



**UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD INGENIERÍA  
CARRERA TELECOMUNICACIONES**

**SISTEMA DE CLASIFICACIÓN DE BOTELLAS MEDIANTE  
EL RECONOCIMIENTO DE IMÁGENES PARA EL  
TRATAMIENTO DE BASURA.**

**Trabajo de Titulación para optar al título de Ingeniero en  
telecomunicaciones**

**Autor:**

Loaiza Alejandro, Bryan José

**Tutor:**

PhD. Leonardo Fabián Rentería Bustamante

**Riobamba, Ecuador. 2023**

## DECLARATORIA DE AUTORÍA

Yo, Bryan José Loaiza Alejandro, con cédula de ciudadanía 070699534-7, autor del trabajo de investigación titulado: **SISTEMA DE CLASIFICACIÓN DE BOTELLAS MEDIANTE EL RECONOCIMIENTO DE IMÁGENES PARA EL TRATAMIENTO DE BASURA**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, a los 12 días del mes de diciembre del 2023.



---

Bryan José Loaiza Alejandro  
C.I: 070699534-7



Dirección  
Académica  
VICERRECTORADO ACADÉMICO

*en movimiento*



UNACH-RGF-01-04-02.19  
VERSIÓN 02: 06-09-2021

## ACTA FAVORABLE - INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN

En la Ciudad de Riobamba, a los 01 días del mes de noviembre del 2023, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **Bryan José Loaiza Alejandro** con CC: **0706995347**, de la carrera **Ingeniería en Telecomunicaciones** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado "**Sistema de clasificación de botellas mediante el reconocimiento de imágenes para el tratamiento de basura**", por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



Firmado digitalmente por:  
LEONARDO FABIAN  
RENERIA BUSTAMANTE

\_\_\_\_\_  
PhD. Leonardo Fabian Rentería Bustamante.

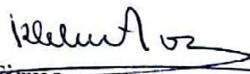
**TUTOR**

## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación **“SISTEMA DE CLASIFICACIÓN DE BOTELLAS MEDIANTE EL RECONOCIMIENTO DE IMÁGENES PARA EL TRATAMIENTO DE BASURA”**, presentado por **BRYAN JOSÉ LOAIZA ALEJANDRO** con cédula de identidad **0706995347**, bajo la tutoría del **PhD. LEONARDO FABIÁN RENTERÍA BUSTAMANTE**; certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba a los 12 días de diciembre de 2023.

Presidente del Tribunal de Grado  
Mgs. Klever Hernán Torres Rodríguez

  
Firma

Miembro del Tribunal de Grado  
PhD. Marlon Danilo Basantes Valverde

  
Firma

Miembro del Tribunal de Grado  
Mgs. José Luis Jínez tapia

  
Firma



Dirección  
Académica  
VICERRECTORADO ACADÉMICO

*en movimiento*



UNACH-RGF-01-04-08.15  
VERSIÓN 01: 06-09-2021

# CERTIFICACIÓN

Que, **Bryan José Loaiza Alejandro** con CC: **0706995347**, estudiante de la Carrera **Ingeniería en Telecomunicaciones**, Facultad de **Ingeniería**; ha trabajado bajo mi tutoría el trabajo de investigación titulado "**SISTEMA DE CLASIFICACIÓN DE BOTELLAS MEDIANTE EL RECONOCIMIENTO DE IMÁGENES PARA EL TRATAMIENTO DE BASURA**", cumple con el **1%**, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 06 de noviembre de 2023



Firmado electrónicamente por:  
**LEONARDO FABIAN  
RENERIA BUSTAMANTE**

PhD. Leonardo Fabian Rentería Bustamante  
**TUTORA**

## **DEDICATORIA**

Dedico el presente trabajo a mis padres por ser los dos principales pilares en los diferentes aspectos de mi vida, por su amor y confianza en mí, y a mis hermanos por su apoyo y servir de ejemplo positivo para mi formación académica.

## **AGRADECIMIENTO**

En primer lugar, agradezco a Dios por darme la fuerza espiritual para lograr mis objetivos.

A mis padres por confiar en mi desde el principio y apoyarme en todos los aspectos durante toda mi vida, por su amor incondicional en todo momento.

A mis hermanos por la ayuda brindada siempre, la cual me permitió cumplir muchos objetivos.

A mi arrendataria la Sra. Lastenia Ulloa, por su amabilidad, generosidad y cariño hacia mi persona durante toda mi estancia en su hogar.

A mi tutor el Dr. Leonardo Rentería por su paciencia, tiempo y compartir sus conocimientos con mi persona a lo largo de la ejecución del presente proyecto.

A la Universidad Nacional de Chimborazo por abrirme sus puertas y permitir que pueda formarme profesionalmente en tan prestigiosa institución.

# ÍNDICE GENERAL

**DECLARATORIA DE AUTORÍA**

**ACTA FAVORABLE**

**CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL**

**CERTIFICADO ANTIPLAGIO**

**DEDICATORIA**

**AGRADECIMIENTO**

**ÍNDICE GENERAL**

**ÍNDICE DE TABLAS.**

**ÍNDICE DE FIGURAS**

**RESUMEN**

**ABSTRACT**

<b>CAPÍTULO I.....</b>	<b>15</b>
<b>1.1 INTRODUCCION.....</b>	<b>15</b>
<b>1.2 ANTECEDENTES. ....</b>	<b>17</b>
<b>1.3 PROBLEMA.....</b>	<b>18</b>
<b>1.4 Justificación. ....</b>	<b>19</b>
<b>1.5 Objetivos. ....</b>	<b>20</b>
1.5.1 GENERAL.....	20
1.5.2 ESPECÍFICOS.....	20
<b>CAPITULO II.....</b>	<b>21</b>
<b>2.1 ESTADO DEL ARTE. ....</b>	<b>21</b>
<b>2.2 MARCO TEÓRICO. ....</b>	<b>22</b>
2.2.1 Internet de las cosas (IoT). ....	22
2.2.2 Inteligencia artificial.....	23
2.2.2.1 Enfoque simbólico. ....	23
2.2.2.2 Enfoque sub-simbólico. ....	23
2.2.2.3 Aplicaciones de la inteligencia artificial.....	23
2.2.2.4 Machine learning. ....	23
2.2.2.5 Deep learning. ....	24
2.2.2.5.1 Aprendizaje supervisado. ....	24
2.2.2.5.2 Aprendizaje no supervisado. ....	24
<b>2.3 Redes neuronales.....</b>	<b>25</b>
2.3.1 Redes neuronales convolucionales.....	26

2.3.2	Redes neuronales recurrentes. ....	26
<b>2.4</b>	<b>Visión artificial. ....</b>	<b>27</b>
2.4.1	Aplicaciones de la visión artificial .....	28
<b>2.5</b>	<b>Dispositivos y programación. ....</b>	<b>28</b>
2.5.1	ESP32 .....	28
2.5.2	Google colab.....	29
2.5.3	YOLOv5.....	29
2.5.3.1	Arquitectura de YOLO. ....	30
2.5.3.2	Funcionamiento de YOLO.....	31
2.5.3.3	Tamaño del set de imágenes. ....	32
2.5.3.4	Set de imágenes de entrenamiento.....	32
2.5.3.5	Set de imágenes de validación.....	32
2.5.3.6	Etiquetas.....	32
2.5.3.7	Épocas.....	33
2.5.3.8	BatchSize. ....	33
<b>CAPÍTULO III.</b>	<b>.....</b>	<b>34</b>
<b>3.1</b>	<b>METODOLOGIA. ....</b>	<b>34</b>
3.1.1	Tipo de Investigación. ....	34
3.1.2	Métodos de investigación. ....	34
3.1.2.1	Método analítico. ....	34
3.1.3	Técnicas de investigación.....	34
3.1.4	Población y muestra.....	34
3.1.4.1	Población.....	34
3.1.4.2	Muestra. ....	35
3.1.5	Operacionalización de variables.....	35
3.1.5.1	Variables Independientes.....	35
3.1.5.2	Variables Dependientes. ....	36
3.1.6	Procedimiento.....	36
3.1.7	Diseño del prototipo. ....	37
3.1.7.1	Software del prototipo.....	37
3.1.7.1.1	Entrenamiento del modelo.....	37
3.1.7.1.1.1	Parámetros de entrenamiento.....	37
3.1.7.1.2	Código para las detecciones. ....	37
3.1.7.1.3	Código del motor de la banda transportadora. ....	37
3.1.7.1.4	Código del sistema de clasificación. ....	38
3.1.8	Parte física del prototipo.....	38
3.1.8.1	Mesa vibratoria. ....	38
3.1.8.1.1	Materiales de la mesa vibratoria.....	39
3.1.8.2	Motor acoplado.....	39
3.1.8.3	Banda transportadora de basura.....	40
3.1.8.3.1	Materiales de construcción de la banda transportadora.....	40
3.1.8.3.1.1	Base.....	40

3.1.8.3.1.2	Rodillos: .....	40
3.1.8.3.1.3	Recámara de detección: .....	41
3.1.8.3.1.4	Cinta transportadora: .....	42
3.1.8.3.1.5	Motor de la banda transportadora. ....	42
3.1.8.3.1.6	Sistema de clasificación. ....	44
<b>CAPÍTULO IV</b>	.....	<b>46</b>
<b>4.1</b>	<b>RESULTADOS Y DISCUSIÓN</b> .....	<b>46</b>
<b>4.2</b>	<b>Hipótesis</b> .....	<b>47</b>
4.2.1	Hipótesis 1. ....	47
4.2.2	Hipótesis 2. ....	47
<b>4.3</b>	<b>Prueba de normalidad de muestras</b> .....	<b>47</b>
<b>4.4</b>	<b>Prueba de Hipótesis</b> .....	<b>48</b>
<b>4.5</b>	<b>Prueba de grupos</b> .....	<b>49</b>
<b>CAPÍTULO V</b>	.....	<b>53</b>
<b>5.1</b>	<b>CONCLUSIONES</b> .....	<b>53</b>
<b>5.2</b>	<b>RECOMENDACIONES</b> .....	<b>53</b>
<b>CAPÍTULO VI</b>	.....	<b>54</b>
<b>6.1</b>	<b>PROPUESTA</b> .....	<b>54</b>
<b>ANEXOS</b>	.....	<b>58</b>

## ÍNDICE DE TABLAS.

Tabla 1. Estudio descriptivo de cada población del error en la detección.	46
Tabla 2. Estudio descriptivo de cada población del error en la clasificación.	46
Tabla 3. Prueba de normalidad para el número de botellas no detectadas y no clasificadas de las tres poblaciones.	48
Tabla 4. Pruebas de hipótesis para la cantidad de botellas no detectadas y no clasificadas.	48
Tabla 5. Prueba de Tukey para la cantidad de botellas no detectadas.	49
Tabla 6. Prueba de Tukey para la cantidad de botellas no clasificadas.	49

## ÍNDICE DE FIGURAS

Figura 1. Ejemplo de aprendizaje no supervisado [24].	25
Figura 2. Ejemplo de una estructura común de red neuronal convolucional [31].	26
Figura 3. Arquitectura de una red neuronal recurrente básica [33].	27
Figura 4. Ejemplo de detención de diferentes objetos con visión artificial [34].	27
Figura 5. Modulo ESP32 usado [37].	29
Figura 6. Velocidad de inferencia (ms)s vs precisión promedio media (mAP) de las versiones de YOLOV5 [39].	30
Figura 7. La arquitectura de YOLO en su versión inicial [41].	31
Figura 8. Mesa Vibratoria.	39
Figura 9. Motor acoplado.	39
Figura 10. Base de la banda transportadora con los rodillos instalados.	40
Figura 11. Rodillos de movimiento.	41
Figura 12. Recamara de detección.	41
Figura 13. Cortinas de la recamara de detección.	42
Figura 14. Cinta transportadora.	42
Figura 15. Poleas (izquierda) y Motor acoplado (Derecha).	43
Figura 16. Servomotor adaptado al pistón.	44
Figura 17. Pistón clasificador.	44
Figura 18. Mesa vibratoria y banda transportadora del prototipo.	45
Figura 19. Media de la cantidad de botellas no detectadas de las tres poblaciones.	50
Figura 20. Media de la cantidad de botellas no clasificadas de las tres poblaciones.	50
Figura 21. Diagrama de cajas para el numero de botellas no detectadas para las tres poblaciones.	51
Figura 22 . Diagrama de cajas para el numero de botellas no clasificadas para las tres poblaciones.	51

## RESUMEN

El presente trabajo tiene como objetivo crear un prototipo para la clasificación de botellas, ya sean estas de vidrio o plástico o cualquier otro material usando reconocimiento de imágenes. El prototipo diseñado consta de dos partes, la primera corresponde a una mesa vibratoria para dispersar la basura, mientras que la segunda parte corresponde a una banda transportadora donde se instala el sistema de detección y clasificación de las botellas.

El modelo utilizado para la detección es YOLOv5s, el código para la detección se escribe en lenguaje Python, mientras que el código para la clasificación se desarrolla a través del IDE de arduino y el microcontrolador ESP32.

El prototipo se realiza en tres fases. En la primera fase, se realiza el estudio del estado del arte, así como el análisis de mecanismos y algoritmos. En la segunda fase se entrena el modelo de detección, se elaboran los códigos respectivos y se elabora la parte física del proyecto. Finalmente, en la fase tres se integra las dos partes y se procede a la recolección de datos.

Después de 2700 pruebas para poder analizar su comportamiento en diferentes escenarios, se determinó que el prototipo tiene tendencia a cometer más errores en la detección y clasificación, con forme sube el número de botellas que ingresan desde la mesa vibratoria. Sin embargo, tras el análisis estadístico aplicado se concluye que el prototipo diseñado posee un alto nivel de acierto en las detecciones y clasificaciones de las botellas.

**Palabras claves:** Reconocimiento de imágenes, YOLOv5s, python, microcontrolador ESP32.

## ABSTRACT

The objective of this work is to create a prototype for the classification of bottles, whether they are made of glass plastic or any other material, using image recognition. The prototype consists of two parts: the first corresponds to a vibrating table to disperse the garbage. In contrast, the second part corresponds to a conveyor belt where the detection and classification system of the bottles is installed.

The model used for the detection is YOLOv5s; the code for the detection is written in Python language, while the code for the sorting is developed through the arduino IDE and the ESP32 microcontroller.

The prototype is done in three phases. In the first phase, the state of the art is studied, as well as the analysis of mechanisms and algorithms. In the second phase, the detection model is trained, the respective codes are developed, and the physical part of the project is elaborated. Finally, phase three integrates the two parts and proceeds to data collection.

After 2700 tests to analyze its behavior in different scenarios, it was determined that the prototype tends to make more errors in detection and classification as the number of bottles entering from the vibrating table increases. However, after the statistical analysis applied, it was concluded that the designed prototype has a high level of accuracy in the detection and classification of the bottles.

**Keywords:** Image recognition, YOLOv5s, python, ESP32 microcontroller.



Reviewed by:  
Mgs. Dario Javier Cutiopala Leon  
**ENGLISH PROFESSOR**  
c.c. 0604581066

# CAPÍTULO I

## 1.1 INTRODUCCION.

A nivel Nacional e internacional los desechos de basura significan un problema, sobre todo debido a la contaminación que estos desechos son capaces de producir en el ecosistema. Los plásticos, vidrios, cartón, entre otros, son materiales que comúnmente se usan para fabricar objetos que luego son desechados por las personas. Desechos de esta clase pueden llegar a ser reutilizados para fabricar otro tipo de elementos mediante un proceso correcto y amigable con el medio ambiente. Solo en Ecuador para el año 2021 se importaron 48.473 toneladas de plásticos entre 2018 y enero de 2022 [1]. Este tipo de residuo viene a Ecuador con el fin de generar hilo sintético, fundas de basura, entre otros. El tema llama la atención, ya que Ecuador entierra el 96 % de su basura y del restante 4 % recuperado, un porcentaje del 85 % es tratado por recicladores que laboran en basurales a cielo abierto [2].

Dentro de los desechos de basura se pueden encontrar diferentes objetos creados de distintos materiales, por mencionar algunos de ellos están botellas, tarrinas y platos. Las botellas puntualmente son una clase de desecho que puede estar elaborado de plástico, cartón o vidrio en la mayoría de los casos. Al no tener un tratamiento adecuado esta clase de objetos son un riesgo para el medio y los seres vivos, teniendo sobre todo un efecto grave sobre la fauna marina y las aves. Los efectos contaminantes se agravan debido a los cientos de años en que tarda el plástico y el vidrio en descomponerse provocando que los residuos sean ingeridos por animales [3].

En Ecuador no existen datos específicos que indiquen un porcentaje del total de basura que ocupan las botellas; Sin embargo, hay datos más generales como el del año 2020 donde se produjeron 375 mil toneladas de residuos sólidos urbanos al año en el país [4]. El dato es de suma importancia para comprender que en el país se maneja un volumen considerable de desechos a los cuales se le puede aplicar un proceso que permita reutilizarlos dependiendo de la clase de material.

La pregunta que surge es del ¿por qué?, si existen los desechos en Ecuador, las empresas del país que trabajan con estos residuos deben recurrir a los de otros países para cubrir su demanda. En una entrevista a empresas que han experimentado la importación de desechos de plásticos en el Ecuador se afirmó que la importación de esos residuos se debe a que en Ecuador no existe suficiente oferta de esta clase de materiales, sumado a que la capacidad de reciclaje en el país no es comparable con la de los países exportadores [2].

La inteligencia artificial (IA, de sus siglas en español), es una tecnología útil en muchos campos, llegando a contribuir como una solución a la contaminación ambiental por desechos como lo son las botellas, y es que Lasse Rouhiainen menciona que la inteligencia artificial, es “La habilidad de los ordenadores para hacer actividades que normalmente requieren

inteligencia humana” [5]. Una aplicación de la inteligencia artificial son las redes neuronales artificiales [6]. Estas redes simulan el comportamiento de un cerebro humano, pueden ser implementadas para una serie de aplicaciones, entre ellas la visión artificial [7] y por ende el reconocimiento de imágenes [8].

Imaginarse un sistema que clasifique de forma automática objetos como las botellas a través de reconocimiento de imágenes, simulando la tarea de los recicladores es un paso importante para el reciclaje correcto. Si en Ecuador uno de los problemas de la importación de desechos de plástico es que en el país no exista un reciclaje adecuado, ¿Qué pasaría si se implementara un sistema que separe automáticamente las botellas que estén hechas de este material y de otros? Sin duda un sistema de esta clase permitiría que los desechos como botellas no sean enterrados junto con los de otro tipo, sino que reciban el tratamiento adecuado o sean reutilizados de acuerdo a la demanda de las empresas.

## **1.2 ANTECEDENTES.**

Trabajos similares al desarrollado en este documento se han planteado como soluciones a diferentes problemas con botellas de diferentes tipos, entre los que están: Detectar el nivel de líquido en botellas [9], Clasificación de botellas por marcas [10], entre otros trabajos, donde se utilizan diferentes técnicas a nivel de hardware y software propios de cada proyecto de acuerdo a sus limitaciones.

Algunos trabajos revisados en el estado del arte fueron clave para comprender las mejores técnicas con las que se han trabajado para la detección de botellas, y así mismo, las limitaciones del proyecto de acuerdo al hardware y software disponible, tales como [11] y [12], donde se buscó precisamente diseñar sistemas que aporten a la clasificación de botellas usando técnicas de inteligencia artificial y redes neuronales artificiales.

La utilización de cámaras especiales, la detección de rasgos distintivos, la implementación de diferentes fondos con colores distintos como el blanco, son algunas de las técnicas propuestas en otros trabajos como [13]. La revisión de estos trabajos previos contribuyó de manera positiva para comprender de forma profunda las técnicas aplicables a la detección y clasificación de botellas, ya que se orientan a la misma línea de trabajo que el presente proyecto de tesis. Para conocer a más detalle cada proyecto siempre se puede consultar los trabajos referenciados en el estado del arte.

### **1.3 PROBLEMA.**

Materiales contaminantes para el medio ambiente como lo son el vidrio, plástico, cartón entre otros, no reciben un tratamiento adecuado en el Ecuador, pues desde un punto de vista ecológico en el país se entierra gran parte de la basura producida sin que este tipo de materiales tenga un tratamiento adecuado, para ser exactos el 96% [2]. Las botellas fabricadas utilizando estos tipos de materiales y otros por lo general acaban enterradas junto con otros residuos en los lugares destinados a la basura. Un prototipo de sistema inteligente a ser desarrollado en el laboratorio de electrónica de la Universidad Nacional de Chimborazo (UNACH), permitiría detectar botellas de un grupo de desechos y luego clasificarlas en un recipiente propio para ellas. Esto determinaría un paso muy importante en el correcto y adecuado reciclaje creando un gran aporte en el tratamiento de estos materiales.

## **1.4 JUSTIFICACIÓN.**

El proyecto presentado aporta con la creación de un sistema inteligente de clasificación de botellas utilizando reconociendo de imágenes en la ciudad de Riobamba, además abre nuevas oportunidades a líneas de investigación futuras en la universidad nacional de Chimborazo en el tratamiento de basura y su importancia en el País. El sistema creado beneficia a la comunidad de tal manera que permite contribuir a la reducción de la contaminación de diferentes espacios con desechos de este tipo, adicionalmente ofrece una oportunidad para que estos desechos no terminen enterrados junto con otros y reciban un tratamiento adecuado que permita reutilizarlos, logrando una ecología amigable y contribuir a la demanda de empresas nacionales que requieren desechos de esta clase para cumplir con sus procesos industriales, facilitando además las tareas de clasificación de residuos por individuos que trabajan en este sector. El proyecto es expandible, pudiendo desarrollarse a mayor escala para cubrir zonas estratégicas donde la recolección de basura sin una clasificación de objetos de este tipo es un problema.

## **1.5 OBJETIVOS.**

### **1.5.1 GENERAL.**

- Diseñar e implementar un sistema de clasificación de botellas mediante el reconocimiento de imágenes, para el tratamiento de basura.

### **1.5.2 ESPECÍFICOS.**

- Analizar los diferentes mecanismos y algoritmos de redes neuronales para el diseño y construcción de un prototipo para la clasificación de botellas.
- Entrenar un modelo adecuado de red neuronal mediante recolección de imágenes, basadas en la morfología, para la detección de botellas.
- Integrar e implementar los mecanismos físicos y lógicos diseñados a través de un prototipo electromecánico funcional para la clasificación de las botellas tras su detección.
- Validar el funcionamiento del prototipo implementado, utilizando los datos recolectados en pruebas de campo, mediante técnicas estadísticas como correlación de variables y ajuste de curvas.

## CAPITULO II

### 2.1 ESTADO DEL ARTE.

Como se explicó previamente en la introducción y el planteamiento del problema, el objetivo de este proyecto es diseñar e implementar un prototipo de clasificación de botellas usando el reconocimiento de imágenes. Se conoce que ya se han desarrollado varios trabajos relacionados con el reciclaje y la utilización de visión artificial para reconocer características propias de objetos, entre ellos envases como las botellas.

Un ejemplo de lo antes expuesto es el proyecto explicado en [9], donde se diseñó e implemento a través de visión artificial un sistema seleccionador de botellas basándose en el líquido y el tapón presente en las mismas. Se usaron técnicas de procesamiento de imágenes como eliminación del fondo (Background Subtraction), Filtro umbral (Filtro Thresholding) y Umbral binario (Threshold Binary), adicionalmente se usa una placa Raspberry pi 48 y se utiliza herramientas de código abierto como OpenCV y herramientas para la creación de prototipos como SolidWorks.

En [14] en cambio se lleva a cabo un reconocimiento de imágenes aplicado a botellas de plástico con el fin de contribuir al reciclaje, el sistema alcanza el 94.7% de precisión, aquí se menciona el valor agregado que tienen las botellas de diferentes colores para el reciclaje. Se usan técnicas de identificación de posición y color como puntos clave para el funcionamiento del sistema, en cuanto a la posición los autores proponen en primer lugar definir una relación entre la posición de las botellas pudiendo estar estas de manera, adjuntas, disjuntas y superpuestas. En cuanto al color se usa el algoritmo ReliefF en la búsqueda de seleccionar las características de color de las botellas y posteriormente se usa el algoritmo de la máquina de vectores de apoyo (SVM) para identificar el color.

En materia de equipos algunos autores difieren en los computadores de bajo costo usadas, es así que una alternativa a la placa Raspberry pi 48 implementada en [9] es la Spartan-6. En [11] se usa esta placa para desarrollar un sistema de clasificación de botellas en tiempo real que se basa en la unión de varios bloques que pueden funcionar como bloques individuales. Tal como lo menciona el autor el sistema es de alto rendimiento para el reconocimiento por color.

De manera similar en [12] se explica generalmente el diseño de un sistema capaz de realizar un reconocimiento automático de botellas plásticas empleando redes neuronales artificiales. El trabajo analiza la importancia de aplicar técnicas de procesamiento de imágenes para identificar patrones robustos en objetos.

En [10] se desarrolló en la ciudad de Milagro-Ecuador un prototipo para la detección de botellas plásticas, el mismo se basó en una red neuronal convolucional con resultados

porcentuales de efectividad en la detección para botellas de la marca CocaCola con un porcentaje de 97%, Sprite con un 85% y Fanta con un 72%.

Así mismo en [13] en la ciudad de Quito-Ecuador se logró mediante visión artificial el diseño de un sistema automático para la clasificación de bebidas, para su funcionamiento se tomaron en consideración parámetros importantes del sistema como lo son la carga máxima de la banda transportadora y la potencia de elementos como los motores. Además, se hace uso del software MATLAB para analizar el sistema de visión artificial, aquí el sistema presenta una buena efectividad, reconociendo de manera correcta 19 de 20 envases de prueba.

Los resultados en trabajos estudiados en este marco referencial permiten comprender como un prototipo de sistema de reciclaje puede asemejarse bastante a las funcionalidades de un equipo industrial de mayor escala. Así mismo, en este punto resulta útil para el desarrollo del trabajo de tesis el análisis de las recomendaciones a lo largo del estudio de trabajos referenciales, tales como el aislamiento del sitio de captura de la imagen de luces externas para una correcta identificación de los objetos, la utilización de una cámara de alta resolución para mejorar el reconocimiento por visión artificial, el uso de placas como la Raspberry PI para aplicaciones de bajo costo.

Por otro lado, frente al problema con la identificación de colores como el azul claro, lila e incoloro que se pueden presentar se aconseja la selección de colores con mejores características para evitar que el sistema se vuelva impreciso. Otro factor importante a tener en cuenta es la fuente de luz y color de la cinta transportadora, la recomendación de probar con diferentes colores de la cinta transportadora concuerda con [13], donde la mejor opción fue implementar un color blanco debido a las técnicas usadas.

## **2.2 MARCO TEÓRICO.**

### **2.2.1 Internet de las cosas (IoT).**

En el mundo actual, con el avance de la tecnología pensar en un futuro donde los mecanismos eléctricos y electrónicos realicen las tareas de los seres humanos ya es una realidad, el internet de las cosas o (IoT) extiende esa posibilidad a muchas áreas, y es que como se menciona en [15] el internet de las cosas ha sufrido una evolución por la necesidad de las personas de conectar objetos con diferentes características a un solo sistema que funcione para recolectar, almacenar y enviar información a través de la red.

A su vez, con el intercambio de información el IoT ha traído consigo retos que deben ser manejados, por ejemplo, la seguridad, privacidad y movilidad. Factores que son clave para que el IoT sea aplicable de forma segura en muchas áreas de trabajo, donde puede ser implementado para mejorar la productividad, eficiencia y calidad de procesos.

## **2.2.2 Inteligencia artificial.**

Se puede definir la inteligencia artificial como la capacidad de un ordenador o un grupo de ellos para la ejecución de acciones que normalmente requieren la intervención de la inteligencia humana para ser llevados a cabo, tal como se menciona en [5]. Esta definición es concordante con la propuesta en [16], donde se cita Marvin Minsky un pionero en inteligencia artificial que definió la misma como una ciencia capaz de desarrollar máquinas con la habilidad de ejecutar procesos para los cuales se necesitaría inteligencia humana.

### **2.2.2.1 Enfoque simbólico.**

Si lo que se busca es introducirse un poco más en el significado de inteligencia artificial, se puede observar la misma a través de dos enfoques, la IA simbólica y la IA neuronal, la primera se basa en la utilización de símbolos, conjuntos de reglas y conclusiones lógicas que conjuntamente emulan el razonamiento humano. Como se entiende en [17] y [18] este enfoque es aplicable a temas lógico-matemáticos, buscando desde sus inicios dotar a las máquinas con la capacidad lógica de resolver cualquier problema que tuviera una posible solución.

### **2.2.2.2 Enfoque sub-simbólico.**

Por otro lado, la IA neuronal conocida también como sub-simbólica, se basa en la utilización de redes neuronales que trabajando de forma conjunta permiten crear sistemas con capacidad de aprendizaje, imitando el comportamiento del cerebro humano [19]. Esto permite que esta clase de IA tenga aplicabilidad en muchas áreas de trabajo donde es requerida la intervención humana. Este enfoque es sobre el cual se basa el presente trabajo.

### **2.2.2.3 Aplicaciones de la inteligencia artificial.**

Si se piensa en las aplicaciones de la inteligencia artificial, sería útil primero pensar donde la inteligencia del ser humano puede ser aplicable, esto deja un millón de posibilidades. Así mismo, la IA puede ser aplicada en muchos procesos, sean estos simples o complejos. Algunas áreas en las que está involucrada la IA son el reconocimiento de imágenes, minería de datos, interfaces inteligentes, algoritmos genéticos, realidad aumentada, Machine learning y Deep learning, entre otros.

### **2.2.2.4 Machine learning.**

Machine learning o aprendizaje automático en español, es una rama de la inteligencia artificial que busca pretender que las máquinas puedan realizar tareas inteligentes [20]. Este

concepto es similar al de inteligencia artificial, pero en específico esta rama de la misma se basa en dotar a las máquinas de una capacidad de aprendizaje que posteriormente sirve como medio para que las mismas puedan ejecutar tareas por sí s solas.

Dentro del machine learning se pueden usar algoritmos matemáticos, árboles de decisión, modelos de regresión, entre otros [21]. Estas técnicas permiten obtener soluciones inteligentes a diferentes problemas, pero sin duda una técnica que ha ganado mucha popularidad en la actualidad son las redes neuronales.

### **2.2.2.5 Deep learning.**

El Deep learning o aprendizaje profundo se deriva directamente del machine learning, con la diferencia que en esta subdivisión se aplican las redes neuronales como técnica para proveer a las máquinas de la habilidad de aprendizaje [21], [22]. Estas redes neuronales, como ya se describió en apartados anteriores buscan imitar el comportamiento del cerebro humano a la hora de aprender y solucionar problemas.

Es lógico pensar que cuando se trata de aprendizaje inducido a una máquina, debe haber un límite en lo que puede o no aprender, para entender un poco mejor esto es necesario mencionar términos como el aprendizaje supervisado y no supervisado, que no son más que la forma en como los seres humanos estamos interviniendo en el proceso de aprendizaje de una máquina.

#### **2.2.2.5.1 Aprendizaje supervisado.**

El aprendizaje supervisado es un tipo de aprendizaje en el cual el ser humano como supervisor le entrega datos de entrada a una máquina y del mismo modo le indica a la misma el resultado que debe obtener con esos datos de entrada [23]. Es decir, la máquina aprende una relación entre los datos de entrada y de salida que deben obtenerse.

Los modelos para detección de objetos, animales, entre otros, son modelos basados en aprendizaje supervisado, si bien se entiende que son capaces de detectar los objetos indicados por sí solos, estos no pudieran realizar estas detecciones sin que antes existiera la intervención humana durante el proceso de aprendizaje de las estos modelos, de allí el nombre de aprendizaje supervisado.

#### **2.2.2.5.2 Aprendizaje no supervisado.**

Por otro lado, el aprendizaje no supervisado hace referencia a un tipo de aprendizaje en el cual el ser humano como supervisor únicamente entrega datos de entrada a una máquina y

la misma por sí sola es capaz de encontrar una relación entre esos datos y arrojar un resultado [23].

Este tipo de aprendizaje puede buscar patrones de similitud entre los datos que se tiene de entrada, pudiendo por consecuencia agrupar o desagrupar los mismos de acuerdo a sus características. Como se puede entender, el aprendizaje no supervisado debe su nombre a que el ser humano únicamente brinda los datos de entrada, más no supervisa el comportamiento que esta máquina tiene con estos datos de entrada.

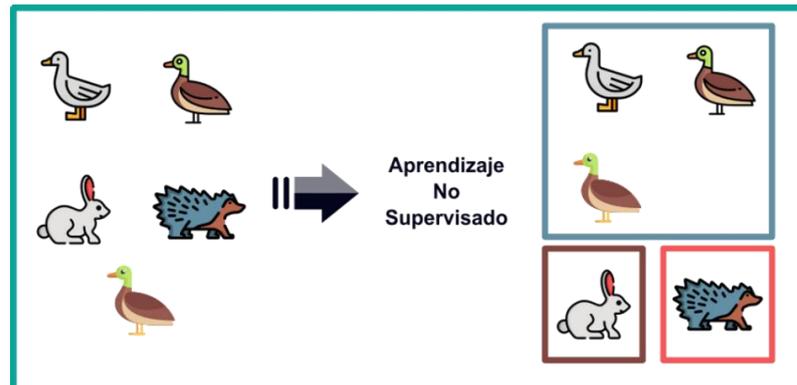


Figura 1. Ejemplo de aprendizaje no supervisado [24].

En la Figura 1 se puede observar el comportamiento del aprendizaje no supervisado, a diferencia del supervisado, este aprendizaje busca similitudes en los datos de entrada, es así que se clasifica el grupo de entrada en 3 subclases de animales diferentes, sin que la intervención humana le diga al modelo como agrupar los datos.

### 2.3 Redes neuronales.

Una neurona es la unidad básica de procesamiento dentro de un sistema o red neuronal, desde un punto de vista biológico se entiende que todos los seres vivíos entre ellos, por ejemplo, los seres humanos, son capaces de realizar tareas como pensar, imaginar, resolver problemas, etc, gracias a las neuronas que el cuerpo humano posee en su interior. Estas neuronas a lo largo de sistema nervioso y el cerebro permiten a los seres vivos disfrutar de varias habilidades cognitivas, entre otras [25].

Esta idea se aplica a las redes neuronales artificiales, donde se tienen neuronas que pueden ser vistas como funciones, pero con un nombre diferente. Estas neuronas funcionan realizando sumas ponderadas, que responden a estímulos de entrada externos. En los modelos de Deep learning, estas neuronas se ubican por capas, permitiendo que cada capa o grupo de neuronas aprendan información específica sobre algo y utilicen esa información para resolver problemas más complejos en otras capas [26], [27].

### 2.3.1 Redes neuronales convolucionales.

Las redes neuronales convolucionales o convolutional Neural Networks (CNN), se caracterizan por ser redes que poseen neuronas divididas por capas [28].

Este tipo de redes neuronales se usa dentro del Deep learning para el análisis o reconocimiento de imágenes, las mismas son usadas por modelos conocidos como YOLO para la detección de una serie de objetos como botellas, cuadernos, carros, entre otros, y también personas y animales [29].

Las redes neuronales convolucionales se caracterizan por ser redes que poseen neuronas divididas por capas [28]. Estas capas tienen la capacidad de identificar características específicas de los datos de entrada (imágenes), las siguientes capas realizan aprendizajes más profundos, así se construye una red neuronal compleja que es capaz de realizar tareas como reconocer objetos de diferentes tipos [30]. Todo esto gracias a un preentrenamiento con set de imágenes. En la Figura 2 se observa la estructura común de una red neuronal convolucional, este tipo de redes cuenta con una serie de capas convolucionales, capas de reducción y otras capas completamente conectadas, también llamadas capas clasificadoras al final de la red.

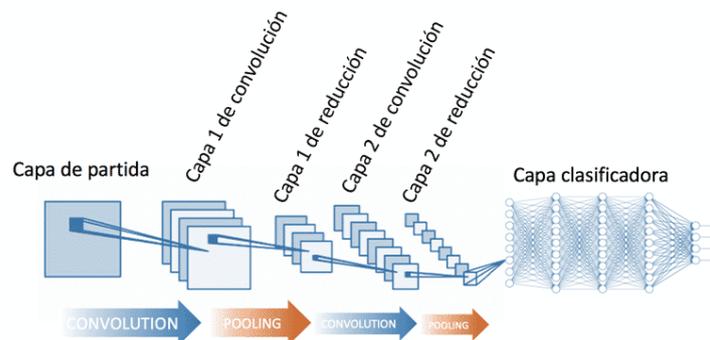


Figura 2. Ejemplo de una estructura común de red neuronal convolucional [31].

### 2.3.2 Redes neuronales recurrentes.

Las redes neuronales recurrentes o Recurrent Neural Network (RNN), se diferencian de las convolucionales por trabajar con datos en forma de serie o secuencia, esto permite que este tipo de redes sean usadas por herramientas conocidas como deepL, aplicaciones para reconocimiento de la voz, etc [32].

Si se piensa en una red neuronal convolucional se entiende que la red es capaz de arrojar un resultado dependiendo de lo que tiene en su entrada, por ejemplo, si se coloca como entrada la imagen de un carro, la red detectará esto y gracias a su preentrenamiento determinará que la salida debe ser carro, sin embargo, en una red neuronal recurrente no sucede esto, este tipo

de redes toma en cuenta la información generada en instantes de tiempo anteriores, de ahí que este tipo de redes tienen una forma memoria.

En las redes neuronales recurrentes, dado que se trabaja con secuencias de valores la información extraída en un instante de tiempo  $t$  afecta a los instantes  $t + 1, 2, 3, \dots$ . Del concepto básico de red neuronal recurrente se despliegan otros tipos de redes como las Long Short Term Memory (LSTM) y las Gated recurrent unit (GRU) que se pueden considerar mejoras de la arquitectura básica de red recurrente [29]. En la Figura 3 se puede observar la arquitectura de una red neuronal recurrente básica.

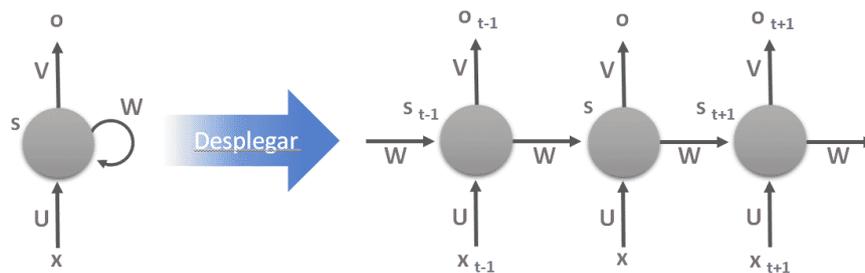


Figura 3. Arquitectura de una red neuronal recurrente básica [33].

## 2.4 Visión artificial.

La visión artificial puede ser considerada como una disciplina cuyo nombre se le da a la capacidad de un software y hardware que combinados son capaces de imitar el comportamiento de la vista humana en ciertos procesos. La visión artificial busca procesar imágenes con el fin de extraer de las mismas cierta información numérica o simbólica, la que será útil para procesos más complejos como la detección de objetos en una imagen o video [9].



Figura 4. Ejemplo de detección de diferentes objetos con visión artificial [34].

La Figura 4 muestra la detección de diferentes clases de objetos como motocicletas y personas a través de la visión artificial, ambas con diferentes niveles de confianza de; 48%, 72%, etc. Estos niveles se pueden aumentar con un entramiento más cuidadoso de la red neuronal.

Cada vez con el avance de la tecnología, sobre todo en temas como la visión artificial, las tareas que realiza una persona pueden ser ya reemplazadas por máquinas realizando los mismos trabajos y de forma incluso más eficiente, esto se debe a que las máquinas no cuentan con el factor del error humano, ya sea por cansancio, descuido, entre otros, las personas son propensas a cometer errores después de largas jornadas continuas de trabajo. Por otro lado, una máquina dotada del software y hardware adecuados es capaz de cumplir satisfactoriamente varias tareas durante largas jornadas de funcionamiento sin requerir descansos.

### **2.4.1 Aplicaciones de la visión artificial**

La visión artificial en la actualidad tiene aplicaciones en la mayoría sin es en todos los campos existentes donde la visión humana es útil. Mientras que un ser humano tiene que descansar para preservar su salud visual, una cámara no tiene que hacerlo, pues con la ingeniería adecuada una maquina es capaz de funcionar de forma continua por largos periodos de tiempo, optimizando los procesos. Por citar algunos ejemplos, tal y como se menciona en [35], el visón artificial aparece en campos como:

- **Agricultura.** - En la detección de plagas altamente dañinas para los cultivos.
- **Inspección.** - y control de calidad. - En el control de calidad de frutas, y envases como botellas.
- **Seguridad.** - En la detección de personas y/o rostros.
- **Clasificación.** - En la clasificación de objetos como botellas, cuadernos, carros, etc.

## **2.5 Dispositivos y programación.**

### **2.5.1 ESP32**

ESP32 como se menciona en su página oficial, es una familia de microcontroladores (MCU) que trae consigo incluida conectividad Wi-Fi y Bluetooth, diseñada para una amplia gama de aplicaciones en diferentes áreas [36].

En este proyecto se usa la placa ESP-WROOM-32 de la familia ESP32, debido a la disponibilidad de la misma y las características que la hacen funcional para el proyecto

realizado, las especificaciones de esta placa se pueden estudiar en la página oficial de espressif. La Figura 5 muestra el modelo de placa utilizada, así como la distribución de sus 30 pines a lo largo de la placa.

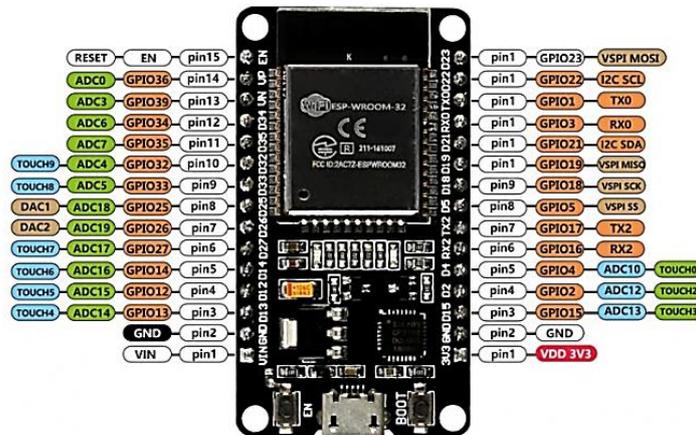


Figura 5. Módulo ESP32 usado [37].

## 2.5.2 Google colab.

Google colab es un producto de Google, el mismo brinda a los usuarios la oportunidad de escribir y ejecutar código en lenguaje Python en la nube, esto se da a través de cuadernos digitales o cuadernos de Colab [38].

Adicional a esto, Google Colab brinda la oportunidad de tener acceso a GPUs sin costo alguno, lo que resulta muy útil para aquellos usuarios que no cuentan con computadores de buenas características para ejecutar código pesado, como el entrenamiento de modelos para detección de objetos. Esta herramienta se utilizó para el entrenamiento del modelo YOLOv5s implementado en este trabajo.

## 2.5.3 YOLOv5.

YOLOv5 es una de las versiones del famoso modelo You Only Look Once (YOLO), que basa su funcionamiento en redes convolucionales. Este modelo permite realizar en una amplia gama de objetos tareas como la detección y segmentación. Es por ello que ha ganado popularidad en los últimos años. Sus diferentes velocidades de funcionamiento y demás características vuelven a YOLO un modelo de detección de objetos aplicable a casi todos los escenarios [39].

Cuando se requiere trabajar con una de las versiones de YOLO es necesario contar con un set de datos ya sea personalizado o preelaborado, estos últimos se pueden encontrar en internet. Aunque YOLO de manera predeterminada está entrenado con el set de datos COCO, siempre es posible crear un set de datos personalizado para realizar un entrenamiento

propio de cada caso, estos sets de imágenes deben contener fotografías de los objetos que se desea detectar, así como imágenes de fondos en algunos casos para descartar falsos positivos.

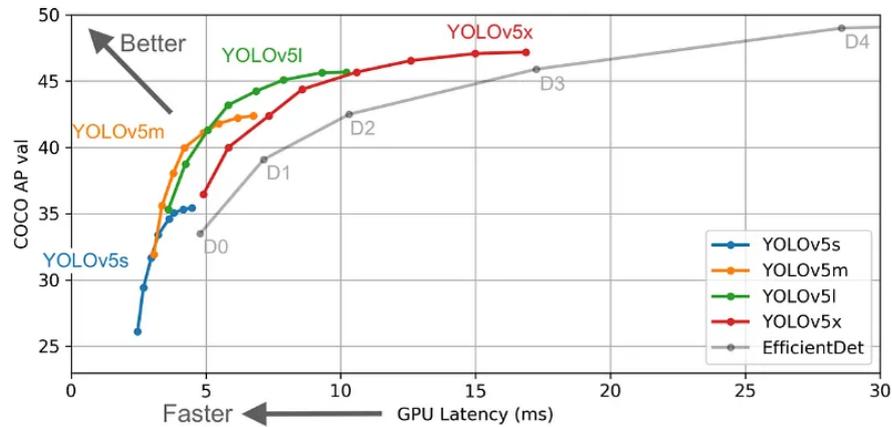


Figura 6. Velocidad de inferencia (ms)s vs precisión promedio media (mAP) de las versiones de YOLOV5 [39].

La velocidad de inferencia por imagen de las diferentes versiones de YOLOv5 y la precisión promedio media (mAP, por sus siglas en inglés) de las mismas se puede observar en la Figura 6, aquí se puede apreciar que la versión con mejores características para detección es la YOLOv5x, pues presenta una precisión media en las detecciones superior al 45%, sin embargo, también se observa que su velocidad de detección es menor en comparación con versiones más simples como la YOLOv5s. Para un análisis más profundo de cada versión se puede consultar su estructura en [39].

### 2.5.3.1 Arquitectura de YOLO.

Las diferentes versiones de YOLO han mejorado en varios aspectos a sus versiones anteriores, es por ello que las versiones cuentan con arquitecturas diferentes, unas más complejas que otras, pero manteniendo el principio de funcionamiento de la versión inicial.

La estructura de YOLO en su versión inicial consta de 24 capas convolucionales y adicionalmente 2 capas completamente conectadas. Para una reducción en el número de capas se usa una convolución 1x1, este procedimiento permite la reducción de la profundidad de los mapas de características, luego aparece una capa de convolución 3x3. Estas capas se alternan a lo largo de la arquitectura. La última capa de convolución, genera un tensor con forma (7,7,1024). Este tensor es reducido aplicando 2 capas completamente conectadas, dando como resultado un tensor de tamaño (7x7x30) [40]. Lo antes dicho se puede observar en la Figura 7.

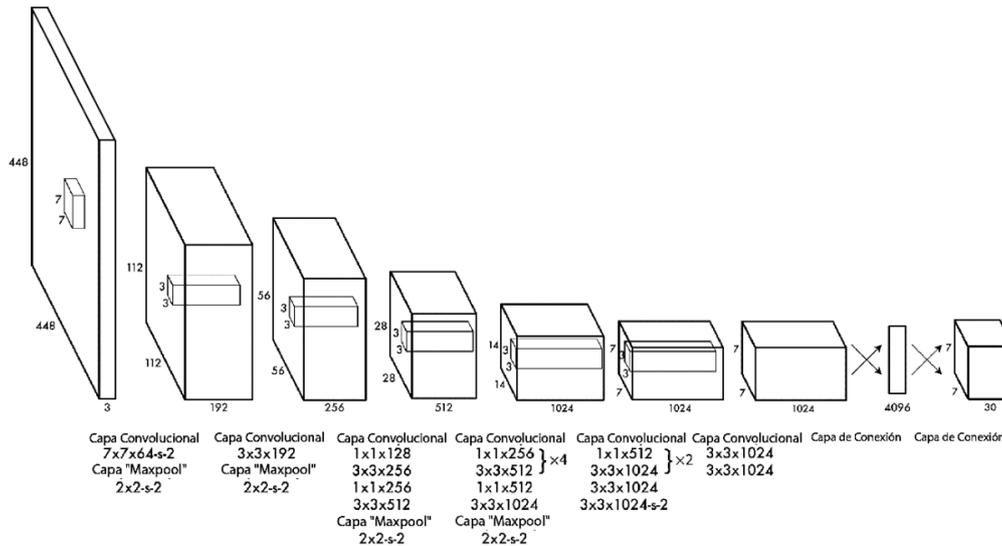


Figura 7. La arquitectura de YOLO en su versión inicial [41].

### 2.5.3.2 Funcionamiento de YOLO.

Conociendo la arquitectura de YOLO, sobre su funcionamiento se entiende que, recibida una imagen, el modelo ya entrenado recibe esta imagen de entrada y la divide en  $N$  rejillas de un tamaño  $S \times S$ , con el fin de que cada rejilla detecte características de la parte de la imagen que le corresponde, cada rejilla dará un nivel de confianza de que en esta rejilla se encuentra el objeto que se desea detectar.

Dentro de YOLO al dividir la imagen en una cuadrícula y realizar detecciones en cada una de las rejillas de esta cuadrícula, puede aparecer el problema de solapamiento de ventanas, que no es más que varias detecciones del mismo objeto en una misma rejilla. Para solucionar esto, YOLO implementa un proceso llamado Non Maximum Suppression, que realiza la eliminación de las detecciones que presentan un nivel de confianza menor al del nivel de confianza máximo.

Para iniciar con el entrenamiento dentro del modelo YOLO se tienen que tener en consideración parámetros de suma importancia a la hora de lograr un modelo con un entrenamiento óptimo, estos parámetros son el tamaño del set de imágenes, las etiquetas, las épocas, el batchsize, entre otros [40].

Para el proyecto de tesis desarrollado en este trabajo se usa YOLOv5s, ya que, tal como se recomienda en [42] para aplicaciones móviles es factible la utilización de YOLOv5s/m, pues versiones superiores pese a ser más precisas en las detecciones, también presentan problemas como su lentitud a la hora de ejecutarse, sobre todo si no se cuenta con buenas características de software y hardware.

### **2.5.3.3 Tamaño del set de imágenes.**

Cuando se realiza detección de objetos en modelos como YOLO, mientras más grande es el set de datos mejor. Un conjunto pequeño de imágenes de un objeto a detectar implicará que ese objeto no pueda ser detectado en todos los casos. Por otro lado, si se cuenta con un set de imágenes de ese mismo objeto en diferentes posiciones, diferentes niveles de luz, etc. Esto garantizará que las detecciones sean mucho más precisas.

Para el proyecto desarrollado se decidió trabajar un set de datos de 2479 imágenes, de las cuales 2181 imágenes se escogieron para el entrenamiento y 298 imágenes para la validación, todas las imágenes tienen un tamaño de (416x416) px, ya que YOLOv5 admite varios tamaños de imágenes.

### **2.5.3.4 Set de imágenes de entrenamiento**

Este set contiene todas las imágenes de entrenamiento para el modelo, como su nombre lo indica, son las imágenes a través de las cuales el modelo aprenderá las características de los objetos que se requiere detectar, para posteriormente poder realizar las detecciones de esos objetos. Este set es el más grande en comparación con el de validación.

### **2.5.3.5 Set de imágenes de validación.**

Este set contiene todas las imágenes de validación, que no son más que el grupo de imágenes que se utilizara para realizar un ajuste de los hiperparámetros del modelo, entendiéndose como hiperparámetros, por ejemplo, el número de capas y el número de neuronas por capa. Las imágenes de validación se utilizarán después de cada una de las épocas de entrenamiento.

Este set de imágenes evita el sobreajuste del modelo, que aparece cuando el modelo ha memorizado los posibles escenarios de respuesta, volviéndolo un modelo incapaz de realizar una detección confiable frente a imágenes nuevas. Esto por lo general sucede al realizar el entrenamiento y la validación con el mismo set de imágenes. De allí la importancia de que las imágenes en ambos sets sean diferentes y no copias. No existe con certeza un porcentaje que se debe respetar entre el set de datos de entrenamiento y validación, sin embargo, algunos autores recomiendan usar desde el 10% hasta el 30% de los datos totales para validación.

### **2.5.3.6 Etiquetas.**

Las etiquetas son un aparte fundamental del modelo YOLO y otros modelos del mismo tipo, estas etiquetas indican al modelo la clase de objeto que se está detectando. Por ejemplo, si se tiene un set de imágenes entre perros y gatos y se quiere detectar ambos animales individualmente, es necesario la utilización de dos etiquetas, una para cada clase de animal.

### **2.5.3.7 Épocas.**

Las épocas son parámetro importante, estas son la cantidad de veces que se itera sobre el set de datos de entrenamiento, por ejemplo, si se tiene un set de 20 imágenes y se configura 30 épocas, el tiempo que se debe esperar para que el modelo entrene con las 20 imágenes será el tiempo en que demore en recorrer 30 veces esas 20 imágenes. Un punto importante a tener en cuenta es que a mayores épocas daría la impresión de que es mejor para el modelo entrenado, pero no es así, el uso de demasiadas épocas innecesariamente se verá reflejado en la aparición de falsos positivos (detecciones que no son verdaderas), tiempo de espera innecesario y en general más recursos de hardware y software.

### **2.5.3.8 BatchSize.**

Durante la época de entrenamiento el set de datos se subdivide en lotes más pequeños, el tamaño de ese nuevo lote se conoce como batchsize. El batchsize indicará cuantas imágenes utilizará el modelo para entrenar al mismo tiempo en cada época. Por ejemplo, si se tiene un set de 100 de imágenes y se establece un batchsize de 50, el modelo entrenará con las primeras 50 imágenes y luego entrenará con las siguientes 50. El tamaño del batchsize dependerá de las características de la máquina con la que se cuente para realizar el entrenamiento, ya que un número muy elevado demandará más recursos de software y hardware.

## **CAPÍTULO III.**

### **3.1 METODOLOGIA.**

Para desarrollar este trabajo se ejecutarán las actividades en 3 fases. En la fase 1 se buscará y recopilará información de diferentes fuentes bibliográfica como libros, artículos de revistas, tesis, informes, entre otros. En la fase 2 se trabajará en el entrenamiento del modelo de red neuronal para la detección de botellas y se elaborará la parte física que complementa al funcionamiento del software. Finalmente, en la fase 3 se fusionará la parte del software con la parte física y se realizará la recolección de datos con pruebas de campo para validar el funcionamiento del prototipo.

#### **3.1.1 Tipo de Investigación.**

La investigación es experimental y exploratoria. Este tipo de investigación permite evaluar el comportamiento del sistema creado, pues maneja variables que contienen datos numéricos como por ejemplo la tasa de error.

#### **3.1.2 Métodos de investigación.**

##### **3.1.2.1 Método analítico.**

Para este trabajo de tesis se usará el método analítico, se analizarán todos los elementos del sistema desde los más básicos, así pues, se comprenderá su contribución al error que puede presentar el prototipo.

#### **3.1.3 Técnicas de investigación.**

La técnica utilizada es la recolección de datos a través de una serie de experimentos con las variables para poder determinar el error que presenta el sistema y así garantizar una mejor fiabilidad y eficiencia.

#### **3.1.4 Población y muestra.**

##### **3.1.4.1 Población.**

En base al experimento realizado se consideró una población de tipo infinita, conformada por el error en la detección y el error en la clasificación, los datos se obtienen con experimentos para: 10 botellas, 24 botellas y 50 botellas, con una distribución de 50% botellas de vidrio y 50% botellas de plástico.

### 3.1.4.2 Muestra.

Dada una población infinita, la muestra es obtenible mediante la fórmula (1):

$$n = \frac{Z^2 \cdot p \cdot q}{e^2} \quad (1)$$

Donde:

\*n = Tamaño de la muestra.

Z = Nivel de confianza (NC), (Para este caso será 90%, equivalente a un Z = 1.645))

p = Probabilidad de que ocurra el evento estudiado, (Dado que no se conoce toma un valor de 50 %).

q = Probabilidad de que no ocurra el evento estudiado, (Dado que no se conoce toma un valor de 50 %).

e = Error de estimación (Para este caso 10%).

$$n = \frac{Z^2 \cdot p \cdot q}{e^2} = 67,65 \approx 68$$

*Datos de la tasa de error en la detección y la Clasificación*

\*La muestra será tomada de forma aleatoria.

La muestra obtenida refleja un valor de 68, es decir de deberá extraer aleatoriamente 68 datos del error en la detección y error en la clasificación para 10 botellas, 24 botellas y 50 botellas. Sin embargo, considerando una población infinita se decide trabajar con un número mucho más grande de muestra, en este caso de 900 datos.

### 3.1.5 Operacionalización de variables.

#### 3.1.5.1 Variables Independientes.

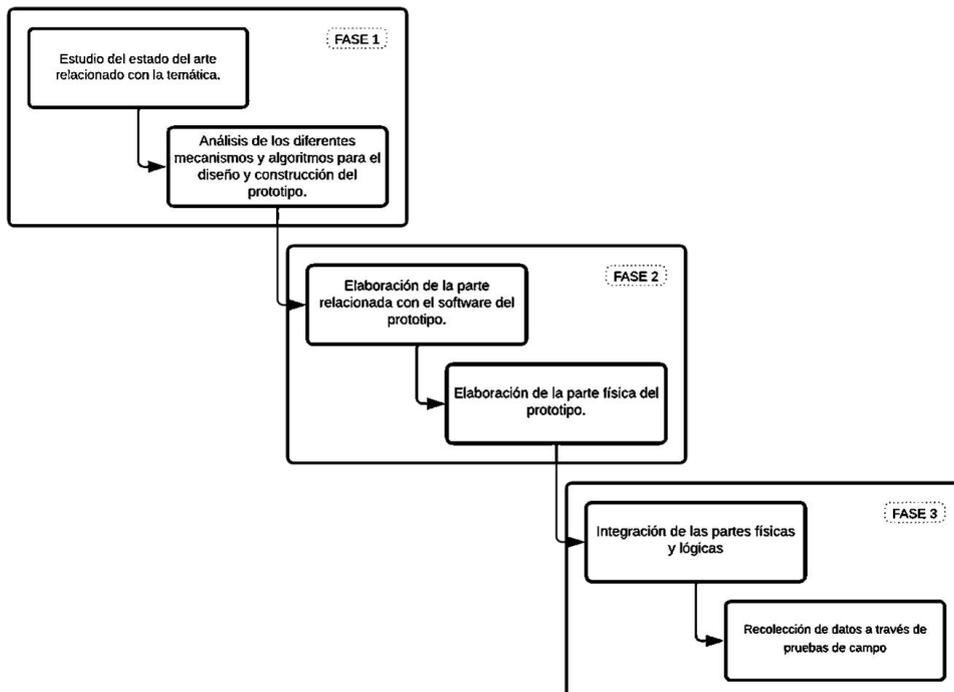
VARIABLES INDEPENDIENTES		DESCRIPCIÓN	INDICADOR	INSTRUMENTOS
DENSIDAD DE BOTELLAS.	DE	Cantidad de botellas en un determinado espacio.	Número de botellas.	Observación

<b>CANTIDAD BASURA</b>	<b>DE</b>	2 bolsas de basura de aproximadamente 20 lbs c/u	2 bolsas de basura	Observación
<b>CANTIDAD BOTELLAS.</b>	<b>DE</b>	10, 24, y 50 botellas	50% botellas de plástico y 50 % de vidrio	Observación

### 3.1.5.2 Variables Dependientes.

VARIABLES DEPENDIENTES	DESCRIPCIÓN	INDICADOR	INSTRUMENTOS
<b>PORCENTAJE DE ERROR EN LA DETECCIÓN DE BOTELLAS.</b>	Número de botellas que ingresan vs Número de botellas que son detectadas.	Error (%)	software
<b>PORCENTAJE DE ERROR EN LA CLASIFICACIÓN DE BOTELLAS.</b>	Número de botellas que son detectadas vs Número de botellas que son clasificadas.	Error (%)	software

### 3.1.6 Procedimiento.



### **3.1.7 Diseño del prototipo.**

#### **3.1.7.1 Software del prototipo.**

La parte del software del prototipo hace referencia al entrenamiento del modelo con las diferentes herramientas; La realización del código para poder realizar las detecciones y finalmente la realización del código que controla la velocidad del motor de la banda transportadora y el código que acciona el sistema de clasificación.

##### **3.1.7.1.1 Entrenamiento del modelo.**

Como se mencionó en apartados anteriores, el entrenamiento del modelo YOLOv5s utilizado en este proyecto se realizó mediante la herramienta Google Colab. Se optó por esta herramienta debido a sus capacidades de GPU disponibles de forma gratuita en la nube para los usuarios de internet, y por no contar con una computadora con especificaciones lo suficientemente buenas para el entrenamiento.

###### **3.1.7.1.1.1 Parámetros de entrenamiento.**

Utilizándose un set de 2479 imágenes, dividido en 2181 imágenes de entrenamiento y 298 imágenes de validación, mediante experimentación se concluyó que el mejor entrenamiento se obtuvo con los siguientes parámetros:

- BatchSize= 16
- Épocas= 250
- Tamaño de imágenes= 416x416

###### **3.1.7.1.2 Código para las detecciones.**

El código para la detección de las botellas se realizó en el entorno PyCharm mediante lenguaje Python. En este código se carga el modelo entrenado previamente en la nube cuya nombre y extensión es “botellas5.pt”. Es importante señalar que las detecciones se realizan en cuanto las botellas se posicionan en el centro de la recámara de detección. El código completo, así como su explicación, se puede encontrar en la sección de anexos.

###### **3.1.7.1.3 Código del motor de la banda transportadora.**

Este código se realiza con el fin de bajar las revoluciones por minuto del motor implementado en la banda, mediante micropython en el IDE arduino y con la utilización del

módulo L298N se consigue reducir las revoluciones desde 6250 rpm a 27 rpm. Este valor fue el óptimo para que las botellas puedan ser detectadas que en movimiento.

Como punto importante se debe recalcar que la velocidad a la que gira el motor puede ser mayor siempre y cuando se cuente con una tarjeta gráfica de buenas características, ya que esta juega un papel importante en la velocidad de las detecciones. El código completo implementado para reducir las RPM se puede encontrar en la sección de anexos.

#### **3.1.7.1.4 Código del sistema de clasificación.**

El sistema de clasificación corresponde al diseño de un pistón, por lo tanto, se debe aclarar que el pistón funciona en sincronización con las detecciones, siendo que cada vez que se realiza una detección el sistema clasificador espera 4.5 segundos para activar el pistón clasificador. Este pistón permite que las botellas sean clasificadas del resto de la basura. El código implementado para este sistema se realiza en arduino sobre un servomotor FUTABA S3003 de 4,1 Kg/cm DE 180<sup>a</sup>, se configura el servomotor para que realice un movimiento de 60 grados, este giro es suficiente para arrojar las botellas mediante el pistón hacia un costado de la banda. El código completo del sistema clasificador se puede encontrar en anexos.

#### **3.1.8 Parte física del prototipo.**

La parte física del proyecto corresponde a la elaboración de dos estructuras por separado, en primer lugar, se construye la mesa vibratoria y en segundo lugar la banda transportadora, los dos elementos al final se unen para formar una sola máquina funcional.

##### **3.1.8.1 Mesa vibratoria.**

La mesa vibratoria es la parte del proyecto que se encarga de dispersar los residuos de basura, como su nombre lo indica se trata de una mesa con un motor de vibración acoplado, esto permite que la mesa vibre de forma constante para que la basura colocada en ella se disperse, tendiendo a reducir los cúmulos de basura que pueden ser perjudiciales para el funcionamiento óptimo de la detección en la banda transportadora.

La mesa vibratoria es de 1,30 m de largo, 60 cm de ancho y 80 cm de Alto. La mesa implementada se puede apreciar en la Figura 8.



Figura 8. Mesa Vibratoria.

### **3.1.8.1 Materiales de la mesa vibratoria.**

Este elemento es realizado en su totalidad con material MDF de 9 mm de grosor, escogido por sus propietarios resistentes y económicamente accesible. Para asegurar las partes por separado de la mesa se usan tornillos de la marca mamut para gypsum de 1 pulgada y arandelas mamut de 1 cm de diámetro.

### **3.1.8.2 Motor acoplado.**

El motor acoplado a la mesa vibratoria es un motor obtenido de una lavadora. Para lograr su función vibratoria, se construye una pieza metálica que es acoplada en el eje del motor, el peso de la pieza metálica permite que el eje del motor vibre con mayor fuerza. Posteriormente, este motor es acoplado en la parte inferior de la mesa y asegurado a través de pernos de 3 pulgadas de longitud y 1 cm de diámetro. El motor acoplado se puede observar en la Figura 9.



Figura 9. Motor acoplado.

### **3.1.8.3 Banda transportadora de basura.**

La banda transportadora es el segundo elemento en ser construido, probando previamente el funcionamiento de la mesa vibratoria se procede a estudiar el mejor diseño para la banda transportadora. La banda tiene 1,70 m de largo, 40 cm de ancho y 70cm de alto.

#### **3.1.8.3.1 Materiales de construcción de la banda transportadora.**

##### **3.1.8.3.1.1 Base.**

La base al igual que la mesa vibratoria es realizada sobre MDF de 9 mm, esto con el fin de dar firmeza a la estructura principal que contiene a los rodillos y a la banda como tal. La base cuenta con 3 patas de soporte de ambos lados y una estructura central alrededor de toda la base con el fin de evitar el movimiento brusco de la estructura a causa del motor instalado en ella.

Adicionalmente, en la base se realizan 5 perforaciones de 3 cm de diámetro para colocar las puntas de los rodillos de soporte y los dos rodillos principales de movimiento. La estructura de la base se puede apreciar en la Figura 10.



Figura 10. Base de la banda transportadora con los rodillos instalados.

##### **3.1.8.3.1.2 Rodillos:**

Los rodillos tienen 46 cm de largo y son contruidos de dos materiales. Las puntas son contruidas de madera, esto permite el acople de los rodillos sobre la base sin que se provoque algún desgaste por fricción. El cuerpo de los rodillos se fabrica sobre tubo PVC liso de 2 pulgadas, este proceso se realiza sobre los cinco rodillos, dentro de estos cinco rodillos existen dos rodillos llamados de movimiento, los cuales se pueden observar en la Figura 11. Estos poseen adicionalmente tiras de 1 cm de ancho, contruidas de lija para madera de número 80, está para facilitar el arrastre de la cinta que se usa para la banda transportadora.



Figura 11. Rodillos de movimiento.

### 3.1.8.3.1.3 Recámara de detección:

La recámara de detección es el compartimiento donde se instala la cámara y se crean las condiciones de luz adecuadas para que la detección de las botellas sea óptima, su estructura se observa en la Figura 12. La recama es realizada sobre material MDF de 6 mm de grosor. Para esta estructura se usa MDF más delgado para disminuir el peso sobre la base de la banda transportadora y por qué una estructura demasiado robusta no es necesario en este caso. Sus dimensiones son de 45 cm de largo, 45 cm de alto y 39 cm de ancho



Figura 12. Recamara de detección.

Adicionalmente, la recámara de detección tiene instaladas dos cortinas de color negro, construidas de plástico y cartulina para evitar que la luz salga de la recámara hacia el exterior y se pierda iluminación dentro de la misma, ver Figura 13. Por otro lado, las luces utilizadas son 4 tiras de 3 luces LED, éstas tiras funcionan con una tensión de 110v. En la parte superior de la recámara se realiza un orificio rectangular con las dimensiones de la cámara a utilizar (celular), de esta forma el celular se coloca en la superficie de forma estable.



Figura 13. Cortinas de la recamara de detección.

#### **3.1.8.3.1.4 Cinta transportadora:**

La cinta transportadora es realizada sobre material de tela Jean color negro, posee unas dimensiones de 35 cm de ancho por 3 m de largo. Su color se escoge debido a que resalta la mayoría de las botellas transparentes, lo que facilita su visualización en la cámara. La cinta instalada se puede observar en la Figura 14.



Figura 14. Cinta transportadora.

#### **3.1.8.3.1.5 Motor de la banda transportadora.**

El motor instalado en la banda transportadora es un motor DC conocido como motor de caja de cambios, ya que adicionalmente el motor DC tiene acoplado una serie de engranajes que logran una fuerza considerable poner, lo suficientemente fuerte para arrastrar el peso de la cinta transportadora. El motor tiene 10000 rpm, las cuales se reducen en primera instancia

mediante un mecanismo de poleas y en segundo lugar con código a través del microcontrolador ESP32 y el IDE de Arduino. Adicionalmente, se implementa el módulo L298N para poder controlar el motor con ayuda de la programación.

Para transferir el movimiento del motor a la cinta se usa un sistema de dos poleas construidas sobre MDF, teniendo la polea conductora un diámetro de 10 cm y la polea conducida un diámetro de 16 cm, la polea más pequeña que posee un diámetro de 10 cm es acoplada al motor y la más grande de un diámetro de 16 cm es acoplada a uno de los rodillos de movimiento. El motor instalado se puede apreciar en la Figura 15.



Figura 15. Poleas (izquierda) y Motor acoplado (Derecha).

La relación de velocidades entre las poleas se puede calcular a través de la fórmula (2), mientras que la relación de transmisión se calcula a través de la fórmula (3), con los siguientes datos:

- $n1 = 10000$  rpm (Número de vueltas de la polea conductora)
- $d1 = 10$  cm (Diámetro de la polea conductora)
- $n2 = ?$  (Número de vueltas de la polea conducida)
- $d2 = 16$  cm (Diámetro de la polea conducida)
- $i = ?$  (Relación de transmisión)

$$n1 \cdot d1 = n2 \cdot d2 \quad (2)$$

$$n2 = \frac{n1 \cdot d1}{d2} = \frac{10000 \text{ rpm} \cdot 10 \text{ cm}}{16 \text{ cm}} = 6250 \text{ rpm}$$

$$i = \frac{n1}{n2} = \frac{10000 \text{ rpm}}{6250 \text{ rpm}} = 1,6$$

Los valores obtenidos anteriormente reflejan en primer lugar que el número de vueltas del de motor pasa de 10000 rpm a 6250 rpm a través de la polea conducida, mientras que la

relación de transmisión indica que, por cada 1,6 vueltas de la polea conductora, la polea conducida dará 1 vuelta.

Como la reducción a través de la polea no fue suficiente y la construcción de una polea conducida de mayor diámetro tampoco, se optó por reducir las revoluciones a través de código tal como se indicó en apartados anteriores, logrando una reducción hasta 27 rpm, valor óptimo para que las botellas sean detectadas en movimiento.

### 3.1.8.3.1.6 Sistema de clasificación.

El sistema de clasificación cuenta con un pistón que se acciona gracias al ángulo de giro de un servomotor, como se indicó en apartados anteriores, este giro es de 60 grados. El sistema consta de una pieza circular acoplada al servomotor, a su vez esta pieza tiene acoplado una estructura metálica que transfiere el movimiento de expulsión y retracción al pistón. La estructura diseñada se puede apreciar en la Figura 16.

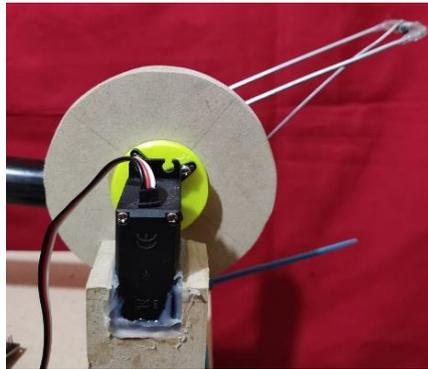


Figura 16. Servomotor adaptado al pistón.

Para el pistón se usa tubo PVC de 1 pulgada de diámetro y 45 cm de largo, para la estructura metálica se usa alambre de 3 mm de grosor, y finalmente el resto de la estructura es construida sobre MDF de 9 mm. La estructura del pistón se puede observar en la Figura 17.



Figura 17. Pistón clasificador.

En la Figura 18. se puede apreciar la parte de la banda transportadora del prototipo junto con el sistema de detección y el sistema de clasificación instalados, además de la mesa vibratoria en la parte derecha.



Figura 18. Mesa vibratoria y banda transportadora del prototipo.

## CAPÍTULO IV

### 4.1 RESULTADOS Y DISCUSIÓN.

Considerando una población infinita, el tamaño de la muestra considerada es de 900 datos para una densidad de 10 botellas, 24 botellas y 50 botellas en un área de 7800 cm<sup>2</sup> correspondiente a la mesa vibratoria.

Tabla 1. Estudio descriptivo de cada población del error en la detección.

Error_De_Detección	Cantidad_Botellas	Estadístico	
		Media	Mediana
	10 bot	Media	14,2778
		Mediana	10,0000
		Mínimo	,00
		Máximo	40,00
	24 bot	Media	8,8889
		Mediana	8,3333
		Mínimo	,00
		Máximo	25,00
	50 bot	Media	10,7844
		Mediana	10,0000
		Mínimo	,00
		Máximo	18,00

Tabla 2. Estudio descriptivo de cada población del error en la clasificación.

Error_De_Clasificacion	Cantidad_Botellas	Estadístico	
		Media	Mediana
	10 bot	Media	7,1111
		Mediana	10,0000
		Mínimo	,00
		Máximo	20,00
	24 bot	Media	3,5185
		Mediana	4,1667
		Mínimo	,00
		Máximo	12,50
	50 bot	Media	4,6468
		Mediana	4,5455
		Mínimo	,00
		Máximo	12,00

Como se puede observar en la Tabla.1 y Tabla.2, si bien los errores porcentuales en la detección y clasificación con 50 botellas es menor que con 10 botellas, estos no son

resultados concluyentes a priori. Esto debido a que el número de botellas detectadas y clasificadas erróneamente con 50 es mayor que con 10 botellas, sin embargo, porcentualmente representan un valor menor. Por lo tanto, se ha considerado el número de botellas no detectadas y no clasificadas como variable de estudio,

## 4.2 HIPÓTESIS

Las hipótesis se plantean con la cantidad de botellas no detectadas y no clasificadas, en vez del error porcentual, ya que como se mencionó anteriormente, una comparación entre la media de los errores porcentuales no sería lógica, pues las poblaciones no tienen una misma cantidad de botellas.

### 4.2.1 Hipótesis 1.

- $H_0$ : La densidad de botellas no afecta a la cantidad de botellas no detectadas, es decir, las medias  $\mu_1 = \mu_2 = \mu_3$ .

Donde:

- $\mu_1$  es la media de botellas no detectadas para una densidad de 10 botellas por área.
  - $\mu_2$  es la media de botellas no detectadas para una densidad de 24 botellas por área.
  - $\mu_3$  es la media de botellas no detectadas para una densidad de 50 botellas por área.
- $H_1$ : La densidad de botellas si afecta a la cantidad de botellas no detectadas, es decir, las medias  $\mu_i \neq \mu_j$ .

### 4.2.2 Hipótesis 2.

- $H_0$ : La densidad de botellas no afecta a la cantidad de botellas no clasificadas, es decir, las medias  $\mu_1 = \mu_2 = \mu_3$

Donde:

- $\mu_1$  es la media de botellas no clasificadas para una densidad de 10 botellas por área.
  - $\mu_2$  es la media de botellas no clasificadas para una densidad de 24 botellas por área.
  - $\mu_3$  es la media de botellas no clasificadas para una densidad de 50 botellas por área.
- $H_1$ : La densidad de botellas si afecta a la cantidad de botellas no clasificadas, es decir, las medias  $\mu_i \neq \mu_j$ .

## 4.3 PRUEBA DE NORMALIDAD DE MUESTRAS.

Hipótesis:

$H_0$  = los datos siguen una distribución normal

$H_1$  = los datos no siguen una distribución normal

Estadístico de prueba:

Como estadístico para comprobar la distribución normal de los datos se selecciona el test de Kolmogorv-Smirnov, ya que el número de datos de la muestra con la que se trabaja implica que  $n > 50$ .

Zona de rechazo:

Si  $p\text{-valor} < 0.05$ , se rechaza  $H_0$

Si  $p\text{-valor} \geq 0.05$ , se acepta  $H_0$  y se rechaza  $H_1$

Tabla 3. Prueba de normalidad para el número de botellas no detectadas y no clasificadas de las tres poblaciones.

	Cantidad_Botellas	Kolmogorov-Smirnov		
		Estadístico	gl	Sig.
Botellas_no_Detectadas	10 bot	,303	900	,000
	24 bot	,238	900	,000
	50 bot	,168	900	,000
Botellas_no_Clasificadas	10 bot	,294	900	,000
	24 bot	,274	900	,000
	50 bot	,232	900	,000

La Tabla.3 muestra que los datos de la cantidad de botellas no detectadas y no clasificadas no se distribuyen normalmente, por lo que para realizar las pruebas de hipótesis se debe aplicar un test no paramétrico.

#### 4.4 PRUEBA DE HIPÓTESIS

Para la prueba de hipótesis, dado que los datos no siguen una distribución normal, se aplica el test de Kruskal-Wallis, ya que se tiene dos variables analizables en tres casos diferentes, y además se tiene grupos independientes.

Tabla 4. Pruebas de hipótesis para la cantidad de botellas no detectadas y no clasificadas.

	Botellas_no_Detectadas	Botellas_no_Clasificadas
H de Kruskal-Wallis	1696,348	774,788
gl	2	2
Sig. asintótica	,000	,000

La Tabla.4 refleja un p valor  $< 0,05$  para las botellas no detectadas y las botellas no clasificadas, esto permite rechazar las hipótesis nulas y aceptar las hipótesis alternativas antes planteadas. Se concluye que la densidad de botellas si afecta a la cantidad de botellas no detectadas y no clasificadas, es decir, las medias  $\mu_i \neq \mu_j$ .

#### 4.5 PRUEBA DE GRUPOS.

La prueba de grupos se realiza con el fin de determinar entre que grupos existe diferencia, una vez comprobada la existencia de la misma con el test de Kruskal-Wallis en pasos anteriores.

Tabla 5. Prueba de Tukey para la cantidad de botellas no detectadas.

Cantidad_Botellas	N	Subconjunto		
		1	2	3
10 bot	900	1,4122		
24 bot	900		2,1067	
50 bot	900			5,3511
Sig.		1,000	1,000	1,000

Tabla 6. Prueba de Tukey para la cantidad de botellas no clasificadas.

Cantidad_Botellas	N	Subconjunto		
		1	2	3
10 bot	900	,7111		
24 bot	900		,8444	
50 bot	900			2,0856
Sig.		1,000	1,000	1,000

Con la Tabla.5 y la Tabla.6 se puede determinar que las medias del número de botellas no detectadas y no clasificadas de las tres poblaciones son todas diferentes entre sí, ya que ninguna media se ubica de bajo de otra en los subconjuntos de las tablas anteriores.

Para lograr una apreciación visual del análisis antes hecho se grafican las medias de la cantidad de botellas no detectadas y no clasificadas. La Fig.19 es de suma importancia para confirmar que el prototipo tiende a tener una mayor cantidad de botellas no detectadas con forme se aumenta la cantidad de botellas a ser procesadas, esto se evidencia a través de las medias obtenidas para las tres poblaciones, siendo la población uno la que presenta menor error. La población uno cuenta con una media de 1,41 botellas no detectadas, mientras que la población dos cuenta con una media de 2,11 botellas no detectadas, entre estos dos valores ya existe una diferencia, finalmente la población tres muestra una media más distante de 5,35 botellas no detectadas.

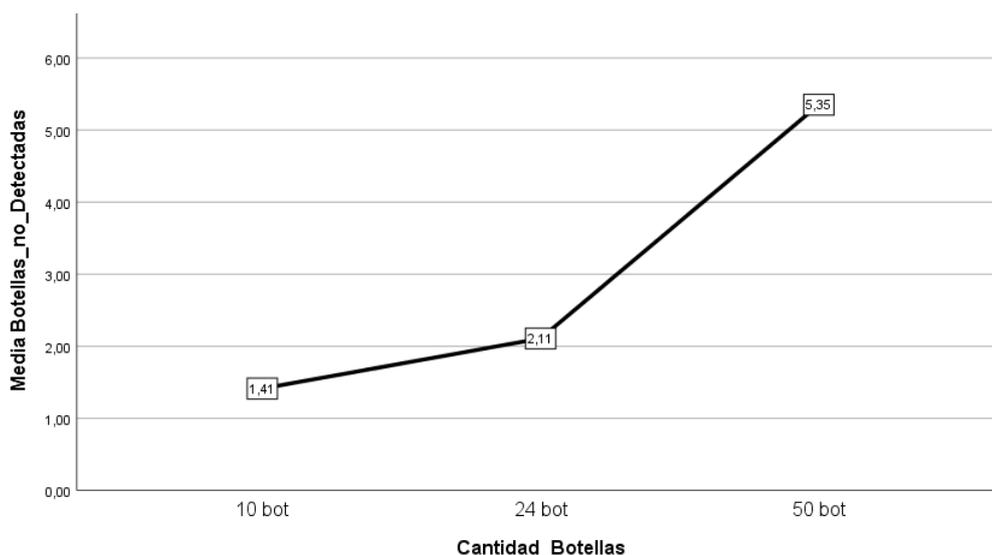


Figura 19. Media de la cantidad de botellas no detectadas de las tres poblaciones.

Una vez comprobado que el aumento del número de botellas que no se detectan se incrementa por el aumento de la cantidad de botellas a ser procesadas, se determina que este aumento se da debido a que la presencia de más botellas juntas en la banda transportadora moviéndose al mismo tiempo, vuelven al sistema lento debido a la baja capacidad de la tarjeta gráfica del computador sobre el que se ejecuta el modelo de detección.

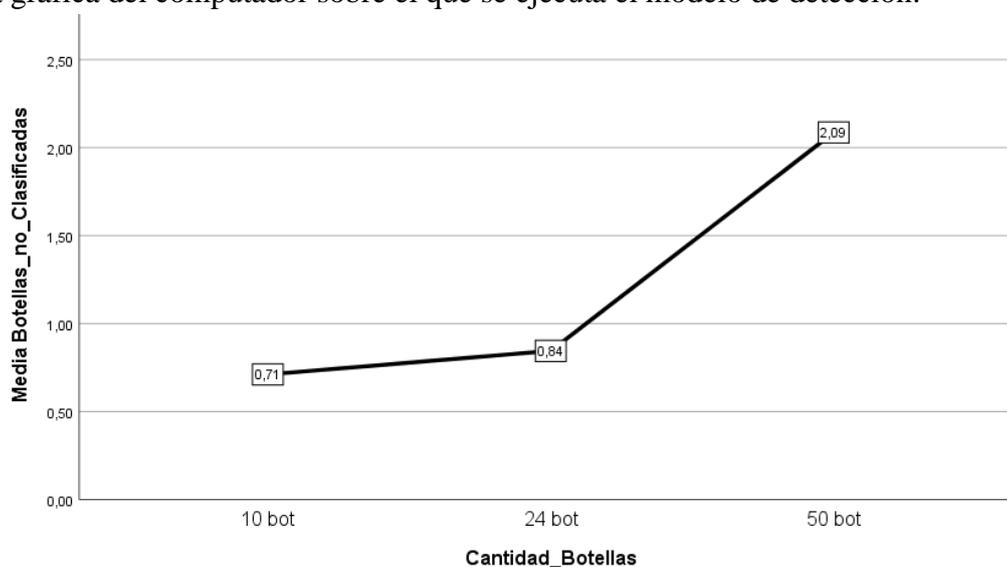


Figura 20. Media de la cantidad de botellas no clasificadas de las tres poblaciones.

El gráfico anterior demuestra que el prototipo también presenta una mayor cantidad de botellas no clasificadas conforme se aumenta la cantidad de botellas que ingresan desde la mesa vibratoria. Las medias obtenidas son 0,71, 0,84 y 2,09 para la población uno, dos y tres respectivamente, siendo la población uno la que presenta menor error.

Al igual que con el caso anterior, se determina que este aumento en la cantidad de botellas o clasificadas es producto de que la sincronización entre la detección y la clasificación no es óptima en todos los casos, esto debido a que la computadora utilizada tiende a ralentizarse

debido a los recursos limitados de la máquina y enviar tarde la señal para la activación del pistón clasificador en ciertos casos.

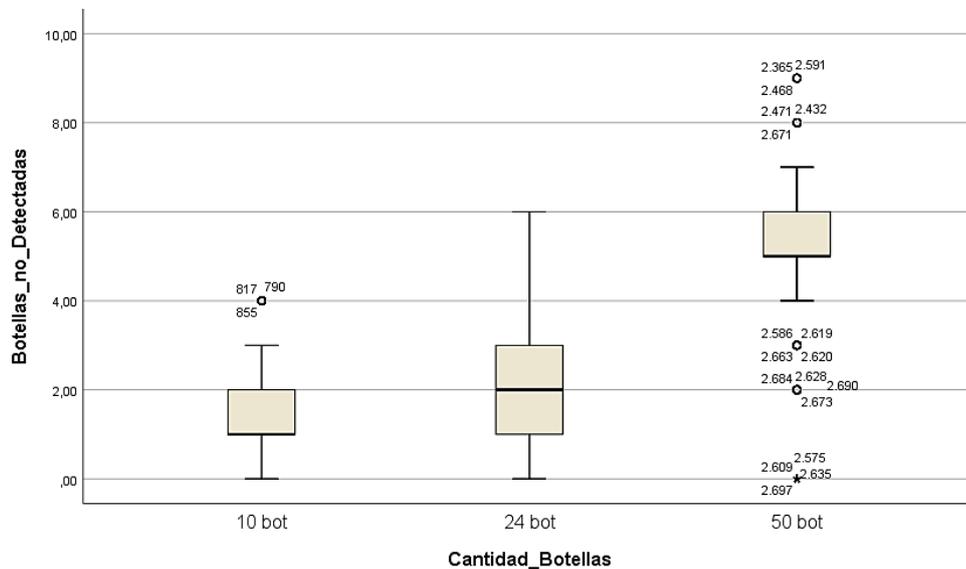


Figura 21. Diagrama de cajas para el numero de botellas no detectadas para las tres poblaciones.

Finalmente, se muestran los diagramas de cajas para la cantidad de botellas no detectadas y no clasificadas, para observar cómo se dispersan los datos. La Fig.21 muestra que para la población uno de 10 botellas los datos están mayormente concentrados por encima de la mediana de valor 1, apareciendo valores extraños de 4 botellas no detectadas en algunos casos. Por otro lado, para la población dos de 24 botellas se observa una concentración de los datos de manera equitativa a ambos lados de la mediana de valor 2. Finalmente, para la población tres de 50 botellas se observa que como valor extraño aparece el número 0, es decir, cuando se ingresan 50 botellas el prototipo rara vez logra detectar el 100%.

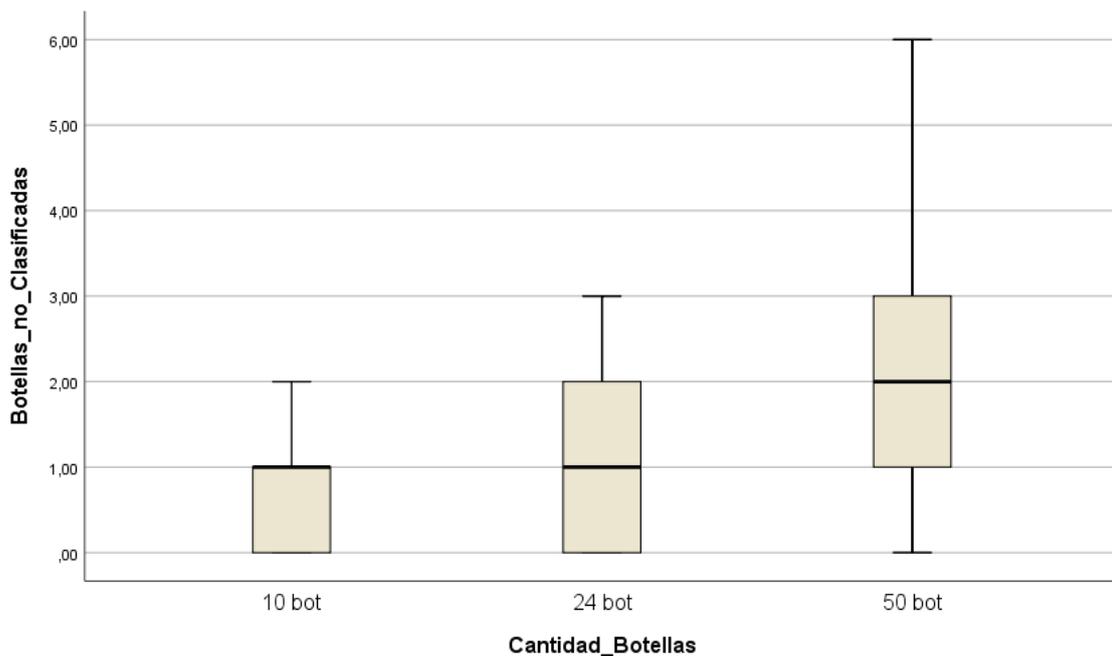


Figura 22 . Diagrama de cajas para el numero de botellas no clasificadas para las tres poblaciones.

La Fig.22 muestra para la población uno de 10 botellas que los datos están mayormente concentrados por debajo de la mediana de valor 1. Para la población dos de 24 botellas se observa una concentración de los datos de manera equitativa a ambos lados de la mediana de valor 1. Por último, para la población tres de 50 botellas se observa que la cantidad de botellas no clasificadas va desde 0 hasta 6 botellas en algunos casos,

## **CAPÍTULO V.**

### **5.1 CONCLUSIONES.**

En este proyecto se ha diseñado e implementado un prototipo para la clasificación de botellas de vidrio y de plástico. El prototipo fue sometido a 2700 pruebas para poder analizar su comportamiento en diferentes escenarios. El análisis del porcentaje de error en la detección y el porcentaje de error en la clasificación de cada una de las poblaciones permitieron comprender su funcionamiento, el prototipo presenta tendencia a cometer más errores en la detección y clasificación conforme sube el número de botellas que ingresan desde la mesa vibratoria.

La media del número de botellas no detectadas fue de 1,41 para 10 botellas, 2,11 para 24 botellas y 5,35 para 50 botellas, mientras que la media del número de botellas no clasificadas fue de 0,71 para 10 botellas, 0,84 para 24 botellas, 2,09 para 50 botellas. Con el análisis estadístico de aplicado se concluye que el prototipo diseñado posee un alto nivel de acierto en las detecciones y clasificaciones que realiza de las botellas que son procesadas.

### **5.2 RECOMENDACIONES.**

Cuando se trabaja con modelos de visión artificial como YOLOv5, siempre es factible realizar un estudio previo de los mismos, así como su aplicabilidad en diferentes ambientes reales, con el fin de determinar el mejor modelo para el uso requerido.

A pesar de contar con un modelo de visión artificial con un buen entrenamiento para el reconocimiento de objetos, siempre es necesario contar con equipos como computadoras que posean buenas características, que permita una ejecución rápida de los modelos sobre su arquitectura. Por ejemplo, la utilización de computadoras con memoria gráfica de bajas características significaría detecciones lentas e imprecisas.

Para el entrenamiento de los modelos de visión artificial, se recomienda que, si no se cuenta con computadoras potentes, se puede recurrir a recursos en la nube como google Colab para entrenar de forma más rápida y sencilla un modelo de visión artificial.

## **CAPÍTULO VI.**

### **6.1 PROPUESTA.**

El prototipo diseñado, a pesar de tener un funcionamiento bastante bueno, siempre puede ser mejorado desde diferentes perspectivas, siendo una de ellas el sistema de clasificación. Por ende, un análisis de nuevos métodos, más rápidos y sencillos para clasificar las botellas una vez se encuentren moviéndose a través de la banda transportadora, puede ser clave para volver el diseño mucho más aplicable a escenarios reales de mayor demanda en la basura a ser procesada.

## BIBLIOGRAFÍA

- [1] Alianza Basura cero Ecuador, “Informe desechos plásticos Alianza Basura Cero Ecuador,” Ecuador, Apr. 2022.
- [2] M. Solíz, *La partida 3915 Importación de desechos plásticos en Ecuador*, 1<sup>a</sup> ed. Quito: Universidad andina simón bolívar, 2021.
- [3] J. Sáez, “Impacto del plástico en el medio ambiente.” Accessed: Aug. 06, 2022. [Online]. Available: <https://blogs.ucjc.edu/cc-transporte-logistica/2019/07/impacto-del-plastico-en-el-medio-ambiente/>
- [4] eluniversocom, “Ecuador genera 375 mil toneladas de residuos sólidos urbanos al año, pero solo recicla el 4% de estos desechos,” Dec. 30, 2022. Accessed: Jul. 09, 2022. [Online]. Available: <https://www.eluniverso.com/noticias/2020/12/30/nota/9111586/ecuador-genera-375-mil-toneladas-residuos-solidos-urbanos-ano-solo/>
- [5] L. Rouhiainen, Ed., *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*, 1<sup>a</sup> ed. España: Alienta Editorial, 2018.
- [6] Oscar Olalla, “Redes Neuronales artificiales: Qué son y cómo se entrenan,” Xeridia. Accessed: Jul. 08, 2022. [Online]. Available: <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>
- [7] UNIR, “La visión artificial: una revolución en la Industria 4.0,” Ingeniería y tecnología. Accessed: Jun. 10, 2022. [Online]. Available: <https://www.unir.net/ingenieria/revista/vision-artificial/>
- [8] Sandoval. Cath, “Reconocimiento de imágenes: Conceptos básicos,” LISA Insurtech. Accessed: Jul. 04, 2022. [Online]. Available: <https://www.lisainsurtech.com/es/2021/11/09/reconocimiento-de-imagenes-conceptos-basicos/>
- [9] J. Castillo, “Sistema seleccionador de botellas mediante visión artificial en Raspberry pi,” Universidad politécnica de valencia, Valencia, España, 2020.
- [10] G. Luna and K. Vasconez, “Desarrollo de una red neuronal convolucional para la clasificación de botellas plásticas en una empresa envasadora de bebidas gaseosas de la ciudad de milagro,” Universidad Estatal de Milagro, Milagro, Ecuador, 2020.
- [11] M. Nuño *et al.*, “Implementation and integration of image processing blocks in a real-time bottle classification system,” *Sci Rep*, vol. 12, no. 1, Dec. 2022.
- [12] A. Perez, J. Vega, and R. Kelmer, “Reconocimiento automático de botellas plásticas utilizando redes neuronales artificiales,” *ResearchGate*, Oct. 2018.
- [13] D. Williams and G. Flores, “Diseño de un sistema automático clasificador de envases de bebidas previo al reciclaje.” Universidad tecnológica equinoccial, Quito, Ecuador, 2017.
- [14] Z. Wang, B. Peng, Y. Huang, and G. Sun, “Classification for plastic bottles recycling based on image recognition,” *Waste Management*, vol. 88, Apr. 2019.
- [15] V. Puertas, P. Montalvo, D. Ordóñez, and J. Rojas, “Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura (Internet of Things and Artificial Vision, Performance and Applications: Literature Review),” vol. 7, no.

- 1, pp. 244–256, Feb. 2017, [Online]. Available: <http://ingenieria.ute.edu.ec/enfoqueute/>
- [16] F. Ruiz, Cazorla Miguel, A. Galipienso, O. Colomina, and M. Lozano, *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Madrid, España: Paraninfo, 2003.
- [17] J. García, “Sistemas de símbolos en inteligencia artificial,” Universidad Autónoma de Querétaro, Querétaro, 2008.
- [18] L. Barrera, “FUNDAMENTOS HISTÓRICOS Y FILOSÓFICOS DE LA INTELIGENCIA ARTIFICIAL,” *Revista de Investigación y Cultura*, vol. 1, no. 1, Jul. 2012, [Online]. Available: <http://www.redalyc.org/articulo.oa?id=521752338014>
- [19] L. Carabantes, “Inteligencia artificial: condiciones de posibilidad técnicas y sociales para la creación de máquinas pensantes,” universidad complutense de madrid facultad de filosofía, Madrid, 2014.
- [20] E. Manrique, “Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo,” *Revista Ibérica de Sistemas e Tecnologías de Informação (risti)*, pp. 586-599., 2020.
- [21] J. Bobadilla, *Machine learning y deep learning: usando Python, Scikit y Keras*, Ediciones de la U. Bogotá: Ra-ma, 2021.
- [22] C. Alba, “DEEP LEARNING,” Universidad de Sevilla, Sevilla, 2019.
- [23] M. Vahid and S. Raschka, *Python machine learning*. España: Marcombo, 2020.
- [24] machinelearningenespanol.com, “Qué es el Machine Learning y Cómo Funciona.” [Online]. Available: <https://machinelearningenespanol.com/2021/07/21/que-es-machine-learning-como-funciona/>
- [25] Bustamante. Ernesto, *El sistema nervioso: desde las neuronas hasta el cerebro humano*. Universidad de Antioquia, 2007.
- [26] J. Ramírez and M. Chacón, “Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década.,” *RIEE&C*, vol. 9, no. 1, Jul. 2011.
- [27] E. Serna, *DESARROLLO E INNOVACIÓN EN INGENIERÍA Editorial IAI*. Medellín – Antioquia: Editorial IAI, 2017. [Online]. Available: <http://fundacioniai.org>
- [28] Á. Artola, “Clasificación de imágenes usando redes neuronales convolucionales en Python,” Universidad de Sevilla, Sevilla, 2019.
- [29] P. Rodríguez and S. Alesanco, “Aplicación de redes neuronales convolucionales y recurrentes al diagnóstico de autismo a partir de resonancias magnéticas funcionales,” thesis, Universidad Politécnica de Madrid, Madrid, España, 2018.
- [30] M. Massiris, C. Delrieux, and J. Fernández, “Detección de equipos de protección personal mediante red neuronal convolucional YOLO,” pp. 1022–1029, Mar. 2020, doi: 10.17979/spudc.9788497497565.1022.
- [31] D. Calvo, “Red Neuronal Convolucional CNN.” [Online]. Available: [https://docs.ultralytics.com/yolov5/tutorials/neural\\_magic\\_pruning\\_quantization/#annotate-cli](https://docs.ultralytics.com/yolov5/tutorials/neural_magic_pruning_quantization/#annotate-cli)
- [32] A. Golfe San Martín, “Diseño de un modelo compuesto de arquitecturas de red neuronal convolucional y recurrente para descripción de vídeo en entornos outdoor/indoor,” M.S. thesis, Universidad Politécnica de Valencia, Valencia, España, 2020.

- [33] “Red Neuronal Recurrente – RNN.” [Online]. Available: <https://www.diegocalvo.es/red-neuronal-recurrente/>
- [34] ultralytics, “Neural Magic’s DeepSparse.” [Online]. Available: [https://docs.ultralytics.com/yolov5/tutorials/neural\\_magic\\_pruning\\_quantization/#how-does-deepsparse-achieve-gpu-class-performance](https://docs.ultralytics.com/yolov5/tutorials/neural_magic_pruning_quantization/#how-does-deepsparse-achieve-gpu-class-performance)
- [35] I. García and V. Caranqui, “LA VISIÓN ARTIFICIAL Y LOS CAMPOS DE APLICACIÓN,” *Tierra Infinita*, vol. 1, pp. 98–108, Dec. 2015, doi: 10.32645/26028131.76.
- [36] Espressif, “Module ESP32.” Accessed: Oct. 04, 2023. [Online]. Available: <https://www.espressif.com/en/products/modules>
- [37] Edison Del Rosario, “1.4 IDE Arduino con ESP32.” [Online]. Available: <http://blog.espol.edu.ec/girni/ide-arduino-con-esp32/>
- [38] Google, “Colab.” Accessed: Oct. 04, 2023. [Online]. Available: [https://colab.research.google.com/?hl=es#scrollTo=Nma\\_JWh-W-IF](https://colab.research.google.com/?hl=es#scrollTo=Nma_JWh-W-IF)
- [39] ultralytics, “YOLOv5.” Accessed: Oct. 04, 2023. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [40] S. Rozada, “Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning,” M.S. thesis, Universidad De Valladolid, Valladolid, España, 2021.
- [41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [42] ultralytics, “Tips for Best Training Results YOLOv5.” Accessed: Oct. 04, 2023. [Online]. Available: [https://docs.ultralytics.com/yolov5/tutorials/tips\\_for\\_best\\_training\\_results/](https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/)

## ANEXOS

### Código del sistema de Detección.

```
# Se importan las librerías a utilizar
import torch
import cv2
import numpy as np
import serial
import time

#Se inicia comunicación serial

ser = serial.Serial('COM3', 115200, timeout=1)
time.sleep(2)

# Se declara un variable que permite enviar una señal
# al sistema clasificador para avanzar
w = 'w'

# Se lee el modelo entrenado indicando su ubicación
model = torch.hub.load('ultralytics/yolov5', 'custom',
                       path
                       ='C:/Users/USUARIO/Desktop/Codigos_de_Deteccion_de_Botellas/Cod_yolov5/yolov5s/
                       model/botellas14.pt')

#video captura
cap = cv2.VideoCapture(2)
bandera = 0

while True:
    #Se realiza la lectura de la videocaptura
    ret, frame = cap.read()

    #Se extrae el ancho y alto de la ventana de detección
    alto, ancho, c=frame.shape

    #Se realiza las detecciones
    detect = model(frame, size = 416)

    #Se extrae la información
    info = detect.pandas().xyxy[0].to_dict(orient="records")
```

```

#Se pregunta si hay predicciones
if len(info) != 0:
    for result in info:
        conf=result['confidence']
        # Se establece un nivel de confianza para las detecciones (conf)
        if conf >= 0.45:
            # Xi
            xi = int(result['xmin'])
            # Yi
            yi = int(result['ymin'])
            # Xf
            xf = int(result['xmax'])
            # Yf
            yf = int(result['ymax'])

            # Se dibuja un rectángulo alrededor de la botella
            cv2.rectangle(frame, (xi, yi), (xf, yf), (255, 0, 0), 2)

            # Se calcula el punto medio de la detección
            xm = (xi + xf) / 2

            #Se calcula el punto medio de la ventana
            pmin = (ancho / 2) - 20
            pmax = (ancho / 2) + 20

            #Se envía la señal al sistema clasificador si se realiza una detección
            if xm >= pmin and xm <= pmax and bandera == 0:
                bandera = 1
                print('Enviando comunicación serial')
                # Enviamos w
                ser.write(w.encode('ascii'))
                #Se imprime la confiabilidad en la consola
                print(conf)

#si no detecta se reinicia la bandera
elif len(info) == 0:
    bandera = 0

#Se muestran FPS
cv2.imshow('Detector de Botellas', frame)

#Se escribe en el teclado ESC para salir

```

```
t = cv2.waitKey(5)
if t == 27:
    break

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

## **Código del sistema de clasificación.**

```
//Se agregan las librerías para controlar el servomotor
#include <ESP32Servo.h>

//Se declara una variable que ayuda a controlar el servo
Servo servo;
int pinServo=12;

//Se instancia una variable para hacer la comunicación serial
int input;

void setup()
{
  //Se indica con que pin de la placa EP32 se trabajara
  //en el servo
  servo.attach(pinServo, 500, 2500);

  //Se Inicializa la comunicación serial
  Serial.begin(115200);
}

void loop() {
  if(Serial.available() > 0){
    input = Serial.read();

    if(input == 'w'){
      delay(4500);
      //Se mueve el servo a 5 grados posición inicial
      servo.write(5);
      //Se espera 1.5 segundos
      delay(900);
      //Se mueve el servo a 130 grados posición final
      servo.write(130);
      //Se espera 1.5 segundos
      delay(900);
    }
  }
}
```

## **Código del motor de la banda transportadora.**

```
//Este código controla la velocidad de la caja de cambios de la banda.
```

```
//Se elige el PIN con el que se trabajara.
```

```
const int ledPin = 14;
```

```
//Se eligen dos pines para controlar el motor con el módulo L298N.
```

```
int IN1 = 12;
```

```
int IN2 = 13;
```

```
// Se configuran las características PWM
```

```
const int freq = 5000;
```

```
const int ledChannel = 0;
```

```
const int resolution = 8;
```

```
void setup(){
```

```
    // Se configuran las funcionalidades LED PWM
```

```
    ledcSetup(ledChannel, freq, resolution);
```

```
    // Se conecta el canal al GPIO a controlar.
```

```
    ledcAttachPin(ledPin, ledChannel);
```

```
    // Se configura el sentido del giro del motor.
```

```
    pinMode(IN1, OUTPUT);
```

```
    pinMode(IN2, OUTPUT);
```

```
    digitalWrite(IN1, LOW);
```

```
    digitalWrite(IN2, HIGH);
```

```
}
```

```
void loop(){
```

```
    ledcWrite(ledChannel, 210);
```

```
}
```