

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES



Proyecto #50

Clasificador Inteligente de Residuos con Visión Artificial

MATERIA : Programación Ensamblador

DOCENTE: Ing. Flores Guzmán Valentín Victor

AREA : Programación

CATEGORIA : Intermedio

INTEGRANTES:

Saby Tuco Javier

Tiñini Leon Fernando Junior

Andrade Iriarte Ezequiel Matias

Mendia Parada Enrique

Arce Choque Pamela

Ing. Redes y Telecomunicaciones

Ing. Redes y Telecomunicaciones

Ing. Redes y Telecomunicaciones

Ing. Redes y Telecomunicaciones

Ing. Sistemas

SANTA CRUZ DE LA SIERRA – BOLIVIA

JULIO - 2025

Contenido

1.	Introducción.....	3
2.	Resumen del Proyecto.....	4
3.	Planteamiento del Problema.....	7
4.	Formulación de Objetivos	7
4.1.	Objetivo General: Diseñar e implementar un sistema inteligente basado en visión artificial que permita la clasificación automática de residuos sólidos, con el fin de optimizar la gestión de desechos y promover la cultura del reciclaje en Bolivia.....	7
4.2.	Objetivos Específicos:.....	8
5.	Desarrollo:	8
5.1.	Titulo del proyecto: Clasificador inteligente de residuos con visión artificial para la gestión sostenible de desechos en Bolivia.....	8
5.2.	Diagrama de bloques :	9
5.3.	Función de Cada Etapa del Diagrama de Bloques.....	9
5.4.	Estructura completa del sistema	11
5.5.	Descripción del funcionamiento de la estructura del sistema	11
5.6.	Descripción de la construcción del prototipo	12
6.	Marco Teórico.....	14
6.1.	Clasificación de residuos.....	15
6.2.	Visión artificial	15
6.3.	Arduino y sistemas embebidos.....	15
6.4.	Sensores de peso y servomecanismos	15
6.5.	Educación ambiental y tecnología.....	16
7.	Marco Metodológico.....	16
7.1.	Enfoque metodológico	16
7.2.	Fases del desarrollo	16
8.	Conclusiones.....	17
8.1.	Recomendaciones.....	18
9.	Bibliografía	20
10.	Anexo A – Código fuente del sistema de clasificación de residuos con visión artificial: CODIGO:	

1. Introducción

Bolivia enfrenta grandes desafíos en la gestión de residuos sólidos, con más de 6.000 toneladas generadas a diario y menos del 10% recicladas adecuadamente. Esta situación se debe, en parte, a la falta de infraestructura, automatización y conciencia ambiental. En contraste con países donde el reciclaje es altamente tecnificado, en Bolivia la clasificación aún es manual, ineficiente y riesgosa para los trabajadores.

En este contexto, el presente proyecto propone el desarrollo de un clasificador inteligente de residuos basado en visión artificial. Mediante una cámara conectada a un sistema de inteligencia artificial, el prototipo reconocerá el tipo de residuo cuando una persona acerque su bolsa de basura, y automáticamente activará la apertura del basurero correspondiente (plástico, metal, papel u orgánico). Esta solución no solo busca optimizar la separación desde el origen, sino también promover la conciencia ciudadana sobre la importancia del reciclaje, facilitando la participación activa en una gestión sostenible de los desechos.

2. Resumen del Proyecto

Este proyecto tiene como objetivo diseñar e implementar un sistema inteligente de clasificación de residuos utilizando visión artificial y componentes electrónicos de bajo costo. El sistema contará con varios basureros diferenciados por colores y etiquetas para residuos **orgánicos, inorgánicos, metálicos y plásticos**. A través del uso de una **cámara**, un modelo de **inteligencia artificial** entrenado con imágenes de distintos tipos de desechos podrá identificar el residuo cuando el usuario acerque la bolsa o el objeto a la cámara. En función del tipo detectado, el sistema abrirá automáticamente la tapa del basurero correspondiente mediante un mecanismo controlado por un **microcontrolador Arduino**.

Este clasificador de residuos no solo facilita el reciclaje desde el origen, sino que también busca promover la conciencia ambiental en hogares, escuelas y espacios públicos, apoyando así una gestión más sostenible de los desechos.

Componentes sugeridos para el sistema:

Arduino UNO o Mega: Para el control central del sistema.

Raspberry Pi (opcional): Ideal para procesar las imágenes si se usa un modelo más complejo de IA.

Cámara USB o cámara Pi (si se usa Raspberry): Para capturar imágenes de los residuos.

Servomotores o actuadores lineales: Para abrir automáticamente las tapas de los basureros según el tipo de residuo.

Módulo de relé (si se usan motores más potentes): Para controlar cargas mayores con

seguridad.

Display OLED o LCD: Para mostrar información sobre el tipo de residuo detectado o alertas.

Altavoz o buzzer: Para emitir sonidos de confirmación o advertencia.

Fuente de alimentación de 5V o batería recargable: Para alimentar el sistema completo.

Chasis de contenedores de basura con compuertas diferenciadas por colores y etiquetas.

Este proyecto es replicable y escalable, pudiendo adaptarse para centros educativos, empresas, plazas o espacios públicos, y representa una solución innovadora para la problemática de la gestión de residuos en Bolivia.

Abstract (English version)

This project proposes the design of an intelligent waste classifier using computer vision to contribute to the sustainable management of waste in Bolivia. Given the country's lack of infrastructure and environmental awareness, a system is proposed that can identify waste through a camera and an artificial intelligence model, which, upon detecting the type of trash, automatically triggers the opening of the corresponding bin. The implementation includes the use of Arduino, servomotors, and an image training platform. This system is expected to encourage waste separation at the source, reduce manual labor, and improve recycling efficiency, making it a replicable tool in educational centers and public space

Palabras clave:

inteligencia artificial, clasificación de residuos, visión artificial, reciclaje automatizado, Arduino, sostenibilidad ambiental, machine learning

Contexto

Este proyecto surge como una iniciativa de carácter universitario con el propósito de que el estudiante desarrolle competencias en el diseño, implementación y análisis de sistemas inteligentes aplicados a problemas sociales reales. En este caso, se enfoca en la problemática del manejo inadecuado de los residuos sólidos, específicamente en la falta de conciencia y educación ambiental sobre el reciclaje en Bolivia.

La finalidad del proyecto no solo es técnica, sino también formativa: busca que el estudiante comprenda el impacto del comportamiento ciudadano en la gestión de los desechos, y cómo la tecnología puede convertirse en una herramienta clave para fomentar hábitos sostenibles. A través del desarrollo de un sistema basado en inteligencia artificial y visión por computadora, el estudiante tendrá la oportunidad de aplicar conocimientos de programación, electrónica, diseño de sistemas embebidos y análisis de datos, al tiempo que reflexiona sobre su rol como agente de cambio en la sociedad.

Además, este proyecto sirve como base para el análisis interdisciplinario entre tecnología, medioambiente y educación, generando soluciones que no solo resuelvan un problema técnico, sino que también promuevan la participación ciudadana, la inclusión digital y el respeto por el entorno. Se espera que esta experiencia académica contribuya a formar profesionales capaces de liderar proyectos tecnológicos con impacto social y ambiental positivo.

3. Planteamiento del Problema

En Bolivia, la gestión de residuos sólidos urbanos presenta serias deficiencias, tanto a nivel de infraestructura como de comportamiento ciudadano. Uno de los problemas más frecuentes es que los desechos son descartados sin ningún tipo de separación previa: todos los residuos, sean orgánicos, plásticos, metálicos o papel, son depositados juntos dentro de una misma bolsa. Estas bolsas son luego acumuladas en esquinas o puntos de recolección informales, donde son retiradas por camiones recolectores de empresas privadas contratadas por los municipios.

Este sistema de manejo de basura, altamente centralizado y poco eficiente, ha generado múltiples dificultades. Por un lado, los vertederos se llenan rápidamente debido a la falta de clasificación, lo que acelera su colapso y genera conflictos por la gestión y ubicación de nuevos espacios de disposición final. Por otro lado, se pierden oportunidades de reciclaje y reutilización de materiales valiosos que podrían reducir significativamente el volumen de residuos enviados a los botaderos.

La ausencia de políticas efectivas de separación en origen, sumada a una baja conciencia ciudadana sobre la importancia del reciclaje, agrava la situación. La automatización del proceso de clasificación de residuos representa una solución innovadora y urgente. Desarrollar un sistema inteligente que identifique y clasifique automáticamente los residuos podría optimizar la cadena de gestión, reducir la presión sobre los vertederos y fomentar una cultura más responsable y sostenible en torno al manejo de la basura.

4. Formulación de Objetivos

4.1. Objetivo General:

Diseñar e implementar un sistema inteligente basado en visión artificial que permita la clasificación automática de residuos sólidos, con el fin de optimizar la gestión de desechos y promover la cultura del reciclaje en Bolivia.

4.2. Objetivos Específicos:

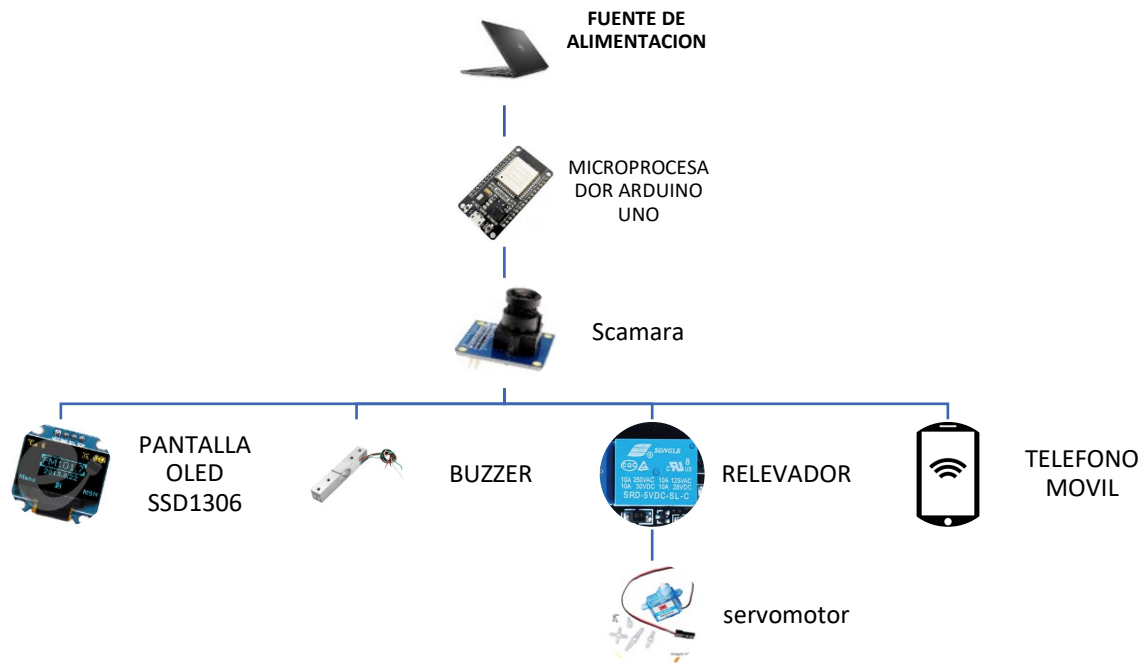
- Desarrollar un modelo de inteligencia artificial capaz de identificar y clasificar los tipos de residuos más comunes (orgánicos, plásticos, metálicos, papel) mediante el procesamiento de imágenes en tiempo real.
- Implementar un sistema de captura visual con cámara y procesamiento de imágenes utilizando herramientas como Python y OpenCV.
- Integrar el sistema de clasificación con un controlador electrónico (por ejemplo, Arduino) para accionar mecanismos automáticos que direccionen los residuos hacia contenedores específicos.
- Evaluar la eficacia del sistema mediante pruebas funcionales y validar su aplicabilidad en entornos educativos y comunitarios para fomentar la conciencia ambiental.

5. Desarrollo:

5.1. Título del proyecto:

Clasificador inteligente de residuos con visión artificial para la gestión sostenible de desechos en Bolivia.

5.2. Diagrama de bloques:



El diagrama de bloques representa la estructura funcional del sistema clasificador de residuos con visión artificial. Está compuesto por una cámara que captura la imagen del residuo, un microcontrolador que procesa la información mediante un modelo de inteligencia artificial, y un sistema mecánico (servomotores y relevadores) que dirige el residuo hacia el contenedor correspondiente. Adicionalmente, un sensor de peso monitorea la capacidad de los basureros para evitar el sobrellenado. Todo el sistema es alimentado por una fuente de energía externa (computadora), y se apoya en una pantalla LED para mostrar el estado del proceso.

5.3. Función de Cada Etapa del Diagrama de Bloques

- **Computadora (PC):**

La computadora actúa como la unidad principal de procesamiento para el sistema de inteligencia artificial. En ella se ejecuta el código de entrenamiento y reconocimiento de imágenes, utilizando

herramientas como Python y librerías de visión artificial (por ejemplo, OpenCV). Recibe las imágenes capturadas por la cámara, procesa y clasifica el tipo de residuo detectado.

- **Arduino:**

El Arduino funciona como controlador intermediario que recibe la señal de clasificación desde la computadora y gestiona las acciones físicas del sistema. Controla los actuadores (servomotores), activa el relevador para abrir el basurero correspondiente y recibe la señal del sensor de peso para monitorear el nivel de llenado.

- **Pantalla LED:**

La pantalla LED muestra información al usuario, como el tipo de residuo detectado, el estado del sistema y alertas cuando un basurero está lleno. Facilita la interacción y comunicación visual con el usuario.

- **Relevador:**

El relevador es un interruptor electromecánico controlado por el Arduino que permite activar o desactivar dispositivos de mayor consumo eléctrico, en este caso, el mecanismo de apertura del basurero (como un motor o servomotor). Protege el sistema y aísla el control de baja tensión de la carga eléctrica.

- **Servomotor:**

El servomotor es el actuador responsable de mover físicamente la tapa del basurero, abriéndola automáticamente cuando se detecta el residuo correspondiente. Su precisión y control permiten una apertura controlada y eficiente.

- **Sensor de peso:**

Este sensor detecta el nivel de llenado del basurero midiendo el peso acumulado. Cuando el sensor indica que el basurero está lleno, envía una señal al Arduino para que se active una alerta o se detenga la recepción de residuos en ese contenedor, evitando desbordes.

5.4. Estructura completa del sistema

La estructura del sistema se compone de un conjunto integrado de componentes de hardware y software que trabajan coordinadamente para identificar, clasificar y redirigir residuos a su contenedor correspondiente. El sistema está conformado por:

Computadora con software de visión artificial (Python + modelo IA entrenado)

Cámara para capturar la imagen del residuo

Arduino UNO como microcontrolador para el control de dispositivos físicos

Pantalla LED para informar al usuario sobre el tipo de residuo y el estado del sistema

Servomotores, encargados de abrir la tapa del basurero correcto

Sensor de peso colocado en cada basurero para detectar si está lleno

Módulo de relevador, para controlar con seguridad los servos o motores

Contenedores físicos separados para residuos orgánicos, plásticos, papel y metales

Fuente de alimentación desde la computadora, que alimenta tanto al Arduino como al resto de los componentes

5.5. Descripción del funcionamiento de la estructura del sistema

El funcionamiento del sistema sigue una secuencia automatizada:

El usuario acerca el residuo frente a la cámara conectada a la computadora.

El software desarrollado en Python, con el modelo de inteligencia artificial previamente entrenado, procesa la imagen y reconoce el tipo de residuo (orgánico, plástico, papel o metal).

El resultado de la clasificación se envía desde la computadora al Arduino a través de comunicación serial.

El Arduino recibe la señal correspondiente y activa el servomotor que abre el basurero adecuado para ese tipo de desecho.

Al mismo tiempo, el sistema puede mostrar en la pantalla LED el tipo de residuo reconocido, y si corresponde, activar un sonido o luz indicadora.

Cada basurero cuenta con un sensor de peso que mide constantemente su carga. Si se detecta que un contenedor está lleno, el sensor envía una señal al Arduino, y este bloquea la apertura de ese basurero, mostrando una alerta visual o sonora al usuario.

Finalmente, cuando se deposita el residuo, el servomotor cierra automáticamente la tapa, completando el ciclo.

5.6. Descripción de la construcción del prototipo

La construcción del prototipo se realizó en varias etapas:

- **Etapas 1: Desarrollo del modelo de IA**

Se recopiló una base de datos de imágenes de residuos clasificados (orgánicos, plásticos, papel, metales). Luego, se entrenó un modelo utilizando técnicas de aprendizaje profundo con Python y TensorFlow/Keras, alcanzando una precisión aceptable para el reconocimiento en tiempo real.

- **Etapla 2: Montaje de la estructura física**

Se utilizaron recipientes plásticos adaptados como basureros. En la tapa de cada contenedor se instaló un servomotor, conectado mecánicamente para permitir la apertura automática. También se fijaron sensores de peso debajo de cada recipiente para detectar su nivel de llenado.

- **Etapla 3: Integración electrónica**

Se conectaron todos los componentes al Arduino UNO, incluyendo servomotores, sensores, pantalla LED y módulo de relevador. La alimentación se hizo a través del puerto USB de la computadora, que también maneja la comunicación con el microcontrolador.

- **Etapla 4: Pruebas funcionales**

Finalmente, se realizaron pruebas completas del sistema, simulando distintos tipos de residuos frente a la cámara, validando que el prototipo identifique correctamente el material, accione el servomotor correspondiente y responda adecuadamente al estado del basurero.

5.7. Costo del sistema:

N.º	Componente	Cantidad	Precio Unitario (Bs.)	Precio Total (Bs.)
1	Arduino UNO (para control externo)	1	65.00	65.00
2	Sensor de peso (HX711 + celda)	1	40.00	40.00
3	Servomotor SG90	2	18.00	36.00

N.º	Componente	Cantidad	Precio Unitario (Bs.)	Precio Total (Bs.)
4	Módulo de relé 1 canal	1	12.00	12.00
5	Pantalla OLED 0.96” (I2C)	1	38.00	38.00
6	Set Cables Dupont / Conexión	1	5.00	5.00
7	Fuente de alimentación (USB / PC)	1	0.00*	0.00
8	Fuente de alimentación (USB / PC)	1	0.00*	0.00
9	Protoboard o base de conexión	1	15.00	15.00

6. Marco Teórico

La gestión de residuos es un problema ambiental crítico en muchos países en vías de desarrollo, como Bolivia. La acumulación desorganizada de basura, sin clasificación previa, no solo satura los vertederos, sino que también genera contaminación del suelo, agua y aire. El reciclaje es una de las soluciones más efectivas para mitigar este problema, pero requiere una separación adecuada de los residuos desde su origen.

6.1. Clasificación de residuos

Los residuos sólidos urbanos se dividen comúnmente en: orgánicos (restos de comida, cáscaras), inorgánicos (plásticos, metales, cartón) y peligrosos. La separación efectiva permite su posterior reciclaje, compostaje o disposición segura. Sin embargo, en Bolivia esta práctica aún no está extendida debido a la falta de conciencia, infraestructura y tecnología accesible.

6.2. Visión artificial

La visión artificial permite que un sistema informático "vea" e interprete imágenes del mundo real. A través de técnicas de procesamiento de imágenes y redes neuronales convolucionales (CNN), es posible entrenar un modelo de inteligencia artificial para que identifique objetos, en este caso, tipos de residuos. Herramientas como OpenCV y TensorFlow permiten implementar estos modelos con cámaras convencionales y bajo costo.

6.3. Arduino y sistemas embebidos

Arduino es una plataforma de desarrollo de hardware libre que permite la creación de prototipos electrónicos. En este proyecto se utiliza como controlador de servomotores, sensores de peso y pantalla LED, recibiendo las instrucciones desde la computadora para accionar el basurero correspondiente. Gracias a su bajo costo y facilidad de programación, es ideal para proyectos educativos y de automatización.

Sensores de peso y servomecanismos

Los sensores de peso (como celdas de carga) permiten detectar si un basurero está lleno, lo cual ayuda a evitar el desbordamiento. Los servomotores, por otro lado, son utilizados para controlar la apertura automatizada de tapas de contenedores con precisión y bajo consumo de

energía.

6.4. Educación ambiental y tecnología

El uso de tecnologías accesibles como IA y Arduino puede ser una herramienta pedagógica poderosa para fomentar la educación ambiental. Implementar este tipo de soluciones en entornos escolares o comunitarios ayuda a generar conciencia sobre la importancia del reciclaje y la sostenibilidad.

7. Marco Metodológico

Este proyecto adopta una metodología experimental y aplicada, orientada al diseño, implementación y prueba de un sistema funcional que clasifica residuos utilizando visión artificial y control físico mediante Arduino.

7.1. Enfoque metodológico

Tipo de investigación: Tecnológica – aplicada.

Método: Experimental.

Diseño: Prototipado iterativo y pruebas funcionales.

Población objetivo: Comunidad educativa, estudiantes y público en general interesado en reciclaje.

7.2. Fases del desarrollo

- Investigación y análisis del problema

Se recopilaban datos sobre el manejo actual de residuos en Bolivia y se revisaron antecedentes de proyectos similares a nivel internacional.

- **Recolección de imágenes y entrenamiento del modelo**

Se construyó una base de datos de imágenes clasificadas por tipo de residuo. Estas se utilizaron para entrenar un modelo de red neuronal usando Python, TensorFlow y Keras.

- **Diseño del sistema y selección de componentes**

Se eligieron los sensores, servomotores y microcontroladores necesarios para construir el prototipo, priorizando el bajo costo y disponibilidad local.

- **Construcción del prototipo**

Se ensambló el sistema físico con servomotores, contenedores, sensores de peso y una cámara conectada a una computadora para el reconocimiento de residuos.

- **Integración y programación**

Se desarrollaron los códigos necesarios en Python (para IA y comunicación) y en Arduino IDE (para control de actuadores y sensores).

- **Pruebas funcionales**

Se realizaron pruebas con residuos reales simulados para verificar la precisión del reconocimiento, el accionamiento de los mecanismos y la respuesta ante basureros llenos.

- **Evaluación de resultados y validación**

Se analizó el desempeño del sistema en términos de precisión de clasificación, tiempo de respuesta y funcionalidad de los sensores.

8. Conclusiones

El presente proyecto logró demostrar que es posible diseñar e implementar un sistema inteligente de clasificación de residuos utilizando visión artificial y componentes electrónicos de

bajo costo. La integración de un modelo de inteligencia artificial con un sistema físico basado en Arduino permitió reconocer correctamente distintos tipos de residuos y dirigirlos hacia su contenedor correspondiente mediante la apertura automática de tapas.

La inclusión de sensores de peso permitió mejorar la gestión operativa del sistema, previniendo el desbordamiento de residuos y generando alertas cuando los basureros se encuentran llenos. Además, el uso de una pantalla y elementos de retroalimentación visual o sonora contribuyó a una mejor interacción con el usuario.

Este prototipo representa una solución educativa, funcional y replicable que puede ser aplicada en entornos académicos para concienciar a los estudiantes sobre la importancia de la separación de residuos y el uso de tecnología con propósito social. Si bien el sistema fue validado en un entorno controlado, se identificó su potencial para escalar hacia entornos más amplios, como centros educativos, instituciones públicas y comunidades, como una alternativa viable hacia una gestión de residuos más sostenible.

8.1. Recomendaciones

- **Implementación en entornos reales:**

Se recomienda adaptar el prototipo para su instalación en espacios públicos como colegios, universidades, parques o plazas, donde su uso pueda fomentar la educación ambiental a mayor escala.

- **Mejora del modelo de IA con más datos:**

Para lograr mayor precisión y robustez, es importante ampliar la base de datos de imágenes de residuos y entrenar modelos más complejos que reconozcan subcategorías (por ejemplo, plásticos PET, envases, cartón, etc.).

- **Uso de materiales duraderos y resistentes:**

En versiones futuras, se sugiere construir la carcasa y los mecanismos de apertura con materiales más robustos (acero inoxidable, acrílico, etc.) para garantizar resistencia al uso continuo y condiciones climáticas adversas.

- **Sistema autónomo e inalámbrico:**

Integrar fuentes de energía independientes (como baterías recargables o paneles solares) y módulos Wi-Fi para que el sistema funcione sin necesidad de estar conectado constantemente a una computadora.

- **Interfaz de usuario amigable y educativa:**

Agregar pantallas o aplicaciones móviles que brinden información al usuario sobre el impacto del reciclaje, estadísticas de residuos clasificados o consejos ecológicos para fomentar una cultura de sostenibilidad.

- **Escalabilidad a nivel municipal o institucional:**

Con mejoras en el hardware y software, este sistema podría convertirse en parte de un plan piloto para ser implementado por gobiernos locales o empresas recolectoras como parte de sus estrategias de separación y reciclaje en origen.

9. Bibliografía

- Atria Innovation. (2023). *Inteligencia artificial aplicada a la clasificación de residuos*. Recuperado de: <https://atriainnovation.com/blog/inteligencia-artificial-aplicada-a-la-clasificacion-de-residuos/>
- ResearchGate. (2024). *Uso de la visión artificial para la clasificación de residuos sólidos*. Recuperado de: https://www.researchgate.net/publication/383823176_Uso_de_la_vision_artificial_para_la_clasificacion_de_residuos_solidos_Use_of_machine_vision_for_solid_waste_classification
- Universidad Francisco de Paula Santander. (2023). *Identificación y clasificación de residuos sólidos mediante procesamiento de imágenes e inteligencia artificial*. Recuperado de: <https://repositorio.ufps.edu.co/handle/ufps/6785?show=full>
- Wikipedia. (2025). *Segmentación (procesamiento de imágenes)*. Recuperado de: https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_%28procesamiento_de_im%C3%A1genes%29
- Organización Mundial de la Salud (OMS). (2021). *La calidad del aire y la salud*. [https://www.who.int/es/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
- Agencia de Protección Ambiental de EE. UU. (EPA). (2022). *Indoor Air Quality (IAQ)*. <https://www.epa.gov/indoor-air-quality-iaq>
- Caloryfrio. (s.f.). *¿Qué es la calidad del aire interior (CAI)?* <https://www.caloryfrio.com/construccion-sostenible/ventilacion-y-calidad-aire-interior/que-es-la->

[calidad-del-aire-interior-cai.html](#)

- Bassi, A., Bauer, M., Fiedler, M., & Kramp, T. (2020). *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*. Springer.

<https://link.springer.com/book/10.1007/978-3-642-40403-0>

- OpenCV (2023). *Open Source Computer Vision Library*.

<https://opencv.org/>

- TensorFlow (2024). *An end-to-end open source platform for machine learning*.

<https://www.tensorflow.org/>

10. Anexo A – Código fuente del sistema de clasificación de residuos

con visión artificial:

CODIGO:

```
MAIN.PY
# Libraries
from tkinter import *
from PIL import Image, ImageTk
import imutils
import cv2
import numpy as np
from ultralytics import YOLO
import math

def clean_lbl():
    # Clean
    lblimg.config(image="")
    lblimgtxt.config(image="")

def images(img, imgtxt):
    img = img
    imgtxt = imgtxt

    # Img Detect
    img = np.array(img, dtype="uint8")
    img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
    img = Image.fromarray(img)

    img_ = ImageTk.PhotoImage(image=img)
    lblimg.configure(image=img_)
    lblimg.image = img_

    # Img Text
    imgtxt = np.array(imgtxt, dtype="uint8")
    imgtxt = cv2.cvtColor(imgtxt, cv2.COLOR_BGR2RGB)
    imgtxt = Image.fromarray(imgtxt)
    img_txt = ImageTk.PhotoImage(image=imgtxt)
    lblimgtxt.configure(image=img_txt)
    lblimgtxt.image = img_txt

# Scanning Function
def Scanning():
    global img_metal, img_glass, img_plastic, img_carton, img_medical
    global img_metaltxt, img_glasstxt, img_plastictxt, img_cartontxt, img_medicaltxt, pantalla
    global lblimg, lblimgtxt

    # Interfaz
```

```

lblimg = Label(pantalla)
lblimg.place(x=75, y=260)

lblimgtxt = Label(pantalla)
lblimgtxt.place(x=995, y=310)
detect = False

# Read VideoCapture
if cap is not None:
    ret, frame = cap.read()
    frame_show = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # True
    if ret == True:
        # Yolo | AntiSpoof
        results = model(frame, stream=True, verbose=False)
        for res in results:
            # Box
            boxes = res.boxes
            for box in boxes:
                detect = True
                # Bounding box
                x1, y1, x2, y2 = box.xyxy[0]
                x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

                # Error < 0
                if x1 < 0: x1 = 0
                if y1 < 0: y1 = 0
                if x2 < 0: x2 = 0
                if y2 < 0: y2 = 0

                # Class
                cls = int(box.cls[0])

                # Confidence
                conf = math.ceil(box.conf[0])
                #print(f"Clase: {cls} Confidence: {conf}")
                # Metal
                if cls == 0:
                    # Draw
                    cv2.rectangle(frame_show, (x1, y1), (x2, y2), (255, 255, 0), 2)
                    # Text
                    text = f'{clsName[cls]} {int(conf) * 100}%'
                    sizetext = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
                    dim = sizetext[0]
                    baseline = sizetext[1]
                    # Rect
                    cv2.rectangle(frame_show, (x1, y1 - dim[1] - baseline), (x1 + dim[0], y1 + baseline), (0, 0, 0), cv2.FILLED)
                    cv2.putText(frame_show, text, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 0), 2)

                # Clasificacion
                images(img_metal, img_metaltxt)

```

```

if cls == 1:
    # Draw
    cv2.rectangle(frame_show, (x1, y1), (x2, y2), (255, 255, 255), 2)
    # Text
    text = f'{clsName[cls]} {int(conf) * 100}%'
    sizetext = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
    dim = sizetext[0]
    baseline = sizetext[1]
    # Rect
    cv2.rectangle(frame_show, (x1, y1 - dim[1] - baseline), (x1 + dim[0], y1 + baseline),
                  (0, 0, 0), cv2.FILLED)
    cv2.putText(frame_show, text, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)

    # Clasificacion
    images(img_glass, img_glasstxt)

if cls == 2:
    # Draw
    cv2.rectangle(frame_show, (x1, y1), (x2, y2), (0, 0, 255), 2)
    # Text
    text = f'{clsName[cls]} {int(conf) * 100}%'
    sizetext = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
    dim = sizetext[0]
    baseline = sizetext[1]
    # Rect
    cv2.rectangle(frame_show, (x1, y1 - dim[1] - baseline), (x1 + dim[0], y1 + baseline),
                  (0, 0, 0), cv2.FILLED)
    cv2.putText(frame_show, text, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

    # Clasificacion
    images(img_plastic, img_plastictxt)

if cls == 3:
    # Draw
    cv2.rectangle(frame_show, (x1, y1), (x2, y2), (150, 150, 150), 2)
    # Text
    text = f'{clsName[cls]} {int(conf) * 100}%'
    sizetext = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
    dim = sizetext[0]
    baseline = sizetext[1]
    # Rect
    cv2.rectangle(frame_show, (x1, y1 - dim[1] - baseline), (x1 + dim[0], y1 + baseline),
                  (0, 0, 0), cv2.FILLED)
    cv2.putText(frame_show, text, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (150, 150, 150), 2)

    # Clasificacion
    images(img_carton, img_cartontxt)

if cls == 4:
    # Draw
    cv2.rectangle(frame_show, (x1, y1), (x2, y2), (255, 0, 0), 2)
    # Text

```



```

        text = f'{clsName[cls]} {int(conf) * 100}%'
        sizetext = cv2.getTextSize(text, cv2.FONT_HERSHEY_SIMPLEX, 1, 2)
        dim = sizetext[0]
        baseline = sizetext[1]
        # Rect
        cv2.rectangle(frame_show, (x1, y1 - dim[1] - baseline), (x1 + dim[0], y1 + baseline),
                      (0, 0, 0), cv2.FILLED)
        cv2.putText(frame_show, text, (x1, y1 - 5), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

    # Clasificacion
    images(img_medical, img_medicaltxt)

    if detect == False:
        # Clean
        clean_lbl()

        # Resize
        frame_show = imutils.resize(frame_show, width=640)

        # Convertimos el video
        im = Image.fromarray(frame_show)
        img = ImageTk.PhotoImage(image=im)

        # Mostramos en el GUI
        lblVideo.configure(image=img)
        lblVideo.image = img
        lblVideo.after(10, Scanning)

    else:
        cap.release()

# main
def ventana_principal():
    global cap, lblVideo, model, clsName, img_metal, img_glass, img_plastic, img_carton, img_medical
    global img_metaltxt, img_glasstxt, img_plastictxt, img_cartontxt, img_medicaltxt, pantalla
    # Ventana principal
    pantalla = Tk()
    pantalla.title("RECICLAJE INTELIGENTE")
    pantalla.geometry("1280x720")

    # Background
    imagenF = PhotoImage(file="setUp/Canva.png")
    background = Label(image=imagenF, text="Inicio")
    background.place(x=0, y=0, relwidth=1, relheight=1)

    # Clases: 0 -> Metal | 1 -> Glass | 2 -> Plastic | 3 -> Carton | 4 -> Medical
    # Model
    model = YOLO('Modelos/best.pt')

    # Clases
    clsName = ['Metal', 'Glass', 'Plastic', 'Carton', 'Medical']

    # Images

```

```

img_metal = cv2.imread("setUp/metal.png")
img_glass = cv2.imread("setUp/vidrio.png")
img_plastic = cv2.imread("setUp/plastico.png")
img_carton = cv2.imread("setUp/carton.png")
img_medical = cv2.imread("setUp/medical.png")
img_metaltxt = cv2.imread("setUp/metaltxt.png")
img_glasstxt = cv2.imread("setUp/vidriotxt.png")
img_plastictxt = cv2.imread("setUp/plasticotxt.png")
img_cartontxt = cv2.imread("setUp/cartontxt.png")
img_medicaltxt = cv2.imread("setUp/medicaltxt.png")

# Video
lblVideo = Label(pantalla)
lblVideo.place(x=320, y=180)

# Elegimos la camara
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
cap.set(3, 1280)
cap.set(4, 720)
Scanning()

# Eject
pantalla.mainloop()
if __name__ == "__main__":
    ventana_principal()

TRASHETECTED.PY
# Importamos librerias
from ultralytics import YOLO
import cv2
import math

# Modelo
#model = YOLO('Modelos/best.pt')
model = YOLO('Modelos/best.pt') # Usa GPU
# o
# Cap
cap = cv2.VideoCapture(0)
cap.set(3, 1280)
cap.set(4, 720)

# Clases
clsName = ['Metal', 'Glass', 'Plastic', 'Carton', 'Medical']

# Inference
while True:
    # Frames
    ret, frame = cap.read()

    # Yolo | AntiSpoof
    results = model(frame, stream=True, verbose=False)
    for res in results:
        # Box

```

```

boxes = res.boxes
for box in boxes:
    # Bounding box
    x1, y1, x2, y2 = box.xyxy[0]
    x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

    # Error < 0
    if x1 < 0: x1 = 0
    if y1 < 0: y1 = 0
    if x2 < 0: x2 = 0
    if y2 < 0: y2 = 0

    # Class
    cls = int(box.cls[0])

    # Confidence
    conf = math.ceil(box.conf[0])
    print(f'Clase: {cls} Confidence: {conf}')

    if conf > 0:
        # Draw
        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
        cv2.putText(frame, f'{clsName[cls]} {int(conf * 100)}%', (x1, y1 - 20),
                    cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 255), 2)

# Show
cv2.imshow("Waste Detect", frame)

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

cap.release()
cv2.destroyAllWindows()

# o
torch.cuda.is_available()

```