 2.93 MiB/s, done.
Resolving deltas: 100% (1/1), done.
WG@raspberrypi:~$ ls
2023-04-27-161648_1024x768_scret.png  Downloads  rpi-bullseye-opencv4.5.5
Bookshelf                             Music       Templates
Desktop                               Pictures    Videos
Documents                             Public
WG@raspberrypi:~$ cd rpi-bullseye-opencv4.5.5/
WG@raspberrypi:~/rpi-bullseye-opencv4.5.5$ ls
install.sh
WG@raspberrypi:~/rpi-bullseye-opencv4.5.5$ sudo chroot / install.sh
WG@raspberrypi:~/rpi-bullseye-opencv4.5.5$ ls
install.sh
WG@raspberrypi:~/rpi-bullseye-opencv4.5.5$ sudo ./install.sh
```

Figura 54: Instalación de Opencv en la Raspberry Pi 4

Verificamos la versión de opencv ingresando **python3** en la consola y como resultado tenemos la versión de Python que es 3.9.2, en la siguiente línea de la consola escribimos: **>>import cv2**, en la línea siguiente digitamos: **>>cv2.__version__** dando como resultado que nuestra versión de opencv es “4.6.0”(Ver Figura 55).

```
WG@raspberrypi:~$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.6.0'
>>>
```

Figura 55: Comandos y versión de Opencv

3.10. Instalación de Tensor Flow Lite

Para instalar las librerías necesarias debemos de digitar comandos en la terminal de la raspberry, primero clonamos el repositorio con el siguiente comando: **sudo git clone**

<https://github.com/freedomwebtech/tensorflow-lite-bullseye.git>, que permite descargar y poder extraer los archivos que contienen este repositorio para almacenarlo en el directorio HOME WG de la tarjeta raspberry Pi 4 (Ver Figura 56). Con el comando **ls** podemos verificar que la carpeta se encuentra dentro del directorio HOME WG. Para ingresar a la carpeta `tensorflow-lite-bullseye` digitamos el siguiente comando: **cd tensorflow-lite-bullseye/**

```

WG@raspberrypi:~$ sudo git clone https://github.com/freedomwebtech/tensorflow-
lite-bullseye.git
Cloning into 'tensorflow-lite-bullseye'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
WG@raspberrypi:~$ ls
2023-04-27-161848_1024x768_screenshot.png  Downloads  rpi-bullseye-opencv4.5.5
Bookshelf                                  Music       Templates
Desktop                                    Pictures   tensorflow-lite-bullseye
Documents                                  Public     Videos
WG@raspberrypi:~$ cd tensorflow-lite-bullseye/
WG@raspberrypi:~/tensorflow-lite-bullseye$

```

Figura 56: Comandos Instalación de Librerías Tensor Flow Lite

Ingresamos el comando **ls** para verificar que estamos dentro del archivo `tensorflow-lite.sh`. Para establecer el permiso completo de ejecución digitamos el siguiente comando en la consola: **sudo chmod 775 tensorflow-lite.sh** donde los archivos de descarga serán almacenados en la carpeta `tensorflow-lite-bullseye`. Escribimos el siguiente comando en la consola: **sudo ./tensorflow-lite.sh** que descargara todos los paquetes y los instalara (Ver Tabla 15).

Tabla 15: Comandos Instalación de Tensorflow Lite

Comandos Instalacion de Tensorflow Lite	
Comandos	Detalle
<code>sudo git clone https://github.com/freedomwebtech/tensorflow-lite-bullseye.git</code>	Permite la extraccion de archivos
<code>Ls</code>	Verifica la carpeta del directorio

cd tensorflow-lite-bullseye/	Ingresar a la carpeta tensorflow-lite-bullseye
sudo chmod 775 tensorflow-lite.sh	Establecer el permiso completo de ejecución
sudo ./tensorflow-lite.sh	Descargar todos los paquetes e instalar

Cuando digitamos **ls** en la consola visualizamos la carpeta `examples tensorflow-lite.sh`. Si digitamos el comando **pwd** nos da como resultado en qué dirección se encuentra la carpeta de tensorflow que sería `/home/WG/tensorflow-lite-bullseye` (Ver Figura 57).

```
WG@raspberrypi:~/tensorflow-lite-bullseye $ pwd
/home/WG/tensorflow-lite-bullseye
WG@raspberrypi:~/tensorflow-lite-bullseye $ ls
examples tensorflow-lite.sh
WG@raspberrypi:~/tensorflow-lite-bullseye $ cd examples/
WG@raspberrypi:~/tensorflow-lite-bullseye/examples $
```

Figura 57: Ubicación de carpeta tensorflow lite

Con el comando **ls** verificamos que estamos dentro de la carpeta `lite`, luego ingresamos el comando **cd lite/** para luego digitar **ls** dando como resultado la carpeta `examples`. Con el comando **cd examples/** y para visualizar la carpeta de `image-segmentation` lo hacemos con el comando **ls** (Ver Figura 58).

```
WG@raspberrypi:~/tensorflow-lite-bullseye/examples $ ls
AUTHORS          courses         lite            templates
CODEOWNERS       __init__.py    README.md      tensorflow_examples
CONTRIBUTING.md LICENSE         setup.py
WG@raspberrypi:~/tensorflow-lite-bullseye/examples $ cd lite/
WG@raspberrypi:~/tensorflow-lite-bullseye/examples/lite $ ls
code4me  examples  README.md  tools
WG@raspberrypi:~/tensorflow-lite-bullseye/examples/lite $ cd examples/
WG@raspberrypi:~/tensorflow-lite-bullseye/examples/lite/examples $ ls
audio_classification  recommendation
bert_qa               reinforcement_learning
classification_by_retrieval  smart_reply
digit_classifier      sound_classification
gesture_classification  speech_commands
image_classification  speech_recognition
image_segmentation    style_transfer
model_personalization  super_resolution
object_detection       text_classification
optical_character_recognition  text_searcher
pose_estimation        video_classification
posenet
```

Figura 58 : Ubicación de carpeta image-segmentation

Para visualizar la versión instalada de tensorflow-lite-bullseye ingresamos los siguientes comandos: **python** y nos da la versión de 3.9.2, luego procedemos a ingresar: **>>>import tflite_runtime**, seguido digitamos: **>>>tflite_runtime.__versión__** y tenemos la versión de tensorflow-lite-bullseye que es 2.5.0 (Ver Figura 59).

```
W0@raspberrypi:~$ python
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license()" for more
>>> import tflite_runtime
>>> tflite_runtime.__version__
'2.5.0.post1'
>>>
```

Figura 59: Versión de tensorflow lite

3.11. Instalación de numpy

Para la instalación de numpy en la raspberry pi 4 ingresamos el siguiente comando para que descargue todos los paquetes: **pip install numpy --upgrade --ignore-installed** (Ver Figura 60).

```
W0@raspberrypi:~$ pip3 install numpy --upgrade --ignore-installed
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting numpy
  Downloading numpy-1.24.3-cp39-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.0 MB)
----- 14.0/14.0 MB 4.4 MB/s eta 0:00:00
Installing collected packages: numpy
  WARNING: The scripts f2py, f2py3 and f2py3.9 are installed in '/home/W0/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.24.3
W0@raspberrypi:~$
```

Figura 60 : Instalación de librerías Numpy

3.12. Instalación de Tesseract OCR

Abrimos una terminal donde escribimos el comando: **sudo apt install tesseract-ocr** y le damos enter, donde solicita que le demos en S para continuar y descargar todos los archivos (Ver Figura 61). Para visualizar la versión del Tesseract instalado ingresamos el comando: **tesseract -v** y da como resultado la versión de tesseract 4.0.0.

Para tener la librería en español que permite reconocer caracteres como la ñ debemos de ingresar el siguiente comando en la terminal **sudo apt install tesseract-ocr-spa** que descarga las librerías de la configuración de tesseract en español.

```
w@raspberrypi:~$ sudo apt install tesseract-ocr
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  liblpt5 libtesseract4 tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  liblpt5 libtesseract4 tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 6,990 kB of archives.
After this operation, 22.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main arm64 liblpt5 arm64 1.79.0-1.1
+deb11u3 [958 kB]
Get:2 http://deb.debian.org/debian bullseye/main arm64 libtesseract4 arm64 4.1.1
-2.1 [1,142 kB]
Get:3 http://deb.debian.org/debian bullseye/main arm64 tesseract-ocr-eng all 1:4
.09-g1t30-7274cfa-1.1 [1,592 kB]
Get:4 http://deb.debian.org/debian bullseye/main arm64 tesseract-ocr-osd all 1:4
.09-g1t30-7274cfa-1.1 [2,891 kB]
Get:5 http://deb.debian.org/debian bullseye/main arm64 tesseract-ocr arm64 4.1.1
```

Figura 61: Instalación de librerías para Tesseract OCR

Instalación de librería necesaria para trabajar con tesseract ingresando el siguiente comando: **pip install pillow pytesseract imutils** y le damos enter para que se instalen todos los paquetes (Ver Figura 62) para poder trabajar con códigos en Python.

```
WG@raspberrypi:~$ pip3 install pillow pytesseract imutils
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages (8.1.2)
Collecting pytesseract
  Downloading https://www.piwheels.org/simple/pytesseract/pytesseract-0.3.10-py3-none-any.whl (14 kB)
Collecting imutils
  Downloading https://www.piwheels.org/simple/imutils/imutils-0.5.4-py3-none-any.whl (26 kB)
Collecting packaging>=21.3 (from pytesseract)
  Downloading https://www.piwheels.org/simple/packaging/packaging-23.1-py3-none-any.whl (48 kB)
 48.9/48.9 kB 192.3 kB/s eta 0:00:00
Installing collected packages: imutils, packaging, pytesseract
  WARNING: The script pytesseract is installed in '/home/WG/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed imutils-0.5.4 packaging-23.1 pytesseract-0.3.10
WG@raspberrypi:~$
```

Figura 62 : Paquetes para Tesseract OCR

3.13. Habilitar 1-Wire e I2C en la Raspberry pi 4

Realizamos la configuración en la consola de la raspberry a través del siguiente comando **sudo raspi-config**, se mostrará una ventana donde seleccionamos **Interface Options** para habilitar 2 protocolos series, seleccionamos **I2C** y damos clic en **yes** para que quede habilitado, el mismo procedimiento realizamos con el **1-Wire** para habilitar los dos tipos de interfaces series (Ver Figura 63).

```
Raspberry Pi Software Configuration Tool (raspi-config)
I1 Legacy Camera Enable/disable legacy camera support
I2 SSH Enable/disable remote command line access using SSH
I3 VNC Enable/disable graphical remote access using RealVNC
I4 SPI Enable/disable automatic loading of SPI kernel module
I5 I2C Enable/disable automatic loading of I2C kernel module
I6 Serial Port Enable/disable shell messages on the serial connection
I7 1-Wire Enable/disable one-wire interface
I8 Remote GPIO Enable/disable remote access to GPIO pins
```

Figura 63: Habilitar 1-Wire e I2C en la Raspberry pi 4

3.14. Instalación de Node-Red en Raspberry Pi 4B

Ingresamos a la consola de la Raspberry Pi 4, escribimos el comando y esperamos a que se cargue todas las librerías necesarias (Ver Figura 64).

```
Bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

```
Running Node-RED install for user GW at /home/GW on debian

This can take 20-30 minutes on the slower Pi versions - please wait.

Stop Node-RED ✓
Remove old version of Node-RED ✓
Remove old version of Node.js ✓
Install Node.js 16 LTS ✓ v16.20.1 Npm 8.19.4
Clean npm cache ✓
Install Node-RED core ✓ 3.0.2
Move global nodes to local ✓
Npm rebuild existing nodes ✓
Install extra Pi nodes ✓
Add shortcut commands ✓
Update systemd script ✓

Any errors will be logged to /var/log/nodered-install.log
All done.
You can now start Node-RED with the command node-red-start
or using the icon under Menu / Programming / Node-RED
Then point your browser to localhost:1880 or http://(your_pi_ip_address):1880
```

Figura 64: Comandos de Instalación Node-Red

Para ejecutar como servicio digitamos **node-red-start** en el terminal, iniciamos el servicio de NodeRed y muestra la salida del registro, se realiza de esta manera para seguir trabajando en la raspberry con las demás aplicaciones (Ver Figura 65).

```
GW@raspberrypi:~$ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.1.39:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
16 Jul 10:06:54 - [info]
Welcome to Node-RED
-----
16 Jul 10:06:54 - [info] Node-RED version: v3.0.2
16 Jul 10:06:54 - [info] Node.js version: v16.20.1
16 Jul 10:06:54 - [info] Linux 6.1.34-v8+ arm64 LE
16 Jul 10:06:55 - [info] Loading palette nodes
16 Jul 10:06:57 - [info] Settings file : /home/GW/.node-red/settings.js
```

Figura 65: Node-Red como Servicio

Para verificar la conexión, ingresamos en nuestro navegador la dirección IP de la raspberry y al final le agregamos 1880 para conectarnos en línea con NodeRed (Ver Figura 66).

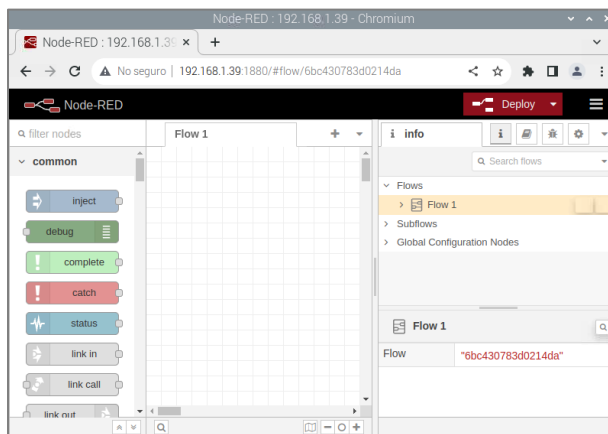


Figura 66: Conexión en línea con Node-Red

3.14.1. Configuración de Seguridad para NodeRed

Emplearemos http para realizar los ajustes de configuración de seguridad.

Generamos el hash del password con el siguiente comando porque la aplicación de NodeRed instalado es superior a 1.1.0.

```
node-red admin hash-pw
```

Ingresamos la contraseña y esta genera el hash para realizar la configuración del acceso a NodeRed. Accedemos al archivo settings.js que se encuentra en el directorio de usuario, digitamos los siguientes comandos (Ver Figura 67).

```
ls -a  
cd .node-red/  
ls  
nano settings.js
```

```
GW@raspberrypi:~$ node-red admin hash-pw
Password:
$2a$08$Rg9T4JWPZ9Rq9p0Qx7
GW@raspberrypi:~$ ls -la
.                Desktop      .pki
..               Documents   .profile
2023-09-08-001556_800x600_scr0t.png  Downloads  Public
2023-09-08-001559_800x600_scr0t.png  .java      PycharmProjects
2023-09-08-002121_800x600_scr0t.png  .local     Templates
.bash_history    Music       Videos
.bash_logout    .node-red  .vnc
.bashrc         .npm       .wget-hsts
Bookshelf       .npmrc     .Xauthority
.cache          pi-apps    .xsession-errors
.config        Pictures   .xsession-errors.old
GW@raspberrypi:~$ cd .node-red/
GW@raspberrypi:~/.node-red$ ls
lib  node_modules  package.json  settings.js
GW@raspberrypi:~/.node-red$ nano settings.js
```

Figura 67: Configuración de acceso a Node-Red

Al ejecutar **nano settings.js** descomentamos las siguientes líneas de seguridad y en password cambiamos por el hash generado para habilitar la contraseña cuando iniciemos NodeRed. Luego procedemos a guardar el archivo con **Ctrl+O** y salimos con **Ctrl+X** (Ver Figura 68).

```
GNU nano 5.4 settings.js *
* - https
* - httpsRefreshInterval
* - requireHttps
* - httpNodeAuth
* - httpStaticAuth
...../
/** To password protect the Node-RED editor and admin API, the following
 * property can be used. See http://nodered.org/docs/security.html for deta
 */
adminAuth: {
  type: "credentials",
  users: [{
    username: "admin"
    password: "$2a$08$zWtXTja0fB1pzD4sHCMyOCMyz2Z6dNbM6t18sJogENOMcxwV
    permissions: "*"
  }]
}
/** The following property can be used to enable HTTPS
AG Help      AO Write Out  AW Where Is  AK Cut       AT Execute   AO Location
AX Exit      AR Read File  AN Replace   AU Paste    AJ Justify   AL Go To Line
```

Figura 68: Líneas de seguridad Node-Red

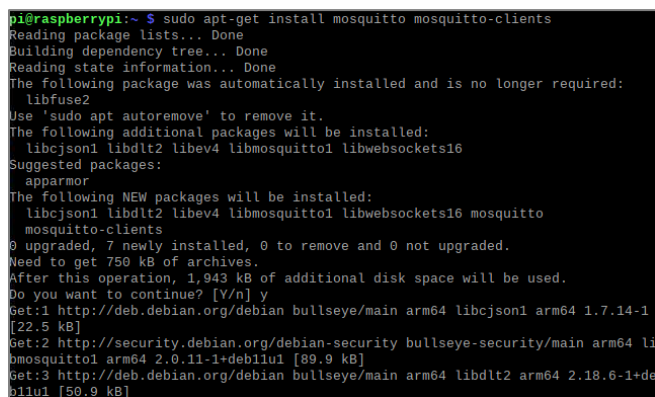
3.15. Instalación de Mosquitto

Antes de realizar con la instalación se procede a actualizar los paquetes de la raspberry pi 4. El Mosquitto es un bróker MQTT con el que se puede enviar mensajes entre clientes y servidores, mediante la técnica de publicación/suscripción. Con la siguiente línea de código

procedemos a instalar Mosquitto: **sudo apt-get install mosquitto mosquitto-clients** (Ver Figura 69).

Con los siguientes comandos verificamos que la instalación de mosquitto se realizó de forma correcta.

```
sudo systemctl status mosquitto
sudo systemctl start mosquitto
sudo systemctl enable mosquitto
```



```
pi@raspberrypi:~$ sudo apt-get install mosquitto mosquitto-clients
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libbson1 libdlt2 libev4 libmosquitto1 libwebsockets16
Suggested packages:
  apparmor
The following NEW packages will be installed:
  libbson1 libdlt2 libev4 libmosquitto1 libwebsockets16 mosquitto
  mosquitto-clients
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 750 kB of archives.
After this operation, 1,943 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
get:1 http://deb.debian.org/debian bullseye/main arm64 libbson1 arm64 1.7.14-1
[22.5 kB]
get:2 http://security.debian.org/debian-security bullseye-security/main arm64 li
bmosquitto1 arm64 2.0.11-1+deb11u1 [89.9 kB]
get:3 http://deb.debian.org/debian bullseye/main arm64 libdlt2 arm64 2.18.6-1+de
b11u1 [50.9 kB]
```

Figura 69: Línea de comando instalación de mosquitto

3.15.1. Configuración de mosquitto

Para ingresar en la carpeta interna de mosquitto copiamos el archivo de configuración de mosquitto y digitamos en la terminal de la raspberry las siguientes líneas de comando:

```
sudo nano mosquitto.conf
sudo cp mosquitto.conf /etc/mosquitto/conf.d/mosquitto.conf
cd /ect/mosquitto/conf.d/
ls
```

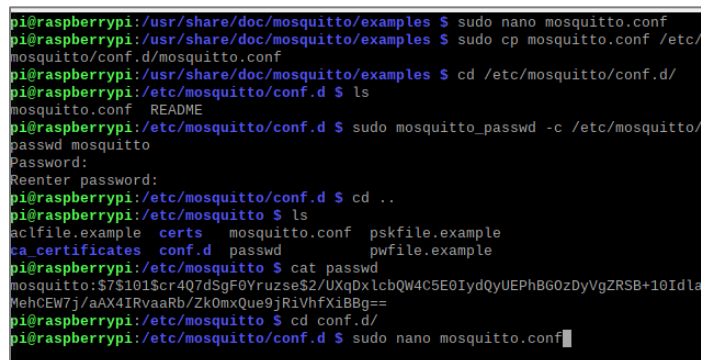
Configuramos el puerto TCP como 1883 que es por donde escucha mosquitto, para que pueda escuchar en la interfaz ethernet, wifi y localhost. Para colocar el usuario y contraseña se ejecuta el siguiente comando:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd
```

Seguido procedemos a digitar el password e ingresamos nuevamente el password, en el directorio de mosquitto verificamos que se encuentre **passwd** con **cd..** y con **cat passwd** visualizamos el passwd encriptado que es el archivo que utilizara el bróker para realizar la autenticación cuando realicemos la conexión(Ver Figura 68).

Con las siguientes líneas de código ingresamos al archivo de configuración de mosquitto donde realizamos las siguientes modificaciones.

```
cd conf.d/  
sudo nano mosquitto.conf
```



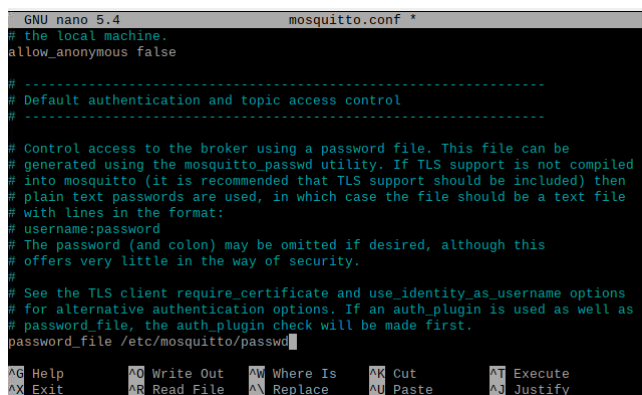
```
pi@raspberrypi:/usr/share/doc/mosquitto/examples $ sudo nano mosquitto.conf  
pi@raspberrypi:/usr/share/doc/mosquitto/examples $ sudo cp mosquitto.conf /etc/  
mosquitto/conf.d/mosquitto.conf  
pi@raspberrypi:/usr/share/doc/mosquitto/examples $ cd /etc/mosquitto/conf.d/  
pi@raspberrypi:/etc/mosquitto/conf.d $ ls  
mosquitto.conf README  
pi@raspberrypi:/etc/mosquitto/conf.d $ sudo mosquitto_passwd -c /etc/mosquitto/  
passwd mosquitto  
Password:  
Reenter password:  
pi@raspberrypi:/etc/mosquitto/conf.d $ cd ..  
pi@raspberrypi:/etc/mosquitto $ ls  
aclfile.example certs mosquitto.conf pskfile.example  
ca_certificates conf.d passwd pwfile.example  
pi@raspberrypi:/etc/mosquitto $ cat passwd  
mosquitto:$7$101$cr4Q7dSgF0Yruzse$2/UxqDxlcBQW4C5E0IydQyUEPhB60zDyVgZRSB+10Idla  
MehCEW7j/aAX4IRvaaRb/Zk0mxQue9jRiVhfXiBBg==  
pi@raspberrypi:/etc/mosquitto $ cd conf.d/  
pi@raspberrypi:/etc/mosquitto/conf.d $ sudo nano mosquitto.conf
```

Figura 70 : Configuración de password

En la configuración de seguridad se modifica lo siguiente, en la línea que contiene **allow_anonymous true**, descomentamos y cambiamos por **allow_anonymous false**, en la

línea `password_file` descomentamos y reemplazamos por `password_file /etc/mosquitto/passwd`, con Ctrl + O procedemos a guardar los cambios realizados y con Ctrl + x salimos del archivo de configuración(Ver Figura 71). Reiniciamos el bróker para que se inicie con los cambios de configuración ejecutando el siguiente comando.

```
sudo systemctl restart mosquitto.service
```



```
GNU nano 5.4 mosquitto.conf *
the local machine.
allow_anonymous false

-----
/ Default authentication and topic access control
-----

/ Control access to the broker using a password file. This file can be
/ generated using the mosquitto_passwd utility. If TLS support is not compiled
/ into mosquitto (it is recommended that TLS support should be included) then
/ plain text passwords are used, in which case the file should be a text file
/ with lines in the format:
/ username:password
/ The password (and colon) may be omitted if desired, although this
/ offers very little in the way of security.
/ See the TLS client require_certificate and use_identity_as_username options
/ for alternative authentication options. If an auth_plugin is used as well as
/ password_file, the auth_plugin check will be made first.
password_file /etc/mosquitto/passwd

^G Help      ^O Write Out  ^W Where Is  ^X Cut       ^T Execute
^X Exit      ^R Read File  ^L Replace   ^U Paste     ^J Justify
```

Figura 71: Configuración de seguridad

3.16. Instalación de InfluxDB

Para instalar InfluxDB debemos descargar y copiar con el siguiente comando en la terminal:

```
curl https://repos.influxdata.com/influxdata-archive.key | gpg --dearmor | sudo tee /usr/share/keyrings/influxdb-archive-keyring.gpg >/dev/null
```

Agregamos la llave del repositorio a la raspberry ejecutando la siguiente línea de comando:

```
echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg] https://repos.influxdata.com/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

Antes de instalar influxdb debemos actualizar los repositorios ejecutando

```
sudo apt update
```

Con el siguiente comando realizamos la instalación de InfluxDB(Ver Figura 72)

```
sudo apt install influxdb
```



```
pi@raspberrypi:~$ echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg] https://repos.influxdata.com/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg] https://repos.influxdata.com/debian bullseye stable
pi@raspberrypi:~$ sudo apt update
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 https://deb.nodesource.com/node_18.x nodistro InRelease
Hit:5 http://archive.raspberrypi.org/debian bullseye InRelease
Get:6 https://repos.influxdata.com/debian bullseye InRelease [7,040 B]
Get:7 https://repos.influxdata.com/debian bullseye/stable armhf Packages [3,417 B]
Get:8 https://repos.influxdata.com/debian bullseye/stable arm64 Packages [6,513 B]
Fetched 17.0 kB in 2s (8,777 B/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~$
```

Figura 72 :Instalación de InfluxDB

Para ejecutar InfluxDB se ejecuta el siguiente comando

```
sudo systemctl start influxdb
```

3.16.1. Configuración de seguridad de InfluxDB

Digitamos en la consola influx, luego ingresamos la siguiente línea para crear el usuario y contraseña otorgándole todos los privilegios, salimos del entorno con exit.

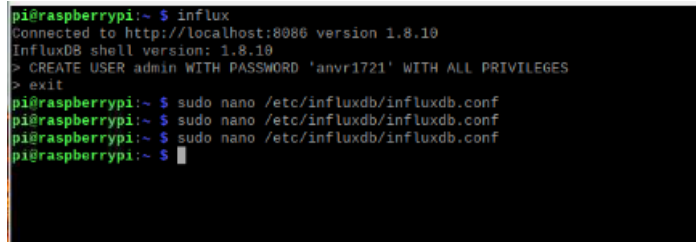
```
CREATE USER admin WITH PASSWORD 'contraseña' WITH ALL PRIVILEGES
```

Modificamos el archivo de configuración para que el servicio de InfluxDB emplee esta configuración con autenticación con el siguiente comando(Ver Figura 73).

```
sudo nano /etc/influxdb/influxdb.conf
```

Estando dentro del archivo de configuración hay que situarnos en **http** y modificamos lo siguiente, con Ctrl+O guardamos los cambios y salimos con Ctrl+X.

```
auth-enabled = true
pprof-enabled = true
pprof-auth-enabled = true
ping-auth-enabled = true
```



```
pi@raspberrypi:~$ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> CREATE USER admin WITH PASSWORD 'anvr1721' WITH ALL PRIVILEGES
> exit
pi@raspberrypi:~$ sudo nano /etc/influxdb/influxdb.conf
pi@raspberrypi:~$ sudo nano /etc/influxdb/influxdb.conf
pi@raspberrypi:~$ sudo nano /etc/influxdb/influxdb.conf
pi@raspberrypi:~$
```

Figura 73: Configuración de seguridad InfluxDB

Para utilizar InfluxDB debemos de reiniciar el servicio ejecutando el siguiente comando

```
sudo systemctl restart influxdb
```

Cuando ingresemos digitamos el usuario y contraseña con el comando

```
influx -username admin -password *****
```

Estando dentro de influxdb creamos la base de datos como CREATE DATABASE IOT, esta base que contiene los datos de fallas de latas se emplea para consultar los datos de la calidad de las latas de atun.

3.16.2. Configuración de InfluxDB en node red

Debemos de instalar los paquetes de software, en la paleta de administración en install buscamos node-red-contrib-influxdb y procedemos a la instalación. Damos un doble clic en el nodo de InfluxDB y clic en el lápiz donde colocamos un nombre a la base de datos, seleccionamos la versión en nuestro caso seria 1.8-flux y en el Host digitamos InfluxDB y

en port 8086, en database escribimos calidad_lata, porque se está ejecutando en la raspberry, digitamos la contraseña y el usuario que se configuro en la base de datos, deshabilitamos la opción de certificado y damos clic en Add para establecer la comunicación entre node red e InfluxDB (Ver Figura 74).

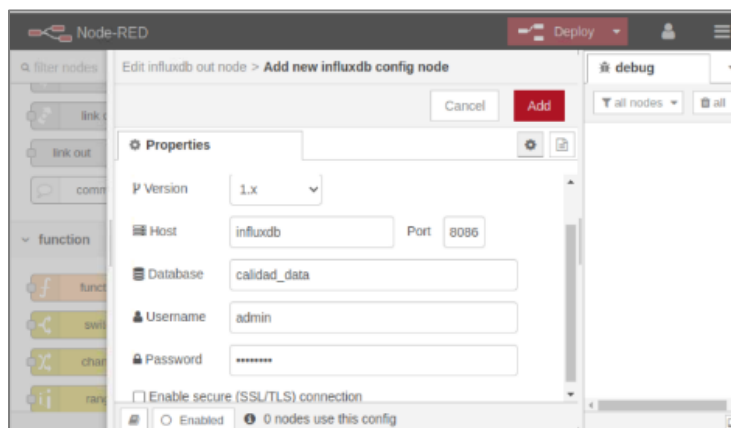


Figura 74: Configuración de propiedades de InfluxDB

3.17. Instalación de Grafana

Realizamos la actualización de los paquetes de la raspberry, luego descargamos los archivos para la instalación de grafana ejecutando las siguientes líneas de comando(Ver Figura 75).

```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/enterprise/release/grafana-enterprise_10.1.5_arm64.deb
sudo dpkg -i grafana-enterprise_10.1.5_arm64.deb
```

```
pi@raspberrypi:~$ wget https://dl.grafana.com/enterprise/release/grafana-enterprise_10.1.5_arm64.deb
--2023-10-17 00:43:43-- https://dl.grafana.com/enterprise/release/grafana-enterprise_10.1.5_arm64.deb
Resolving dl.grafana.com (dl.grafana.com)... 151.101.54.217, 2a04:4e42:d::729
Connecting to dl.grafana.com (dl.grafana.com)|151.101.54.217|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 103057082 (98M) [application/octet-stream]
Saving to: 'grafana-enterprise_10.1.5_arm64.deb'

grafana-enterprise_100%[=====] 98.28M 3.74MB/s in 35s
2023-10-17 00:44:20 (2.80 MB/s) - 'grafana-enterprise_10.1.5_arm64.deb' saved [103057082/103057082]
pi@raspberrypi:~$
```

Figura 75: Instalación de Grafana

Para que grafana se inicie cuando se ejecute la raspberry, habilitamos el servicio con el comando, con esto grafana se inicia cuando se arranque la raspberry(Ver Figura 76).

```
$ sudo systemctl enable grafana-server
```



```
pi@raspberrypi:~$ sudo systemctl enable grafana-server
Synchronizing state of grafana-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable grafana-server
pi@raspberrypi:~$ wget https://dl.grafana.com/oss/release/grafana-rpi_8.0.6_armhf.deb
--2023-11-05 12:20:28-- https://dl.grafana.com/oss/release/grafana-rpi_8.0.6_armhf.deb
Resolving dl.grafana.com (dl.grafana.com)... 151.101.54.217, 2a04:4e42:d::729
Connecting to dl.grafana.com (dl.grafana.com)[151.101.54.217]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 52076928 (50M) [application/vnd.debian.binary-package]
Saving to: 'grafana-rpi_8.0.6_armhf.deb'

grafana-rpi_8.0.6_a 100%[=====] 49.66M 3.24MB/s in 18s

2023-11-05 12:20:48 (2.73 MB/s) - 'grafana-rpi_8.0.6_armhf.deb' saved [52076928/52076928]

pi@raspberrypi:~$
```

Figura 76: Comando ejecución de grafana

Con la instalación de grafana en la raspberry se puede acceder desde un navegador web digitando la siguiente dirección: <http://192.168.1.8:3000>, para iniciar sesión hay que ingresar el usuario y contraseña para realizar el dashboards del control de calidad.

3.17.1. Conexión fuente de datos InfluxDB a Grafana

Para la configuración de grafana en la pantalla principal damos clic data sources y agregamos una nueva que en nuestro caso sería el **InfluxDB**, primero le damos un nombre a data sources, luego definimos el lenguaje que emplearemos, seguido en el Url digitamos la dirección colocando la IP de la raspberry, y el puerto 8086 donde se encuentra ejecutando el influxdb. Activamos la autenticación dando clic en Basic auth, luego en Basic Auth Details colocamos el usuario y la contraseña.

Especificamos a que base de datos se quiere el acceso en este caso sería iot donde seguido colocaremos nuevamente el usuario y la contraseña luego damos clic en guardar y tenemos listo grafana para utilizar.

3.18. Desarrollo del sistema de visión artificial

Para el desarrollo del sistema de visión artificial se emplearán 5 etapas detalladas a continuación:

- **Transporte:** se empleará una banda transportadora que se encarga de movilizar las latas de atun hasta el sistema de visión y luego al sistema de clasificación.
- **Control de señales:** para controlar las señales utilizaremos la placa de control Arduino que se encargara del sistema de activación de los servomotores del brazo robótico y del actuador que controla en movimiento de la banda transportadora.
- **Reconocimiento:** se implementará un sistema de visión artificial que a través de la cámara raspberry HQ y desarrollo de un algoritmo de detección de objetos y el procesamiento de la imagen, determinar la calidad de cada lata.
- **Clasificación:** el sistema de clasificación la realizara el brazo robótico donde determinara su ubicación de acuerdo a los parámetros establecidos.
- **Comunicación:** se realizará la comunicación entre la raspberry Pi 4 que posee el algoritmo y el Arduino que controlará los servomotores del brazo robótico. Para el desarrollo del algoritmo de visión artificial se utilizará el software de Python para realizar las diferentes líneas de código debido a que soporta orientación a objetos, siendo un lenguaje ideal para el análisis de datos como imágenes o textos.

El programa Makesense.AI donde realizaremos las anotaciones de cada una de las imágenes de nuestro dataset, para el entrenamiento del algoritmo utilizaremos la herramienta de Google Colab que facilita los entornos de ejecución como GPU y TPU, también permite importar nuestro propio dataset que utilizaremos para realizar el entrenamiento de la lata de atun y evaluar el modelo.

3.18.1. Construcción del dataset del modelo de Visión Artificial

Antes de continuar con la construcción del dataset, clonamos el siguiente repositorio con dirección url en el terminal de la raspberry: <https://github.com/AprendeIngenia/Inferencia-con-YOLOv5>, luego procedemos a descomprimir para obtener la carpeta que contiene los archivos necesarios para ejecutar el algoritmo de detección de objetos. Procedemos a renombrar la carpeta como TESIS_OBJECT_DETECTION_YOLOV5 (Ver Figura 77).

Dentro del archivo data creamos una carpeta con el nombre de **images** que contiene todas las imágenes de latas de atun que utilizaremos, previamente las imágenes fueron capturadas con la cámara Raspberry HQ y una carpeta con el nombre de **labels** donde se guardarán las anotaciones.

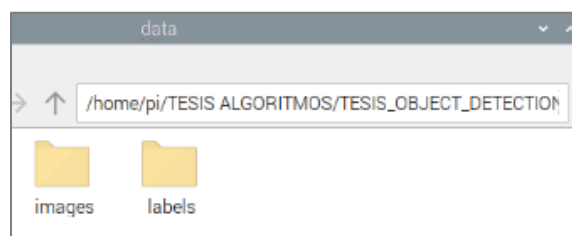


Figura 77: Carpetas para Imágenes y Anotaciones

La carpeta data, contiene el 100% de imágenes, debemos de seleccionar el 80 % para el entrenamiento y el 20% para la validación. El archivo de “train” la utilizaremos para entrenar el modelo y el archivo “val” es la que contiene imágenes que no se emplearon en el entrenamiento pero que se las utilizara para evaluar que tan preciso es el modelo.

El software MAKESENSE, proporciona un entorno donde importamos las imágenes para realizar las anotaciones de cada estado de calidad de las latas de atun, damos clic en **Get Started** proporcionándonos la opción de importar las imágenes que servirán para realizar las anotaciones de cada lata de atun, damos clic en **Drop images** (Ver Figura 78)

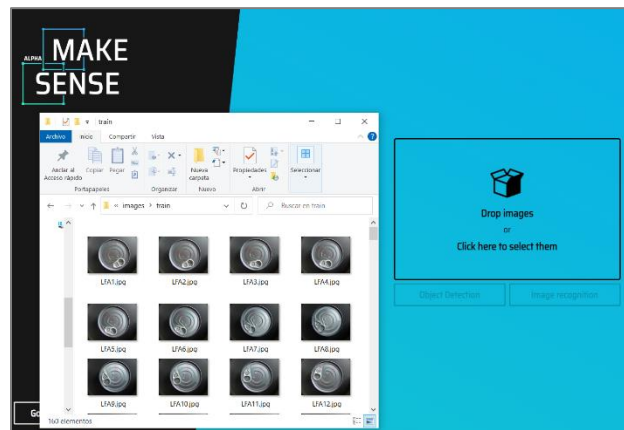


Figura 78: Importacion de Imágenes al software Makesense

Luego damos clic en **Object Detection** donde se nos abrirá una ventana para crear las etiquetas, le damos un clic en el símbolo “+” y escribimos el nombre de cada etiqueta (Ver Figura 79).

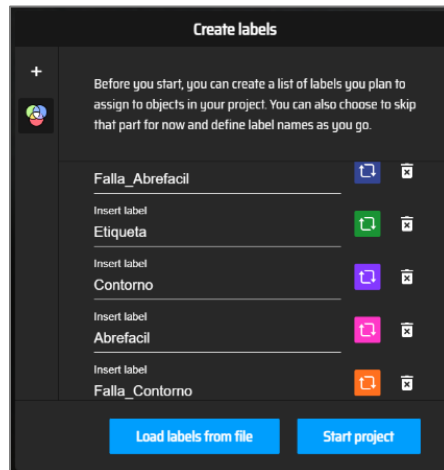


Figura 79: Creación de labels de las imágenes

Las etiquetas quedan establecidas de la siguiente manera:

- Falla_Abrefacil
- Falla_Contorno
- Falla_Etiqueta
- Abrefácil
- Etiqueta
- Contorno

Damos clic en Start Project que proporciona un entorno donde realizamos las anotaciones de las latas de atun, seleccionamos la opción de **Rect** y en cada imagen encerramos el área de interés para determinar los diferentes estados de calidad del producto con sus respectivas etiquetas, este proceso se realiza con cada una de las imágenes que contiene la carpeta data.

Damos un clic en **Rect** y nos muestra el siguiente entorno donde realizamos la segmentación de cada imagen, en esta imagen encerramos en un rectángulo el abrefácil que posee falla en la ubicación, su etiqueta y contorno pasan el control de calidad (Ver Figura 80).



Figura 80: Etiqueta Falla Abrefácil

En la siguiente imagen encerramos en un rectángulo la falla que presenta el contorno de la tapa de la lata, el abrefácil se encuentra bien posicionado y la etiqueta posee los datos correspondientes (Ver Figura 81).

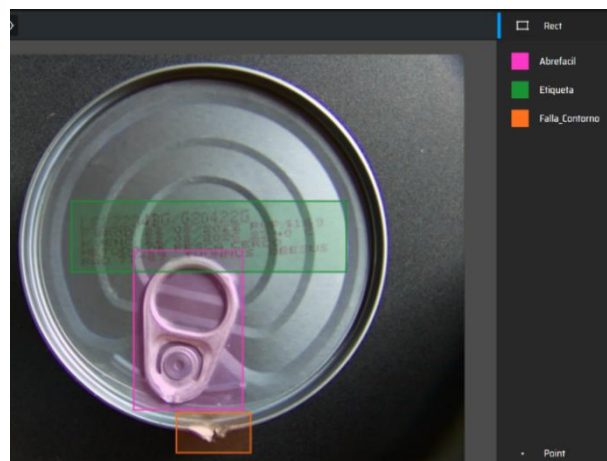


Figura 81: Etiqueta Falla Contorno

En la siguiente figura realizamos el rectángulo del abrefácil, contorno y etiqueta, apreciamos que se encuentran con los estándares de calidad establecidos (Ver Figura 82).

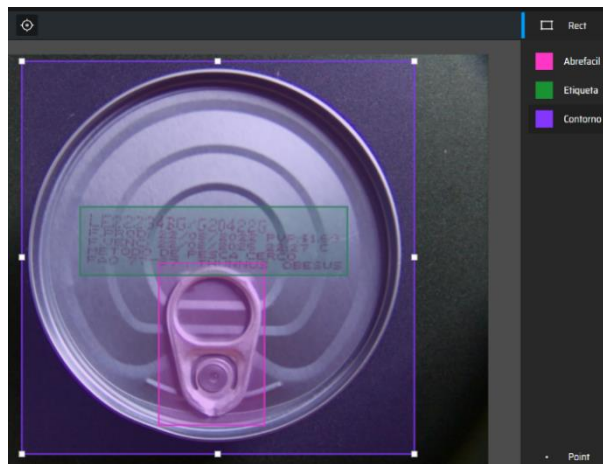


Figura 82 : Etiqueta en buen estado

En la figura vemos ausencia de la etiqueta, procedemos a colocar el rectángulo, el abrefácil y contorno pasan los controles de calidad (Ver Figura 83).

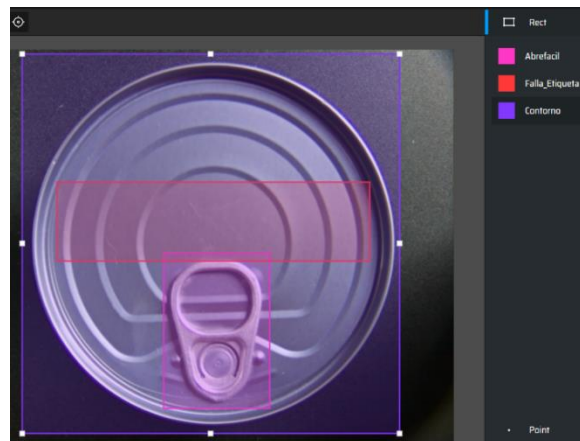


Figura 83: Falla en Etiqueta

Cuando se realicen todas las anotaciones de cada una de las imágenes debemos de dar clic en **Actions** y elegir **Export Annotations** donde seleccionaremos el casillero de ‘A.zip package containing files in YOLO format’ y damos clic en **Export** (Ver Figura 84).

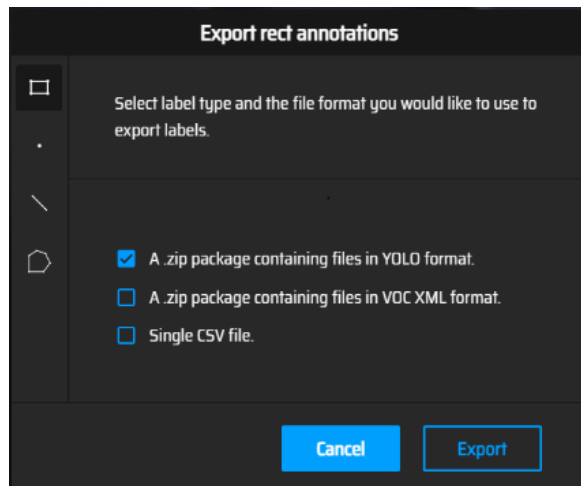


Figura 84: Exportación de las anotaciones de imágenes

3.19. Entrenamiento del modelo de object detection en Google Colab

Ingresamos en nuestro navegador a la página de Google Colab, donde seleccionamos en entorno de ejecución para cambiar el tipo de acelerador por hardware dando la opción de escoger entre NONE - GPU o TPU, en nuestro programa escogeremos GPU porque la ejecución del programa es más rápida (Ver Figura 85).

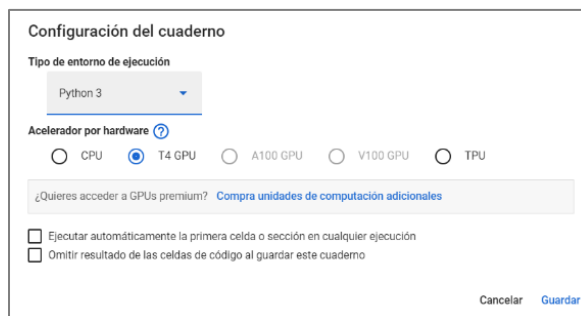
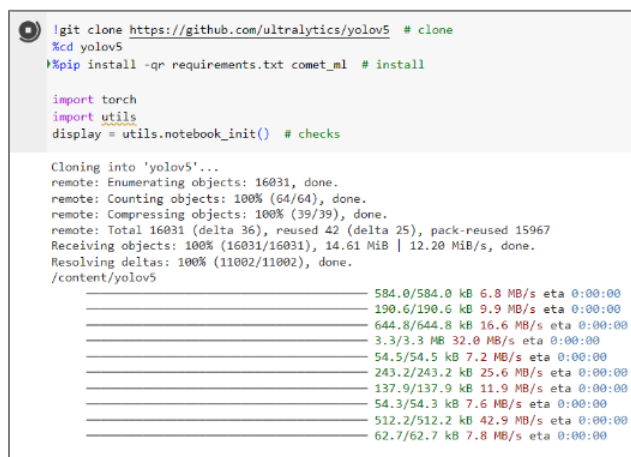


Figura 85 : Selección de GPU en Google Colab

Abrimos el archivo OBJECT_DETECTION_YOLOv5.ipynb en Google Colab, empleamos este tipo de red de entrenamiento porque nos permite trabajar con diferentes clases. En la primera línea de código debemos de clonar el repositorio de Ultralytics, se crea la carpeta yolov5 e instalamos los requerimientos que necesitaremos para el entrenamiento (Ver Figura 86).

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
!pip install -qr requirements.txt comet_ml # install
import torch
import utils
display = utils.notebook_init() # checks
```

Importamos la librería torch y utils necesarias para seguir con el entramiento de la detección de cada una de las imágenes.



```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
!pip install -qr requirements.txt comet_ml # install

import torch
import utils
display = utils.notebook_init() # checks

Cloning into 'yolov5'...
remote: Enumerating objects: 16031, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 16031 (delta 36), reused 42 (delta 25), pack-reused 15967
Receiving objects: 100% (16031/16031), 14.61 MiB | 12.20 MiB/s, done.
Resolving deltas: 100% (11002/11002), done.
/content/yolov5
----- 584.0/584.0 kB 6.8 MB/s eta 0:00:00
----- 190.6/190.6 kB 9.9 MB/s eta 0:00:00
----- 644.8/644.8 kB 16.6 MB/s eta 0:00:00
----- 3.3/3.3 MB 32.0 MB/s eta 0:00:00
----- 54.5/54.5 kB 7.2 MB/s eta 0:00:00
----- 243.2/243.2 kB 25.6 MB/s eta 0:00:00
----- 137.9/137.9 kB 11.9 MB/s eta 0:00:00
----- 54.3/54.3 kB 7.6 MB/s eta 0:00:00
----- 512.2/512.2 kB 42.9 MB/s eta 0:00:00
----- 62.7/62.7 kB 7.8 MB/s eta 0:00:00
```

Figura 86 : Clonación del repositorio Ultralytics yolov5

Cargamos a Google Colab la carpeta data.zip que contiene las imágenes y anotaciones para descomprimir en una línea del código de entrenamiento de la detección de objetos (Ver Figura 87).

```
!unzip -q /content/data.zip -d /content/
```

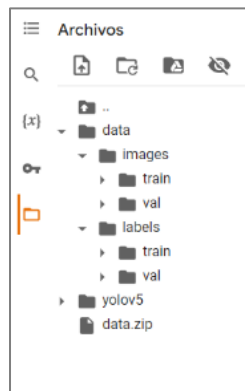


Figura 87 : Carpeta data images y labels

Se instalan otras librerías necesarias para seguir con el entrenamiento.

Matplotlib: es una librería de Python que permite crear o personalizar tipos de gráficos a partir de vectores y arrays.

Opencv-Python: es una API de Python para Opencv, posee una interfaz con varios lenguajes.

Numpy: es una librería empleada para el cálculo numérico o el análisis de datos, su procesamiento con los arrays las realiza de forma rápida.

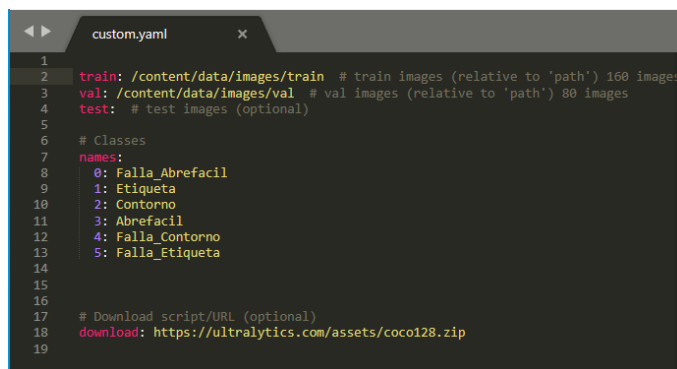
Ipykernel: permite ejecutar código de Python dentro de Júpiter u otros notebooks, por poseer

esta funcionalidad la convierte en un invaluable para el aprendizaje automático y el análisis de datos.

Torch: esta librería contiene las funciones que se necesitan para que Yolo funcione de forma correcta.

Pandas: la librería se emplea para crear los dataframe, esta contiene los datos que se emplearan para el entrenamiento.

En la carpeta yolov5 vamos a descargar **coco128.yaml** y luego realizamos la modificación con el editor Sublime Text 3, en **train** y **val** cambiamos la dirección que contienen a estas carpetas y en **names** digitamos cada una de las etiquetas del estado de calidad de las latas y guardamos, luego procedemos a cargar este archivo en la carpeta data de Google Colab donde estamos realizando el entrenamiento(Ver Figura 88).



```
1
2 train: /content/data/images/train # train images (relative to 'path') 160 images
3 val: /content/data/images/val # val images (relative to 'path') 80 images
4 test: # test images (optional)
5
6 # Classes
7 names:
8 0: Falla_Abrefacil
9 1: Etiqueta
10 2: Contorno
11 3: Abrefacil
12 4: Falla_Contorno
13 5: Falla_Etiqueta
14
15
16
17 # Download script/URL (optional)
18 download: https://ultralytics.com/assets/coco128.zip
19
```

Figura 88 : Modificación de archivo coco128.yaml

Para el entrenamiento se escogió la arquitectura de yolov5x (XLarge), por ser pequeño y tener una velocidad para realizar las detecciones cuando en la tarjeta raspberry en tiempo real.

Realizamos el entrenamiento con las imágenes de nuestro propio dataset, con la siguiente línea de código se empieza con el entrenamiento mediante el algoritmo **train.py**, el **img** especifica el tamaño de la entrada de la imagen que es 640, el **batch** que es el tamaño del Batch de 16, **epochs** representa las iteraciones del entrenamiento que para nuestro modelo sería de 200 épocas, el **data** contiene el archivo custom.yaml, en su interior encontramos las carpeta de entrenamiento y validación con los nombres de las etiquetas y **weights** donde se guardan los modelos entrenados para próximas ejecuciones (Ver Figura 89).

```
# Train YOLOv5s on COCO128 for 200 epochs
!python train.py --img 640 --batch 16 --epochs 200 --data /content/yolov5/data/custom.yaml --weights yolov5x.pt --cache
```

```
30 epochs completed in 0.175 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 173.1MB
Optimizer stripped from runs/train/exp/weights/best.pt, 173.1MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 322 layers, 86207059 parameters, 0 gradients, 203.9 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	80	240	0.95	0.802	0.883	0.658
Falla_Abrefacil	80	20	1	0.909	0.972	0.828
Etiqueta	80	60	1	0.992	0.995	0.842
Contorno	80	60	1	0.709	0.853	0.659
Abrefacil	80	60	0.897	1	0.976	0.876
Falla_Contorno	80	20	0.974	1	0.995	0.551
Falla_Etiqueta	80	20	0.827	0.2	0.500	0.193

```
Results saved to runs/train/exp
COMET INFO: -----
COMET INFO: Comet.ml OfflineExperiment Summary
COMET INFO: -----
COMET INFO: Data:
COMET INFO:   display_summary_level : 1
COMET INFO:   url                   : [OfflineExperiment will get URL after upload]
COMET INFO: Metrics [count] (min, max):
COMET INFO:   loss [93]              : (0.15458714962005615, 0.8688702583312988)
COMET INFO:   metrics/mAP_0.5 [60]  : (0.15080876377946048, 0.9101758581405445)
COMET INFO:   metrics/mAP_0.5:0.95 [60] : (0.043746094543242615, 0.6577578105197274)
COMET INFO:   metrics/precision [60] : (0.015711299300093507, 0.9659795410702109)
COMET INFO:   metrics/recall [60]    : (0.275, 0.8873835769926304)
COMET INFO:   train/box_loss [60]    : (0.02753448486328125, 0.10215472429990768)
COMET INFO:   train/cls_loss [60]   : (0.008312088437378407, 0.05282182618975639)
COMET INFO:   train/obj_loss [60]   : (0.020065320655703545, 0.049939122051000595)
COMET INFO:   val/box_loss [60]     : (0.021187862381339073, 0.07216823101043701)
COMET INFO:   val/cls_loss [60]    : (0.006806130520959432, 0.04019925441990362)
COMET INFO:   val/obj_loss [60]    : (0.000895771408081095, 0.02383260154724121)
COMET INFO:   x/lr0 [60]           : (0.0007599999999999998, 0.07075000000000001)
COMET INFO:   x/lr1 [60]           : (0.0007599999999999998, 0.009262166666666669)
COMET INFO:   x/lr2 [60]           : (0.0007599999999999998, 0.009262166666666669)
```

Figura 89:Entrenamiento con YOLOV5

3.19.1. Descarga del modelo de entrenamiento para raspberry pi 4

Cuando terminamos el entrenamiento del sistema de visión de calidad de latas de atun en Google Colab ejecutamos las siguientes líneas de código para descargar el archivo **best.pt** de la red neuronal, que podemos leer con el marco de aprendizaje pytorch (Ver Figura 90).

```
#export your model's weights for future use
from google.colab import files
files.download('./runs/train/exp/weights/best.pt')
```

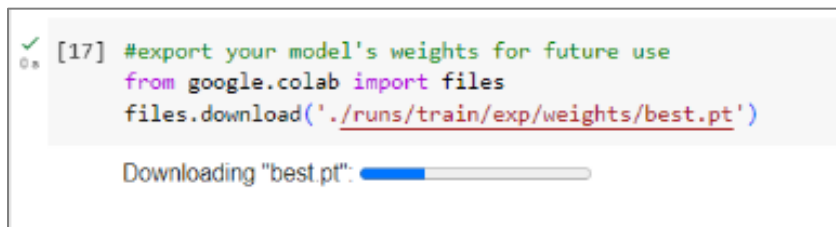


Figura 90 : Descarga de archivo de entrenamiento best.pt

3.19.2. Algoritmo de Detección de Objetos en Tiempo Real

Antes de ejecutar el algoritmo debemos de instalar ciertos requerimientos con la siguiente línea de comando en la terminal de la raspberry pi 4.

```
pip install -r https://raw.githubusercontent.com/ultralytics/yolov5/master/requirements.txt
```

Para ejecutar el programa de detección de objetos en latas de atun, abrimos la carpeta TESIS ALGORITMOS/TESIS_OBJECT_DETECTION_YOLOV5 donde encontramos el archivo ObjectDetect.py que contiene las líneas de código detalladas a continuación.

Abrimos el archivo en Thonny y lo primero que hacemos es importar las librerías de torch, opencv y numpy que son importantes para la ejecución del algoritmo. EL picamera2 sirve para tener una comunicación con la cámara por eso importamos Picamera2.

```
# Importamos librerías
import torch
import cv2
from picamera2 import Picamera2
import numpy as np
import pandas
```

Cargamos el modelo **best.pt** que previamente descargamos del entrenamiento realizado en Google Colab.

```
# Leemos el modelo
model = torch.hub.load('ultralytics/yolov5', 'custom',
    path = '/home/pi/TESIS ALGORITMOS/TESIS_OBJECT_DETECTION_YOLOV5/model/best.pt')
```

Las siguiente líneas es para darle ciertas configuraciones a la cámara definimos los valores de nuestra imagen que es este caso sería de 640x480, con la siguiente línea creamos un objeto de cámara y que esta es igual a Picamera2, la siguiente línea es para darle ciertas configuraciones a la cámara del tamaño de la imagen previamente definidos, seguido definimos el formato RGB888 que serían los bits de cada color, con la siguiente línea definimos una vista previa de la cámara y con picam2.start la cámara estará encendida.

```
dispW=640
dispH=480

picam2=Picamera2()
picam2.preview_configuration.main.size=(dispW,dispH)
picam2.preview_configuration.main.format='RGB888'
picam2.preview_configuration.align()
picam2.configure("preview")
picam2.start()
```

Para la activación de la cámara utilizamos el comando: cv2.VideoCapture(unicam),el parámetro “unicam” especifica el periférico que estamos utilizando. Definimos la altura y ancho de cada fotograma en la transmisión del video, se trabajará con 30 fotogramas por segundo. En la variable unicam se registrará cada frame que es captado por la cámara, para poder visualizar lo que se está procesando por la banda transportadora en tiempo real.

```
#Realizamos Videocaptura
unicam= '/dev/video0'
cap=cv2.VideoCapture(unicam)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, dispW)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, dispH)
cap.set(cv2.CAP_PROP_FPS, 30)
```

En las siguientes líneas de código establecemos un bucle donde establecemos el frame de cada imagen para capturar su matriz dimensional de filas y columnas de la que está compuesta una imagen.

```
while True:
    #Realizamos la lectura de frame
    frame=picam2.capture_array()
    frame=cv2.flip(frame, -1)
```

En la variable **detect** se guardan todos los frame de las detecciones, también obtenemos la información del objeto en el espacio esto sería el resultado de las detecciones, la variable **info** contiene los cuadros delimitadores que se encierran según el objeto que se detecte.

```
# Realizamos las detecciones
detect = model(frame)

info = detect.pandas().xyxy[0] # im1 predictions
print(info)
```

Procedemos a visualizar la imagen con **cv2.imshow**, mostramos los fotogramas por segundos, cuando se detecte la imagen aparece una leyenda con el nombre de **Detector de Latas**, para que la red neuronal pueda dibujar las latas debemos de utilizar la librería **numpy(np)**.

```
# Mostramos FPS
cv2.imshow('Detector de Latas', np.squeeze(detect.render()))
```


En la siguiente línea hay una condición que permitirá salir del programa siempre que digitemos el número 27, el break sirve para salir del ciclo cuando presionamos el número 27.

```
#Leemos el teclado
if cv2.waitKey(1)&0xFF==27:
    break
```

Las líneas siguientes la utilizamos para liberar la captura del video y cerrar las ventanas abiertas que se están visualizando y poder realizar otra verificación de otro algoritmo.

```
cap.release()
cv2.destroyAllWindows()
```

Luego se visualiza en nuestro monitor las imágenes que al activarse la cámara proporciona la detección de calidad de las latas de atun previamente entrenados en donde visualizamos un rectángulo dibujado sobre las latas dependiendo del estado de calidad siendo estas, fallas en el abrefácil, latas con daños en el contorno, latas que no tienen etiqueta y latas en buen estado de calidad refiriéndose a las que no tienen las fallas anteriormente mencionadas.

3.19.3. Entrenamiento del modelo reconocimiento de texto en Google Colab y aplicación de EasyOCR.

Primero instalamos EasyOCR que contiene todas las librerías a utilizar para el entrenamiento

```
!pip install easyocr
```

Importamos la librería EasyOCR que es necesaria para la detección del modelo.

```
import easyocr
```

```
#download the model
reader = easyocr.Reader(['en','es'], gpu = True)
```

Establecemos la importación de la carpeta EasyOCR en Google drive.

```
from google.colab import drive
drive.mount('/content/drive')
!unzip
```

Importamos las siguientes librerías como PIL y ImageDraw, mostramos la imagen para realizar el reconocimiento (Ver Figura 91).

```
# show an image
import PIL
from PIL import ImageDraw
im = PIL.Image.open("/content/drive/MyDrive/EasyOCR/images/LOCR_40.png")
```



Figura 91: Etiqueta de codificación OCR

Realizamos el OCR y obtenemos los cuadros delimitadores de texto que se encuentra en la parte superior de la etiqueta, con el siguiente comando (Ver Figura 92).

```
# Doing OCR. Get bounding boxes.
bounds = reader.readtext('/content/drive/MyDrive/EasyOCR/images/LOCR_40.png')
bounds
```

```

[[[177, 199], [400, 199], [400, 225], [177, 225]],
 'LE222348G/ G20422G',
 0.40105420271798625),
 ([[182, 220], [392, 220], [392, 244], [182, 244]],
 'F.PROD: 22/08/2022',
 0.44098132994964045),
 ([[410, 218], [518, 218], [518, 242], [410, 242]],
 'PVP:$1.69',
 0.9138708990030512),
 ([[183, 243], [389, 243], [389, 263], [183, 263]],
 'F.VENC: 22/08/2026',
 0.4440168442187067),
 ([[415, 241], [477, 241], [477, 261], [415, 261]],
 '23:40',
 0.5688945729495297),
 ([[183, 267], [267, 267], [267, 285], [183, 285]],
 'Metodo',
 0.3559444080209504),
 ([[272, 260], [446, 260], [446, 284], [272, 284]],
 'DE Pesca CERco',
 0.16831157752969292),
 ([[181, 285], [285, 285], [285, 305], [181, 305]],
 'FAO 77-87',
 0.46857200993766635),
 ([[304, 282], [493, 282], [493, 303], [304, 303]],
 'Thunnus OBESus',
 0.2574585833745286)]

```

Figura 92: Cuadros delimitadores de texto

En la siguiente línea del algoritmo dibujamos los cuadros delimitadores en cada línea de texto reconocida en la etiqueta de la lata (Ver Figura 93).

```

# Draw bounding boxes
def draw_boxes (image, bounds, color='blue', width=2):
    draw = ImageDraw.Draw(image)
    for bound in bounds:
        p0, p1, p2, p3 = bound[0]
        draw.line([*p0, *p1, *p2, *p3, *p0], fill=color, width=width)
    return image
draw_boxes(im, bounds)

```

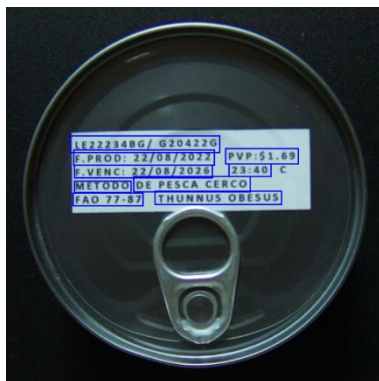
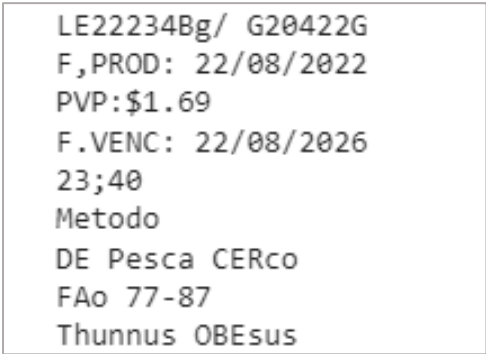


Figura 93: Cuadro delimitador del texto

Con el ciclo **for** mediante la variable **i** en el **bounds** se imprime el texto reconocido en el modelo entrenado (Ver Figura 94).

```
len(bounds)
for i in bounds:
    print(i[1])
```



```
LE22234Bg/ G20422G
F,PROD: 22/08/2022
PVP:$1.69
F.VENC: 22/08/2026
23;40
Metodo
DE Pesca CERco
FAo 77-87
Thunnus OBEsus
```

Figura 94: Reconocimiento de etiqueta

3.20. PRUEBAS Y RESULTADOS.

3.20.1. Pruebas de funcionamiento

Con el modelo de IA con 200 epocas se detalla información de la inspección de visión con 10 muestras de latas que al realizar el proceso de detección proporcionan datos de las pruebas realizadas. Los resultados muestran los estados de calidad de cada lata procesada, superando el 86% de precisión, como podemos ver en la siguiente tabla donde se puede verificar si la lata se encuentra en buen estado o presenta alguna falla al momento de su detección (Ver Tabla 16).

Tabla 16 Pruebas del modelo de IA con 200 épocas.

N° de Prueba	Detección latas sistema de IA	Observaciones											
		Falla Abrefácil	Precisión %	Falla Contorno	Precisión %	Falla Etiqueta	Precisión %	Abrefácil Ok	Precisión %	Contorno OK	Precisión %	Etiqueta Ok	Precisión %
1	Lata1							✓	96	✓	86	✓	93
2	Lata 2	✓	98							✓	95	✓	91
3	Lata 3			✓	90			✓	85			✓	94
4	Lata 4	✓	92							✓	89	✓	96
5	Lata 5					✓	90	✓	92	✓	88		
6	Lata 6			✓	95			✓	97			✓	89
7	Lata 7			✓	89			✓	93			✓	91
8	Lata 8							✓	96	✓	92	✓	82
9	Lata 9	✓	95							✓	91	✓	94
10	Lata10					✓	98	✓	97	✓	96		

3.20.2. Resultados en tiempo real del sistema de visión con cámara Raspberry

HQ.

En la imagen se observa 3 detecciones por parte de la cámara del sistema de visión, se visualiza una falla que es la falta de etiqueta con un 69 % de precisión, la etiqueta contiene datos de producción de la lata y también se detecta el buen estado de calidad en su contorno y el abrefácil con porcentajes superiores al 70 y 96 % respectivamente. En la figura se detalla en un recuadro las coordenadas mínimas y máximas con el porcentaje de confiabilidad y la clase a la que pertenece cada etiqueta (Ver Figura 95).

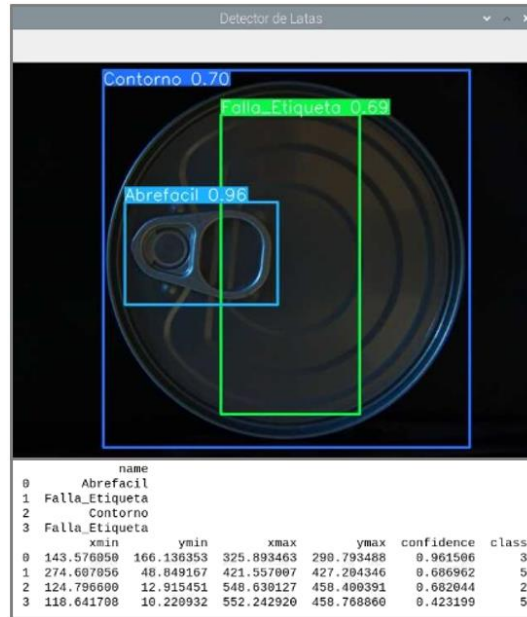


Figura 95 : Resultado de inspección falla en etiqueta

En la siguiente figura observamos la detección de una falla con filamentos en el contorno del sellado de la lata con un porcentaje de precisión del 95% , también se detecta la etiqueta con un porcentaje de 94% y el abrefácil con el 97 % , en la imagen se detallan datos del porcentaje de confiabilidad y las coordenadas de cada bounding box (Ver Figura 96).

Figura 96: Resultado de inspección falla en contorno

En la siguiente imagen se detectó una falla en el abrefácil con un porcentaje del 96%, la ubicación existente puede provocar una abertura de la tapa de sellado por ejercer presión al moverse de su sitio, el contorno se encuentra con un porcentaje de confiabilidad del 96% y la etiqueta se encuentra con un 94%, estos valores de porcentajes son altos relacionados a la precisión que se tiene al momento de las detecciones de las latas. En la parte inferior de la figura se detalla el número de clases a la que pertenece cada etiqueta, el porcentaje de confiabilidad y las coordenadas de la ubicación del rectángulo como son sus mínimos y máximos (Ver Figura 97).

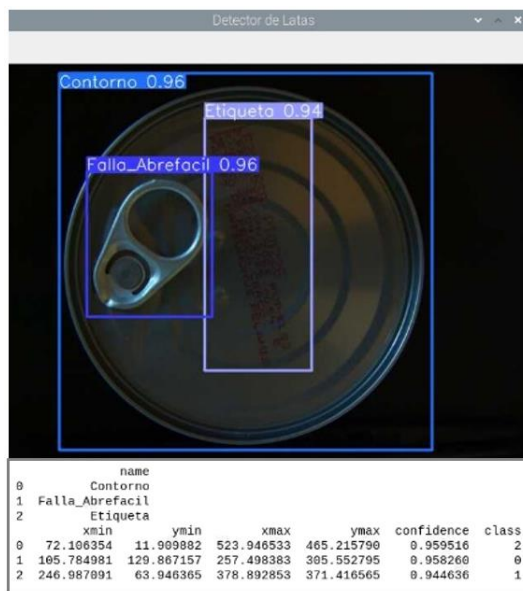


Figura 97: Resultado de inspección falla en abrefácil

En la figura se tiene como resultado una lata con su abrefácil, contorno y etiqueta que se encuentra en óptimo estado de calidad, en la parte inferior se detalla datos informativos como clase, precisión y mínimos y máximos de cada etiqueta (Ver Figura 98).

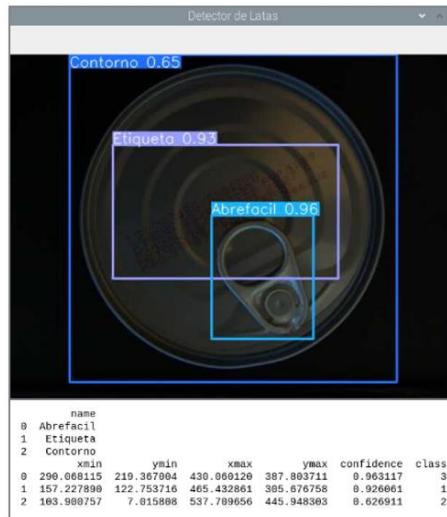


Figura 98: Resultado de inspección lata en buen estado de calidad

3.20.3. División de las clases etiquetadas

En la imagen observamos que cuando se cumplen las 40 iteraciones las barras referentes a las detecciones de las 3 fallas se mantienen a un mismo nivel, mientras que avanza el entrenamiento, cuando se llega a las 120 iteraciones, visualizamos en las barras de etiqueta, contorno y abrefácil llegan a otro nivel, las etiquetas que se detallan son la representación de una lata en óptimas condiciones de calidad (Ver Figura 99).

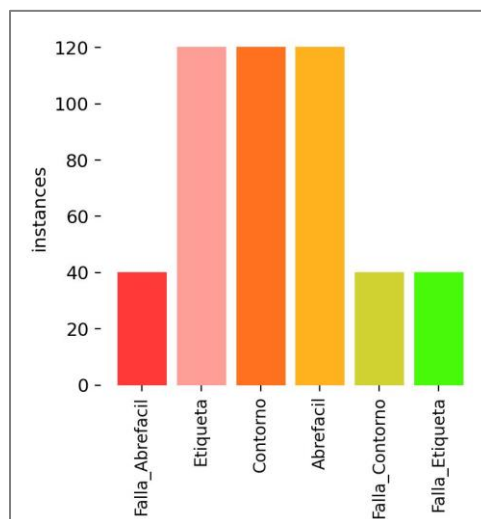


Figura 99 : División de las clases etiquetadas

3.20.4. Analisis de métricas de precisión con 200 epocas

Los resultados del entrenamiento con 200 iteraciones proporcionan gráficas de las métricas de precisión, recall, mAP0.5 y mAP0.5:0.95, con estos valores se busca que el modelo tenga valores de mejor precisión promedio media(mAP). En las metrics/precisión cuando se alcanza las 200 epocas se obtiene resultados de una precisión que bordea el 99%, es decir los bounding box fueron correctos cuando se detectó una lata.

En la gráfica de metrics/recall (métricas/sensibilidad) observamos que, al cumplirse todas las iteraciones, el resultado que proporciona son los verdaderos positivos como los falsos negativos. El TP(verdaderos positivos) y el FN(falsos negativos), el primero se emplea para mostrar que la imagen tuvo una detección correcta, el segundo es para no etiquetar la imagen cambiada.

Las metrics/mAP0.5 al determinar valores superiores a 0.5 en sus detecciones, sus predicciones se consideran en un umbral de aceptación en relación a las predicciones detectadas, mientras que las metrics/Map0.5:0.95 cuando se detectan predicciones superiores a 0.95 son consideradas como perfectas en sus detecciones. En el entrenamiento se obtuvo un 98% de precisión (Ver Figura 100).

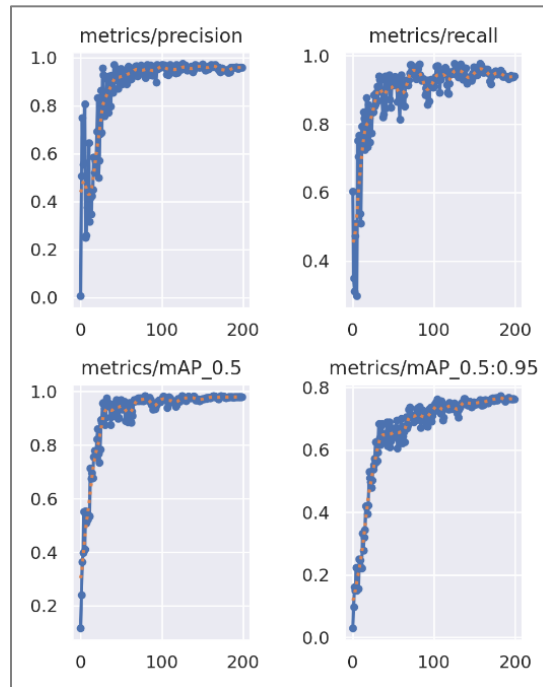


Figura 100: Graficas de metric/precision-recall-mAP0.5-mAP0.5:0.95

3.20.5. Matriz de confusión

En la figura se detalla una matriz que representa a las detecciones de cada lata al momento de realizar las validaciones durante el entrenamiento. Se visualiza cuantas veces nuestro modelo tuvo aciertos en la detección del objeto y cuantas veces tuvo confusión, el bounding box que se obtiene al momento de detectar y las veces que el modelo fallo en su totalidad en relación a las detecciones. En cada fila se detallan las clases verdaderas y en las columnas las clases de predicción, en la diagonal se encuentran los verdaderos positivos y en la parte externa los falsos negativos y falsos positivos. En la parte inferior de la imagen se tiene los

falsos negativos, mientras que en el eje de las *y* se ubican los falsos positivos (Ver Figura 101).

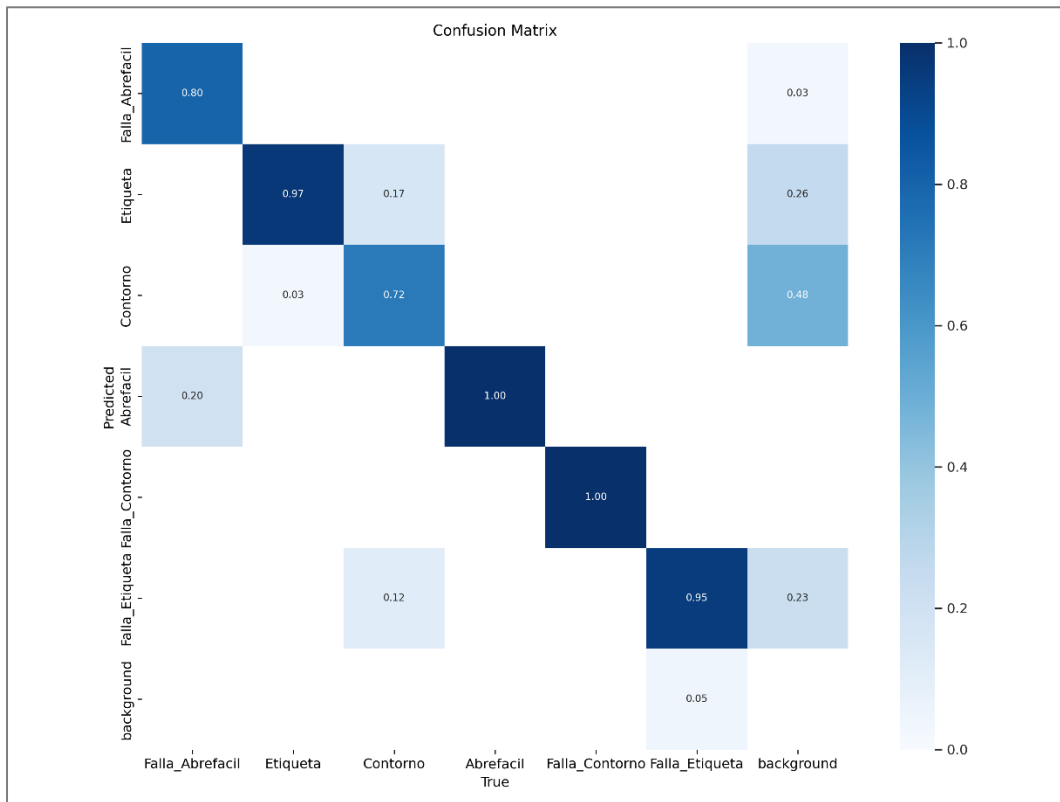


Figura 101: Matriz de confusión

3.20.6. Graficas con curvas de confianza y de precisión

En la siguiente grafica visualizamos en el eje de las *x* los valores de confianza y en el eje de las *y* los datos de precisión del modelo, la línea azul representa el total de todas las clases con un valor de 1 de confianza hasta los 0.985 de precisión. Se observa cómo cambian los valores de precisión de cada etiqueta mientras avanzan las iteraciones (Ver Figura 103).

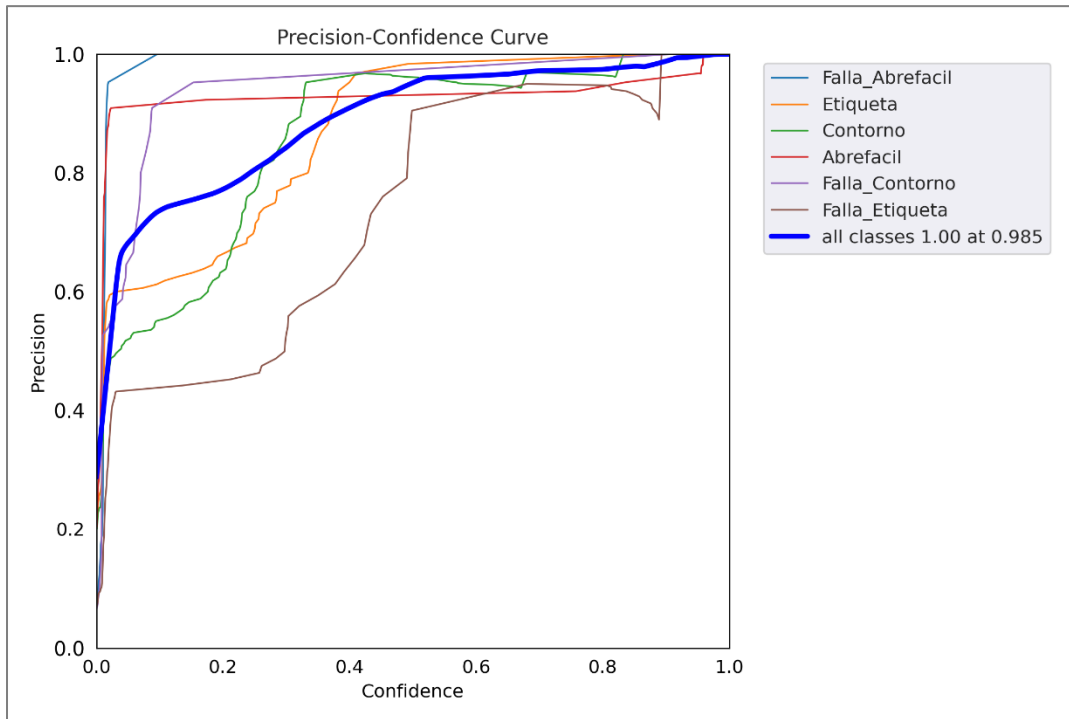


Figura 103 : Graficas de curva de confianza y precisión

3.20.7. Resultados de métricas con entrenamiento de 200 épocas.

En la siguiente tabla se detalla los resultados de un entrenamiento con 200 épocas que determinan las iteraciones o fases del entrenamiento, los valores proporcionados de las métricas de precisión están por encima del 95 % de precisión (Ver Tabla 17).

En el entrenamiento se pueden verificar los valores de las métricas, como metrics/precisión que permite determinar si el modelo que realizó la detección de la lata fue preciso, en la tabla se detallan datos de las métricas, considerando la época 100 hasta la 199, donde se obtiene el valor de 0.95917 como resultado del porcentaje de precisión, es decir realizó una detección perfecta. La métrica mAP (Precisión promedio media) se emplea para escoger los mejores pesos y luego realizar el cálculo en las imágenes de prueba, esta métrica es importante porque

define si el modelo es lo suficientemente robusto cuando se realicen las detecciones en el sistema de visión.

Para la metrics/mAP0.5 se toma como referencia la última iteración dando como resultado 0.979 con un alto porcentaje de precisión y confianza superior a 0.5. En metrics/mAP0.5:0.95 en la época 199 se obtiene un valor de 0.95 de confianza y un valor de 0.76313 de precisión en la detección del objeto. Estos valores mejoran cuando se le agrega más épocas al entrenamiento y también podemos conseguir disminuir las pérdidas.

Tabla 17: Métricas del entrenamiento en Google Colab con 200 épocas

Epoch	Metrics/precisión	Metrics/recall	Metrics/mAP0.5	Metrics/mAP0.5:0.95
190	0.95869	0.93715	0.98006	0.76361
191	0.95992	0.93761	0.98025	0.76555
192	0.96057	0.9319	0.98032	0.76698
193	0.95859	0.93525	0.97954	0.76375
194	0.95499	0.94051	0.97913	0.76153
195	0.95687	0.94146	0.97906	0.76425
196	0.95982	0.93792	0.97891	0.76249
197	0.95965	0.93861	0.97891	0.76291
198	0.95935	0.93749	0.97921	0.76414
199	0.95917	0.94134	0.979	0.76313

3.20.8. Resultados de las inferencias en Google Colab

En la imagen visualizamos los porcentajes de precisión de cada etiqueta en el momento que son detectadas cuando pasan por cada iteración del entrenamiento (Ver Figura 105).

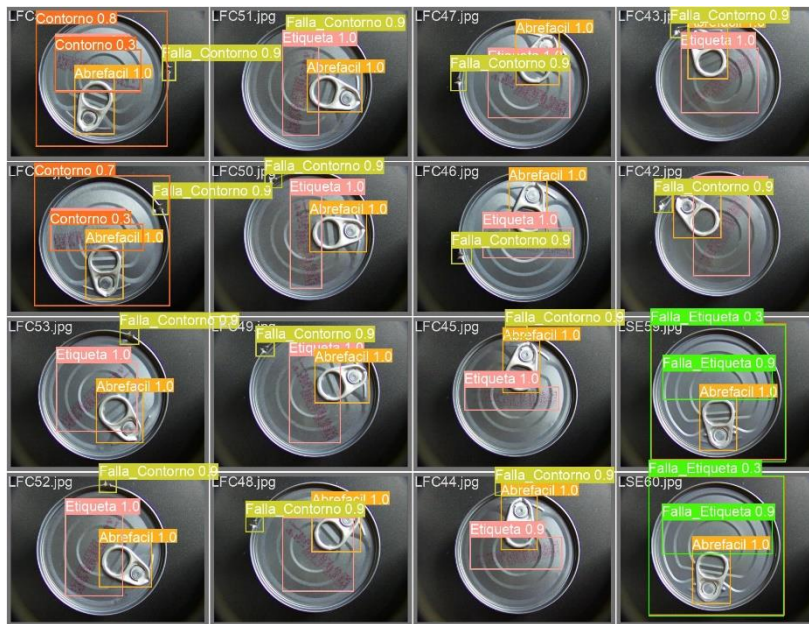


Figura 105 : Inferencias del entrenamiento en Google Colab

3.20.9. Sistema de Clasificación con el Brazo Robótico

Para la programación de las secuencias que realiza el brazo robot se utilizó el software Arduino Ide y para realizar las conexiones, la tarjeta Sensor Shield V5.0 que permite un mejor orden de las conexiones a realizar, el sistema de visión mediante la detección proporciona datos de calidad de cada lata y con el sistema de clasificación del brazo robótico que previamente se programó en Arduino, cuando estamos en esta etapa el brazo recibe la información para realizar el control con los servomotores del robot.

La generación de los movimientos es por medio de los servomotores, para el proyecto utilizamos los servos MG9996R que permiten realizar los movimientos necesarios de la base, antebrazo y brazo del robot y los servos MG90S que realizan los movimientos de muñeca y

mano del brazo robótico (Ver Tabla 18). El sistema de clasificación tiene una tarjeta sensor shield, utiliza una fuente de voltaje de 5V a 5 A que es la recomendada para que el sistema de brazo robótico funcione de manera óptima. En la tabla 18 se detalla los pines de cada servomotor que se utilizó para el sistema de clasificación.

Tabla 18: Pines PWM para el sistema de control del brazo robótico

Servomotor	Pin PWM (Sensor Shield V 5.0)	Descripción
Base	3	Control giro de la base
Antebrazo20	5	Control de posición de arriba-abajo del antebrazo
Antebrazo21	6	Control de posición de arriba-abajo en sentido contrario al antebrazo20
Brazo	9	Control de arriba-abajo del movimiento del brazo
Muñeca	10	Control de movimiento vertical de la muñeca
Mano	11	Control de movimiento horizontal de la mano

3.20.10. Secuencias del funcionamiento del brazo robótico

El control de la posición de cada servomotor se establece por ángulos, estos tienen un giro total de 180°, en el proyecto utilizamos giros de 90° en cada posición. El sistema de clasificación del brazo robótico empieza la secuencia en el centro de su base con una posición inicial establecida en 90° para cada uno de los 6 servomotores, cuando el sensor fotoeléctrico que está ubicado en la parte inferior del brazo robótico detecta una lata, realiza la secuencia dependiendo de la detección que realizó el sistema de visión, si la detección es con fallas en abrefácil, contorno y etiqueta, el brazo robótico succiona la lata y utilizara la secuencia de desplazamiento hacia su extremo derecho depositando el producto en ese lugar (Ver Figura 106).



Figura 106: Desplazamiento hacia la derecha lata con fallas

Cuando el sistema de visión realiza el reconocimiento de una lata con los estándares de calidad como su abrefácil, etiqueta y contorno, entonces al pasar la lata por el sensor fotoeléctrico se activa y empieza la secuencia del brazo robótico realizando una inclinación para succionarla, luego realiza la secuencia de subir para girar 90° a su izquierda, cuando llega a esa posición, baja para dejar la lata, seguido sube para regresar a su posición inicial en el centro de su base establecida de 90° (Ver Figura 107).



Figura 107: Desplazamiento del brazo hacia la izquierda

3.20.11. Análisis de resultados del sistema de control con visión artificial

Después de realizar varios entrenamientos en Google Colab con diferentes épocas, se realiza una comparativa de los resultados para determinar si con 29 épocas o 200 épocas obtenemos resultados más precisos en las detecciones.

En el entrenamiento con 29 épocas se escogió los primeros 10 valores de las métricas de precisión proporcionando datos inferiores al 70%, en la tabla 19 se detalla los resultados de las 10 primeras épocas de las métricas de precisión, recall, mAP0.5 y mAP0.5:0.95

Tabla 19 : Métricas del entrenamiento con 29 épocas

Epoch	Metrics/precisión	Metrics/recall	Metrics/mAP0.5	Metrics/mAP0.5:0.95
0	0.015711	0.73056	0.15081	0.043746
1	0.5929	0.24167	0.15089	0.049589
2	0.74393	0.32573	0.2507	0.097365
3	0.41246	0.37878	0.3384	0.092484
4	0.72012	0.38331	0.42067	0.12896
5	0.49394	0.4	0.27651	0.10554
6	0.4611	0.60833	0.42372	0.19401
7	0.80286	0.42271	0.59497	0.24802
8	0.73813	0.44799	0.52937	0.25504
9	0.26218	0.81586	0.45344	0.18171
10	0.81448	0.54737	0.65227	0.3341

Para el entrenamiento con 200 épocas se escogió desde la época 100 hasta la 110 para realizar el análisis donde se obtuvieron los siguientes resultados. Cuando se utiliza más iteraciones hay un aumento de la precisión en cada detección, hay que considerar la cantidad de imagen a utilizar para que el modelo de entrenamiento sea estable y tener un procesador con bastantes recursos computacionales.

Tabla 20 : Métricas del entrenamiento con 200 épocas

Epoch	Metrics/precisión	Metrics/recall	Metrics/mAP0.5	Metrics/mAP0.5:0.95
100	0.95765	0.9096	0.96082	0.70992
101	0.96316	0.90773	0.96968	0.73094
102	0.97056	0.91132	0.97266	0.72429
103	0.977133	0.92687	0.97808	0.74749
104	0.96711	0.93904	0.98198	0.73971
105	0.96404	0.96944	0.97706	0.75366
106	0.96942	0.97131	0.9789	0.76156
107	0.95296	0.97013	0.97461	0.745
108	0.9309	0.94321	0.96653	0.73554
109	0.95639	0.89282	0.93932	0.72268
110	0.92737	0.92984	0.95314	0.7163

Para el proyecto se utilizó el entrenamiento con 200 épocas por darnos una precisión superior al 92 % en cada una de las detecciones.

Para la implementación de nuevos procesos donde se aplique inteligencia artificial hay que considerar la utilización de un hardware para IA como NVIDIA que está diseñado para aplicaciones de visión artificial como detección, clasificación y segmentación de objetos, tienen un coprocesador como es su GPU que está diseñado principalmente para el procesamiento de imágenes, al tener un GPU potente los entrenamientos tienen un tiempo de minutos para realizar el procesamiento de visión.

Se puede utilizar una Jetson Nano al tener características de emplear varias redes neuronales, también posee un optimizador de inferencias dando como resultado un alto rendimiento y una precisión mas exacta para el aceleramiento de las aplicaciones de aprendizaje automático.

3.20.12. Comunicación del Sistema IoT

Con los datos que se obtiene del sistema de visión mediante los frame que genera la cámara se realiza la clasificación de las fallas que se encuentren en la línea de producción, estos datos serán almacenados en influxdb para luego ser visualizados en el software grafana permitiendo ver estos datos en tiempo real por parte del gerente de la empresa.

Con la programación en Python del sistema de visión se incorpora ciertas líneas de código para trabajar con el bróker MQTT, este se conecta con node red donde se establece la comunicación, a su vez los datos que se proporcione se guardaran en una base de datos en influxdb para enlazarlos mediante grafana por medio de un dashboards para su visualización.

El sistema de monitoreo consta de la adquisición de datos y el procesamiento, la cámara raspberry HQ es la encargada de la visualización de los frame que contiene los datos que necesitamos y el sistema de comunicación se realiza mediante MQTT.

Para el intercambio de datos con node red es necesario instalar ciertas librerías que permitirán realizar la conexión con el código programado en Python. Para tener comunicación se empleará una conexión por medio de ethernet o wifi , donde se configurará las direcciones IP fija para tener una comunicación estable con node red.

3.20.13. Inicialización de Node-Red en la Raspberry pi 4

Para dar inicio a node red ingresamos por consola el siguiente comando para habilitarla.

node-red-start (Ver Figura 108).

```
pi@raspberrypi:~ $ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.1.8:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop           to stop Node-RED
Use node-red-start          to start Node-RED again
Use node-red-log             to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
```

Figura 108: Inicialización de Node-Red en la Raspberry pi 4

En el navegador de la raspberry digitamos la dirección IP 192.168.1.8 seguido del puerto 1880, se abrirá una pantalla donde se solicita usuario y contraseña para permitir el ingreso a node red, en la pantalla principal encontramos los nodos para realizar la programación(Ver Figura 109).

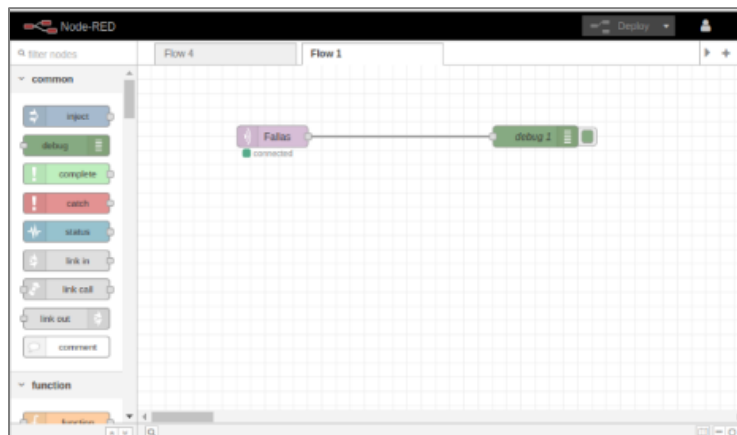


Figura 109: Pantalla de programación nodos de Node-Red

Se detalla la integración de la cámara con el desarrollo del algoritmo del procesamiento de los frame en tiempo real y como estos datos se transmiten a través de MQTT todo esto se realiza en node red

3.20.14. Integración de la cámara raspberry HQ

Para integrar y controlar la cámara empleamos la aplicación de node red, ejecución(exec), con este nodo se puede realizar capturas de cada frame las capturas poseen una resolución de 640 x 380

3.20.15. Transmisión de frame en tiempo real

Los frames del video en tiempo real se realiza por medio del protocolo MQTT , se debe considerar que el bróker admite 128 kb . El algoritmo cuando recibe una señal de la cámara, realiza la respectiva captura del frame para ser enviado a MQTT y después enviar a node red se debe emplear un valor bajo en fotogramas por segundo esto se realiza para no colapsar el envío de los mensajes MQTT.

3.20.16. Inicialización de Mosquitto en la Raspberry pi 4

Con el comando status en la consola de la raspberry se puede verificar si se encuentra activo el protocolo mosquitto, digitamos el siguiente comando (Ver Figura 110).

```
sudo systemctl status mosquitto
```

Si el protocolo mosquitto no se encuentra activo digitamos la siguiente línea de comando.

```
sudo systemctl start mosquitto
```

```
pi@raspberrypi:~$ sudo systemctl status mosquitto
mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor pre
   Active: active (running) since Thu 2023-11-23 15:29:41 -05; 23min ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
    Process: 566 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exi
    Process: 566 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exi
    Process: 569 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, p
    Process: 570 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, p
   Main PID: 571 (mosquitto)
      Tasks: 1 (limit: 3723)
         CPU: 836ms
   CGroup: /system.slice/mosquitto.service
           └─571 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

Nov 23 15:29:40 raspberrypi systemd[1]: Starting Mosquitto MQTT Broker...
Nov 23 15:29:40 raspberrypi mosquitto[571]: 1706771380: Loading config file /et
Nov 23 15:29:41 raspberrypi systemd[1]: Started Mosquitto MQTT Broker.
lines 1-18/18 (END)
```

Figura 110: Comando inicialización de mosquitto

3.20.17. Integración de Node-Red con Mosquitto

Configuramos el node de entrada de mqtt que se encuentra en node red, procedemos a colocar en el entorno donde al darle doble clic procedemos a realizar la configuración del bróker.

En connection, digitamos en Server la IP o dejamos como localhost y en port 1883(Ver Figura 111). En security colocamos el username que es mosquitto y el password, damos clic en Add para agregar los cambios realizados.

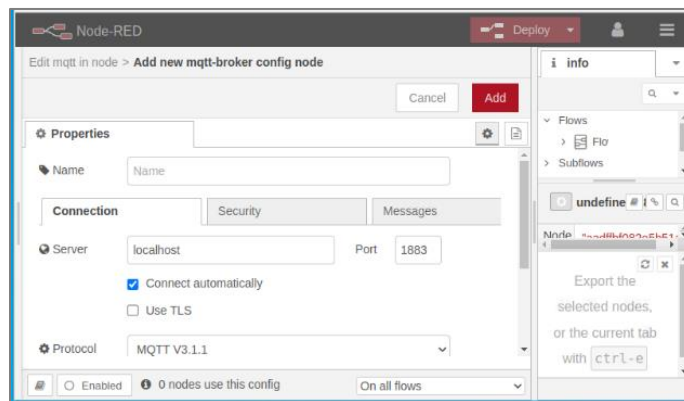


Figura 111: Configuración node red -mosquitto

Configuramos el Server como Mosquitto y definimos el Topic como calidad y damos clic en Done para configurar el servidor, este nodo se suscribe al t3pico calidad, si se quiere ver los mensajes se agrega un node debug, luego procedemos a dar clic en Deploy, se puede visualizar que la comunicaci3n se realiz3 de manera satisfactoria (Ver Figura 112).

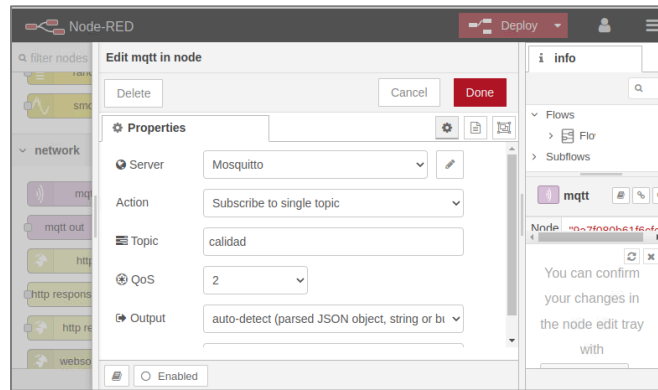


Figura 112: Configuraci3n del Server Mosquitto

3.20.1. Comunicaci3n remota con mosquitto

Ingresamos al sitio web de mqtt-explorer.com para descargar el software para instalar en la laptop, con esto se puede realizar la comunicaci3n cliente mqtt con el br3ker mosquitto, en el entorno creamos una nueva conexi3n, digitamos el nombre de la comunicaci3n y escribimos en el host la IP de la raspberry pi y le damos clic en **connect** para realizar la conexi3n de mqtt (Ver Figura 113). Configuraci3n de conexi3n remota con MQTT

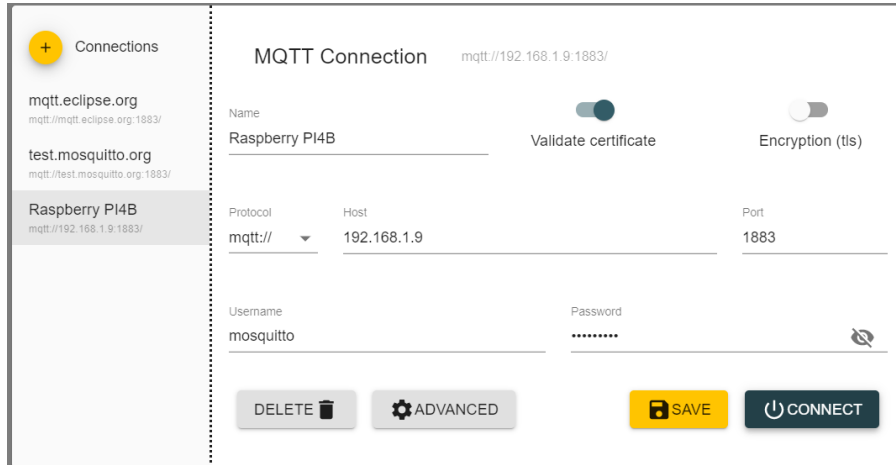


Figura 113: Comunicación remota de Mqtt

3.20.2. Inicialización de InfluxDB en la Raspberry pi 4

Para habilitar el servicio de influxDB digitamos en la consola los siguientes comandos.

```
sudo systemctl enable influxdb
sudo systemctl start influxdb
```

3.20.3. Creación de Database InfluxDB

Cuando tenemos autenticado influxdb, realizamos la creación de la base de datos que utilizaremos para el proyecto, primero debemos de ingresar los datos de autenticación con el siguiente comando (Ver Figura 114).

```
Influx -username admin -password
```

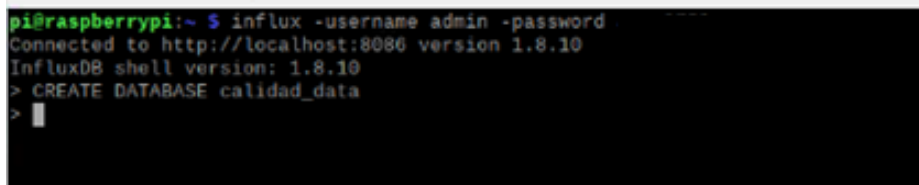


Figura 114: Comando creación de data base influxdb

Para verificar la base de datos que contiene influxdb digitamos en la consola el comando (Ver Figura 115).

```
>Show databases
>name: databases
>use calidad_data
```

```
pi@raspberrypi:~$ influx -username admin -password anvri721
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> show databases
name: databases
name
----
internal
iot
IOT
CALIDAD
calidad_data
> use calidad_data
Using database calidad_data
> INSERT calidad,lata1=1,unidad=F,Abrefacil=1,Contorno=0,Etiqueta=0
> select lata1,Abrefacil from calidad
name: calidad
time                lata1 Abrefacil
----                -
1760788816314329744 1      1
```

Figura 115: Comando verificación de base de datos influxdb

3.20.4. Creación del dashboards en Grafana

Para ingresar a grafana digitamos en el url la ip 192.168.1.8:3000 luego colocamos el usuario y contraseña, estando dentro de grafana vamos a crear el databoards, damos clic en data sources y en add data source seleccionamos InfluxDB para crear una nueva configuración. A continuación, digitamos los siguientes datos para configurar el Grafana ver tabla 21 .

Tabla 21 : Configuración de Grafana

Datos	Descripción
Name	Control-Calidad
Query Languaje	InfluxQL
URL	http://192.168.1.8:8086
Basic auth	Activamos la casilla
Basic Auth Details	Colocamos el usuario y contraseña de influxDB

database	Control-Calidad
Sabe & test	Le damos un clic para guardar configuración

Damos clic en Build a Dashboards y se abre una página donde seleccionamos add visualization para seleccionar la data que creamos para este caso sería Control-Calidad y creamos nuestro panel (Ver Figura 116).

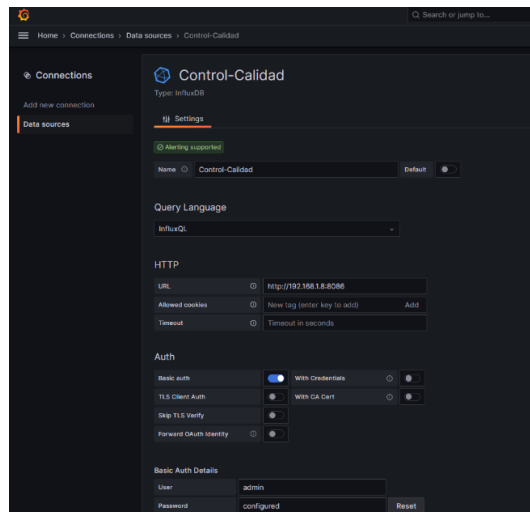


Figura 116: Ingreso de datos al data sources

3.21. ESTUDIO DE FACTIBILIDAD

3.21.1. Factibilidad Técnica

Los dispositivos empleados como tarjeta raspberry pi 4, cámara, sensores fotoeléctricos, motor y brazo robótico, en el proyecto nos permiten cumplir con los objetivos para el funcionamiento del sistema automatizado de control de calidad de las latas de atún.

La programación en la tarjeta raspberry pi 4 permite un análisis mediante visión artificial de las fallas que pueden presentarse en el sellado de latas. En la actualidad existen empresas que

cuentan con la industria 4.0, debido a esto el proyecto está orientado a este tipo de industria por lo que se utiliza el software InfluxDB y Grafana, estos permiten obtener datos del proceso en tiempo real, donde podemos acceder desde cualquier parte del mundo. Con los equipos instalados y la programación se pueden realizar pruebas para verificar que las latas tengan un sello hermético y de esta manera demostramos su funcionamiento.

3.22. ANALISIS DE COSTOS DEL PROYECTO.

El proyecto se basa en la implementación de un prototipo automatizado para el control de calidad en las latas de atún, donde se establecen los costos de cada dispositivo eléctrico y electrónico.

3.22.1. COSTOS DE EQUIPOS

Se detalla los materiales y costos de equipos que se empleara para la elaboración del sistema automatizado para el control de calidad de latas de atún mediante visión artificial (Ver Tabla 22).

Tabla 22 *Costo de Equipos*

Cantidad	Detalle	Valor Unitario \$	Valor Total \$
1	Kit Raspberry PI4B + 4GB Ram 32GB Msd	\$266.00	\$266.00
1	Modulo Camara Raspberry Pi High Quality Camera	\$74.68	\$74.68
1	Arducan Lente fija 6MM montura CS	\$53.49	\$53.49
1	Robot Manipulador ARM-N2	\$273.92	\$273.92
15	Cable extensor servomotor 3 pines 30cm	\$0.75	\$11.25
1	Fuente de alimentacion	\$20.00	\$20.00
1	Prototipo de la banda transportadora	\$500.00	\$500.00
1	Caja de control	\$45.00	\$45.00
10	Metros cable eléctrico flexible 12 AWG	\$0.91	\$9.10
10	Metros cable eléctrico flexible 16 AWG	\$0.54	\$5.40
16	Metros cable UTP categoría 6	\$0.75	\$12.00
2	Sensor Fotoeléctrico	\$28.00	\$56.00
1	Breaker Riel 2 polos 10 A Chint	\$10.00	\$10.00
1	Breaker Riel 2 polos 16 A Camsco	\$7.00	\$7.00
1	Materiales adicionales	\$200.00	\$200.00
TOTAL			\$1.543,84

CAPITULO IV

4.1. CONCLUSIONES.

- En el desarrollo de esta tesis se empleó el modelo de YoloV5 para la detección de fallas en los bordes de la tapa de latas de atun, mediante la preparación de un conjunto de datos y el entrenamiento se realizó la identificación con un rectángulo las imágenes para localizar las fallas y con las convoluciones que realiza este algoritmo se llegó a obtener los resultados esperados de detección.
- Se requiere emplear un hardware muy potente que permita realizar la adquisición de los datos de etiqueta de las latas para el reconocimiento de texto OCR, se realizó un entrenamiento en Google Colab que brinda una GPU para realizar las iteraciones a gran velocidad de procesamiento, esto genero la posibilidad de reconocimiento del texto y extraer los datos que contiene la etiqueta aplicando de esta forma las detecciones de texto OCR.
- Para la detección del posicionamiento de la anilla abrefácil en tiempo real se empleó el modelo de YOLOV5x, porque contiene un modelo comprimido que permiten aumentar la velocidad cuando se realicen las detecciones de las fallas según las normas establecidas de productos enlatados, mediante el entrenamiento en Google Colab se determinó mediante los bounding box las coordenadas de ubicación del posicionamiento de la anilla abrefácil.
- Mediante el analisis de resultados obtenidos, con la implementación de un Sistema IOT permite conocer el estado actual en que se encuentra la producción, donde se

pueden presentar problemas y corregirlos de forma oportuna tomando acciones para de esta forma optimizar recursos y tener productos de buena calidad para su venta y distribución. Emplear un sistema de visión en el control de calidad cuenta con muchas ventajas, donde el sistema mediante tareas determinadas puede ejecutarlas sin presencia de un operario y que esta persona pueda realizar otras tareas dentro de la empresa.

4.2. RECOMENDACIONES.

- En el prototipo se pueden implementar equipos de tipo industrial, como cámaras industriales que generen un incremento de tolerancia de las fallas del sistema para que sus resultados sean más precisos.
- Se recomienda emplear un dataset con una gran cantidad de imágenes para el desarrollo del sistema de visión que detectara las fallas de calidad en los productos debido a que el sistema debe aprender mejor esos defectos.
- Procurar guardar el entrenamiento realizado en Google Colab, importando esos datos en un archivo zip, para cuando se requiera utilizar de nuevo el entrenamiento porque el software los mantiene de forma temporal.
- Es recomendable utilizar un procesador Nvidia que tienen una capacidad de procesamiento para trabajar con inteligencia artificial como clasificación, segmentación y detección de objetos. Estas al tener más recursos computacionales permiten realizar las inferencias de manera óptima.

Anexos

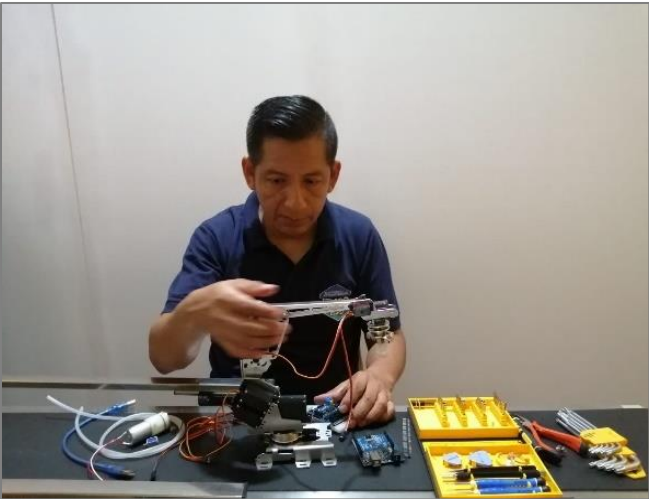
Anexo A: Primera parte de programación en Arduino del sistema de clasificación del brazo robótico.

```
CONTROL_BRAZO_ROBOTICO_ARM_21N2.ino
1  //****Objetivo: Realizar el control de los servomotores del brazo robotico ****.
2  #define BAS1 3 //BASE
3  #define ANTEB20 5 //ANTEBRAZO 20
4  #define ANTEB21 6 //ANTEBRAZO 21
5  #define BRAZ3 9 //BRAZO
6  #define MRV5 10 //MANO ROTACION VERTICAL
7  #define MRH6 11 //MANO ROTACION HORIZONTAL
8  #define Bomba 12 //Ventosa succionador
9  #define Libvent 13 //Valvula liberacion de ventosa
10
11 #include <Servo.h> // incluimos la biblioteca para accionar los servomotores.
12 Servo base; // Servomotor rotación de Base.
13 Servo hombro20; // Servomotor acoplado a ANTEB21 desde el hombro. Rotación del antebrazo.
14 Servo hombro21; // Servomotor acoplado a ANTEB20 desde el hombro. Rotación del antebrazo.
15 Servo brazo; // Servomotor de codo. Rotación de brazo.
16 Servo manov; // Servo de muñeca. Rotación vertical de la muñeca.
17 Servo manoh; // Servo de mano. Rotación horizontal de la mano.
18
19 void setup()
20 {
21     base.attach(BAS1); // El servomotor BAS1 se acciona por el puerto 3.
22     hombro20.attach(ANTEB20); // El servomotor ANTEB20 se acciona por el puerto 5.
23     hombro21.attach(ANTEB21); // El servomotor ANTEB21 se acciona por el puerto 6.
24     brazo.attach(BRAZ3); // El servomotor BRAZ3 se acciona por el puerto 9.
25     manov.attach(MRV5); // El servomotor MRV5 se acciona por el puerto 10.
26     manoh.attach(MRH6); // El servomotor MRH6 se acciona por el puerto 11.
27     //Definir pin de valvula y succionador como salida
28     pinMode(Bomba,OUTPUT);
29     pinMode(Libvent,OUTPUT);
30     //Apagar valvula y succionador para evitar interferencias
31     digitalWrite(Libvent,LOW);
32     digitalWrite(Bomba,LOW);
33 }
```

Anexo B : Primera parte de programación en Python del sistema visión artificial.

```
Tesis_object_d...ct_tuna_cans.py x
1 # Importamos librerias
2 import torch
3 import cv2
4 from picamera2 import Picamera2
5 import numpy as np
6
7 # Leemos el modelo
8 model = torch.hub.load('ultralytics/yolov5', 'custom',
9 path = '/home/pi/TESIS ALGORITMOS/TESIS_OBJECT_DETECTION_YOLOV5/model
10 dispW=640
11 dispH=480
12
13 picam2=Picamera2()
14 picam2.preview_configuration.main.size=(dispW, dispH)
15 picam2.preview_configuration.main.format='RGB888'
16 picam2.preview_configuration.align()
17 picam2.configure("preview")
18 picam2.start()
19
20 # Realizo Videocaptura
21 unicam='/dev/video0'
22 cap=cv2.VideoCapture(unicam)
23 cap.set(cv2.CAP_PROP_FRAME_WIDTH, dispW)
24 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, dispH)
25 cap.set(cv2.CAP_PROP_FPS, 20)
```


Anexo C: Ensamble y diseño final del sistema de control con visión artificial



Bibliografía

[Lideres, «Ecuador es potencia atunera,» Abril 2018. [En línea]. Available:
1 <https://www.revistalideres.ec/lideres/ecuador-industria-procesa-atun-exportacion.html>.
]

[H. Morell, «La inteligencia artificial como complemento a nuestra inteligencia y motor de
2 la innovacion,» Agosto 2021. [En línea]. Available:
] <https://www.universia.net/ec/actualidad/orientacion-academica/la-inteligencia-artificial-como-complemento-a-nuestra-inteligencia-y-motor-de-la-innovacion.html>.

[Codex Alimentarius, «Norma para el atun y el bonito en conserva cxs70-1981,» 2018. [En
3 línea]. Available: [https://www.fao.org/fao-who-codexalimentarius/sh-](https://www.fao.org/fao-who-codexalimentarius/sh-proxy/en/?lnk=1&url=https%253A%252F%252Fworkspace.fao.org%252Fsites%252Fcodex%252Fstandards%252FCXS%2B70-1981%252FCXS_070s.pdf)
] [proxy/en/?lnk=1&url=https%253A%252F%252Fworkspace.fao.org%252Fsites%252Fcodex%252Fstandards%252FCXS%2B70-1981%252FCXS_070s.pdf](https://www.fao.org/fao-who-codexalimentarius/sh-proxy/en/?lnk=1&url=https%253A%252F%252Fworkspace.fao.org%252Fsites%252Fcodex%252Fstandards%252FCXS%2B70-1981%252FCXS_070s.pdf).

[Ministerio de comercio exterior, «Informe sobre el sector atunero ecuatoriano,» Agosto
4 2017. [En línea]. Available: [https://www.produccion.gob.ec/wp-](https://www.produccion.gob.ec/wp-content/uploads/2019/06/Reporte-del-sector-atunero.pdf)
] [content/uploads/2019/06/Reporte-del-sector-atunero.pdf](https://www.produccion.gob.ec/wp-content/uploads/2019/06/Reporte-del-sector-atunero.pdf).

[N. Aniorte, «Diseño de la investigacion,» Septiembre 2022. [En línea]. Available:
5 http://www.aniorte-nic.net/apunt_metod_investigac4_4.htm.
]

[M. Cortez y M. Iglesias, «Generalidades sobre metodologia de la investigacion,» 2004.
6 [En línea]. Available:
] https://www.unacar.mx/contenido/gaceta/ediciones/metodologia_investigacion.pdf.

[D. Llopis Castello, «Metodologia experimental,» [En línea]. Available:
7 <https://poliformat.upv.es/access/content/user/24389381/Contenido%20abierto%20al%20>
] [p%C3%BAblico/Metodolog%C3%ADa%20de%20la%20investigaci%C3%B3n/3.2%20Metodologi%CC%81a%20experimental.pdf](https://poliformat.upv.es/access/content/user/24389381/Contenido%20abierto%20al%20p%C3%BAblico/Metodolog%C3%ADa%20de%20la%20investigaci%C3%B3n/3.2%20Metodologi%C3%A1a%20experimental.pdf).

[Revista Lideres, «Atun: el producto clave en la canasta exportable de Ecuador y el hogar,»
8 Abril 2021. [En línea]. Available: [https://www.revistalideres.ec/lideres/atun-clave-](https://www.revistalideres.ec/lideres/atun-clave-canasta-exportacion-ecuador.html)
] [canasta-exportacion-ecuador.html](https://www.revistalideres.ec/lideres/atun-clave-canasta-exportacion-ecuador.html).

[C. Galindo Leal, «Katsuwonus pelamis,» Septiembre 2022. [En línea]. Available:
9 <https://www.naturalista.mx/taxa/56943-Katsuwonus-pelamis>.
]

[Mundo Latas, «Como debe ser un buen envase,» [En línea]. Available:
1 <https://mundolatas.com/como-debe-ser-un-buen-envase/>.
0
]

[M. Bustamante, «Mejora del proceso de conserva de esparrago verde en la empresa Alpes
1 Chiclayo S.A.C para aumentar la productividad,» 2019. [En línea]. Available:
1 https://tesis.usat.edu.pe/bitstream/20.500.12423/2505/1/TL_BustamanteVillegasMariane
] [la.pdf](https://tesis.usat.edu.pe/bitstream/20.500.12423/2505/1/TL_BustamanteVillegasMariane).

[IEEC, «Funciones basicas, características y arquitectura de los sistemas automatizados,»
1 [En línea]. Available:
2 http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia
] [_ISE2_1_1.pdf](http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/Informacion_de_referencia).

[Assembly, «Beneficios y tipos de sistemas de transporte,» Octubre 2016. [En línea].
1
3 Available: <https://www.assembly.es/beneficios-tipos-sistemas-transporte/>.
]

[Proteval, «Cintas transportadoras, que tipos hay y sus ventajas,» [En línea]. Available:
1 <https://proteval.com/cintas-transportadoras-usos-ventajas-y-tipos/>.

4

]

[R. Leira y J. Gomez, «Electronica Analogica,» [En línea]. Available:
1 [https://www.edu.xunta.gal/centros/iesblancoamorculledo/aulavirtual/pluginfile.php/37](https://www.edu.xunta.gal/centros/iesblancoamorculledo/aulavirtual/pluginfile.php/375095/mod_resource/content/2/analo%CC%81gica.pdf)
5 [095/mod_resource/content/2/analo%CC%81gica.pdf](https://www.edu.xunta.gal/centros/iesblancoamorculledo/aulavirtual/pluginfile.php/375095/mod_resource/content/2/analo%CC%81gica.pdf).

]

[Geya, «Diferentes tipos de sensores de proximidad y sus aplicaciones,» 6 Marzo 2023.
1 [En línea]. Available: <https://www.geya.net/es/different-types-of-proximity-sensors/>.

6

]

[M. Gandhi, «Que es un sistema de control?,» Noviembre 2022. [En línea]. Available:
1 <https://www.autycom.com/que-es-un-sistema-de-control/>.

7

]

[Atri Innovation, «La vision artificial,» Abril 2020. [En línea]. Available:
1 <https://www.atriainnovation.com/vision-artificial-ventajas-aplicaciones/>.

8

]

[Cardozo, Mendez, L. Ayala y V. Noguera, «Mejora de imagenes a color utilizando un
1 enfoque morfologico multiescalar,» Octubre 2018. [En línea]. Available:
9 [http://sedici.unlp.edu.ar/bitstream/handle/10915/73181/Documento_completo.pdf-](http://sedici.unlp.edu.ar/bitstream/handle/10915/73181/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
] [PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/73181/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y).

[R. Alvarado, «Operadores morfologicos sobre reticulados completos para imagenes a
2 escala de grises,» 2016. [En línea]. Available:
0 <https://dspace.unitru.edu.pe/handle/UNITRU/8481>.

]

[Savedra, «Deteccion de borde en tiempo real empleando el filtro de sobel y tecnologia
2 reconfigurable,» Octubre 2014. [En línea]. Available:
1 <http://itcelaya.edu.mx/ojs/index.php/pistas/article/view/1470>.

]

[J. Robles Bokun, «Aporte en las metodologias matematicas para mejorar la comprension
2 y calidad de video,» Agosto 2018. [En línea]. Available:
2 http://revistas.urp.edu.pe/index.php/Perfiles_Ingenieria/article/view/2372/2417.

]

[M. Jimenez, «Operacionjes basicas con imagenes a escala de grises,» Agosto 2018. [En
2 línea]. Available: [http://ri.uaemex.mx/bitstream/handle/20.500.11799/103697/secme-
3 37094_1.pdf?sequence=1](http://ri.uaemex.mx/bitstream/handle/20.500.11799/103697/secme-337094_1.pdf?sequence=1).

]

[A. Canteli, «Inteligencia artificial y reconocimiento optico de caracteres,» 4 Abril 2023.
2 [En línea]. Available: [https://www.openkm.com/es/blog/inteligencia-artificial-y-
4 reconocimiento-optico-de-caracteres.html](https://www.openkm.com/es/blog/inteligencia-artificial-y-4-reconocimiento-optico-de-caracteres.html).

]

[A. Mitaritonna, «Deep learning: tipos de aprendizajes,» 31 Enero 2020. [En línea].
2 Available: [https://www.linkedin.com/pulse/deep-learning-tipos-de-aprendizajes-
5 alejandro-mitaritonna/?originalSubdomain=es](https://www.linkedin.com/pulse/deep-learning-tipos-de-aprendizajes-5-alejandro-mitaritonna/?originalSubdomain=es).

]

[Unir, «Deep Learning: e en que consiste, ejemplos y aplicaciones,» 10 Junio 2021. [En
2 línea]. Available: <https://www.unir.net/ingenieria/revista/deep-learning/>.

6
]

[Pypi, «Ultralytics Yolov5,» 8 Junio 2022. [En línea]. Available:
2 <https://pypi.org/project/ultralytics-yolov5/>.

7
]

[P. Garcia, «Sistema de seguimiento de personas desde un dron,» 19 Julio 2022. [En línea].
2 Available: <http://rua.ua.es/dspace/handle/10045/125320>.

8
]

[P. I. Orellana Rueda, «Segmentacion semantica y reconocimiento de lugares usando
2 cararacteristicas CNN pre-entrenadas,» 2019. [En línea]. Available:
9 [https://repositorio.uchile.cl/bitstream/handle/2250/173733/cf-
\] orellana_pr.pdf?sequence=1](https://repositorio.uchile.cl/bitstream/handle/2250/173733/cf-orellana_pr.pdf?sequence=1).

[W. Icaza Paredes, «Desarrollon de sistema de vision artificial con labview,que permita
3 detectar imperfecciones en las latas de atun,» Octubre 2019. [En línea]. Available:
0 [https://repositorio.unemi.edu.ec/xmlui/bitstream/handle/123456789/4921/1.-](https://repositorio.unemi.edu.ec/xmlui/bitstream/handle/123456789/4921/1.-%20DESARROLLO%20DE%20SISTEMA%20DE%20VISION%20ARTIFICIAL%20CON%20LABVIEW%2c%20QUE%20PERMITA%20DETECTAR%20IMPERFECCIONES%20EN%20LAS%20LATAS%20DE%20AT%c3%9aN.pdf?sequence=1&isAllowed=y)

] %20DESARROLLO%20DE%20SISTEMA%20DE%20VISION%20ARTIFICIAL%20
CON%20LABVIEW%2c%20QUE%20PERMITA%20DETECTAR%20IMPERFECCI
ONES%20EN%20LAS%20LATAS%20DE%20AT%c3%9aN.pdf?sequence=1&isAllo
wed=y.

[W. S. Palma Santacruz, «Implementacion de un sistema de vision artificial en lineas de
3 produccion de atun, para la deteccion de fallas de codificado en la empresa PUERTOMAR
1 S.A Provincia de Manabi,» Septiembre 2020. [En línea]. Available:
] <http://dspace.esPOCH.edu.ec/bitstream/123456789/14101/1/20T01339.pdf>.

[J. Tierra y M. Galarza, «Implementación de un sistema de control y monitoreo en base al
3 procesamiento de imágenes digitales en los sistemas de visión artificial aplicado al

2 reconocimiento de la maquina selectora de botellas en el laboratorio de la Escuela de
] Ingeniería Industria,» 2017. [En línea]. Available:
http://dspace.esPOCH.edu.ec/bitstream/123456789/9346/1/85T00483.pdf.

[N. Branque, «Implementación de un sistema automatizado, mediante el uso de visión
3 artificial para la clasificación del maracuyá, según su color de madurez y el uso de un
3 sistema scada para el monitoreo de la productividad,» 2022. [En línea]. Available:
] <https://repositorio.upse.edu.ec/xmlui/bitstream/handle/46000/8467/UPSE-TEA-2022-0005.pdf?sequence=1&isAllowed=y>.

[Raspberry, «Camara raspberry pi de alta calidad,» [En línea]. Available:
3 <https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/>. [Último
4 acceso: 2022].
]

[Vevor, «Cinta transportadora vevor de transporte industrial,» 2023. [En línea]. Available:
3 https://www.vevor.com/belt-conveyor-c_10439/belt-conveyor-pvc-conveyor-belt59x-515-7-inch-motorized-conveyor-w-guardrails-p_010476271198?lang=es.
]

[Vevor, «Vevor maquina transportadora electrica,» 2023. [En línea]. Available:
3 https://www.vevor.es/cinta-transportadora-c_10439/cinta-transportadora-150x19-8cm-6-tabla-transportadora-0-25-m-min-doble-guardrail-p_010464727863.
]

[Indiamart, «Ac geared motor speed controller 180W,» 2023. [En línea]. Available:
3 <https://pdf.indiamart.com/impdf/18974023730/MY-4133501/us-52-180w-90w-120w-760w-speed-controller.pdf>.
]

[H. Store, «AliExpress,» Bomba de succion de carga,brazo robotico 6dof para arduino,
2022. [En línea]. Available:
3 <https://es.aliexpress.com/item/1005002425906299.html?spm=a2g0o.detail.1000014.13.5>

8 5ee6df1H0q3Vf&gps-
id=pcDetailBottomMoreOtherSeller&scm=1007.40050.281175.0&scm_id=1007.40050.
] 281175.0&scm-url=1007.40050.281175.0&pvid=7944772f-e5c4-40a4-929c-
4970a2f90407&_.

[Naylampmechatronics, «Servo mg996R 11kg - tower pro original,» 2023. [En línea].
3 Available: [https://naylampmechatronics.com/servomotores/343-servo-mg996r-
9 11kg.html](https://naylampmechatronics.com/servomotores/343-servo-mg996r-9-11kg.html).
]

[Amazon, «Bomba de aire micro-dc 12V,» 2023. [En línea]. Available:
4 [https://www.amazon.com/-/es/Bomba-aire-micro-instrumento-
0 tratamiento/dp/B07FGFPKNS](https://www.amazon.com/-/es/Bomba-aire-micro-instrumento-0-tratamiento/dp/B07FGFPKNS).
]

[Transelec, «Soporte-Transelec,» Tableros electricos, sus requerimientos y clasificacion,
4 [En línea]. Available: [https://www.transelec.com.ar/soporte/18408/tableros-el-ctricos-
1 sus-requerimientos-y-clasificaci-n/](https://www.transelec.com.ar/soporte/18408/tableros-el-ctricos-1-sus-requerimientos-y-clasificaci-n/).
]

[Naylamp mechatronics, «Sensor de proximidad fotoelectrico infrarrojo e18-d80nk,» [En
4 línea]. Available: [https://naylampmechatronics.com/sensores-proximidad/236-sensor-de-
2 proximidad-fotoelectrico-infrarrojo-e18-d80nk.html](https://naylampmechatronics.com/sensores-proximidad/236-sensor-de-2-proximidad-fotoelectrico-infrarrojo-e18-d80nk.html).
]

[Arqhys, «Conductor electrico,» 2022. [En línea]. Available:
4 <https://www.arqhys.com/construccion/conductor-electrico.html>.
3
]

[S. A. Castaño Giraldo, «Relevador con arduino,» [En línea]. Available:
4 <https://controlautomaticoeducacion.com/arduino/relevador-con-arduino/>.

4
]

[p. Componentes, «Mundo PcComponentes,» Características de la nueva raspberry pi 4 model B, [En línea]. Available: <https://www.pccomponentes.com/caracteristicas-5-raspberry-pi-4>.
]

[N. Mechatronics, Sensor Shield V5, 2021. [En línea]. Available: <https://naylampmechatronics.com/ardusystem-shields/109-sensor-shield-v5.html>.
6
]

[Arduino, «Store Arduino,» 2021. [En línea]. Available: <https://store.arduino.cc/products/arduino-uno-rev3>.
7
]

[U. Electronics, Fuente de alimentacion 5V, 2023. [En línea]. Available: <https://uelectronics.com/producto/fuente-de-alimentacion-5v-10a/>.
8
]

[Transelec, «Funcionamiento de un interruptor termomagnetico,» [En línea]. Available: <https://www.transelec.com.ar/soporte/18632/funcionamiento-de-un-interruptor-termomagnetico/>.
9
]

[Raspberry, «Install raspberry pi os using raspberry pi imager,» [En línea]. Available: <https://www.raspberrypi.com/software/>.
5
0
]

[Wikiversity, «Python concepts,» 2022. [En línea]. Available:
5 https://en.wikiversity.org/wiki/Python_Concepts.

1

]

[Opencv, «Open source computer vision library,» 2021. [En línea]. Available:
5 https://docs.opencv.org/4.5.3/d6/d00/tutorial_py_root.html.

2

]

[Tensorflow, «Conceptos basicos de tensorflow,» 2022. [En línea]. Available:
5 <https://www.tensorflow.org/?hl=es-419>.

3

]

[Pandas, «Pandas potente kit de herramientas de analisis de datos de python,» 20
5 Septiembre 2023. [En línea]. Available: <https://pypi.org/project/pandas/>.

4

]

[Ultralytics, «Ultralytics yolov8 docs,» 27 Septiembre 2023. [En línea]. Available:
5 <https://docs.ultralytics.com/>.

5

]

[PickData by Circutor, «Node-red, la herramienta de programacion visual para el internet
5 of things,» Mayo 2020. [En línea]. Available: [https://www.pickdata.net/es/noticias/node-6](https://www.pickdata.net/es/noticias/node-6-red-programacion-visual-iot)
6 red-programacion-visual-iot.

]

[Mosquitto, «Eclipse mosquitto an open source mqtt broker,» [En línea]. Available:
5 <https://mosquitto.org/>.

7
]

[Ionos, «Infludb,» 12 Mayo 2023. [En línea]. Available:
5 <https://www.ionos.com/digitalguide/hosting/technical-matters/what-is-influxdb/>.

8
]

[Redhat, «¿Que es grafana?,» 13 Mayo 2022. [En línea]. Available:
5 <https://www.redhat.com/es/topics/data-services/what-is-grafana>.

9
]