

Comparative performance evaluation of waste detection models in urban environments

Evaluación comparativa del rendimiento de modelos de detección de residuos en entornos urbanos

Niño, C. V.¹; CASTRO, S. A.²; DÍAZ, C. Z.³

¹**MSc. Carlos Vicente Niño Rondón, Ingeniero Electrónico.** Programa de Ingeniería Electrónica. Facultad de Ingeniería. Universidad Francisco de Paula Santander. e-mail: carlosvicentenr@ufps.edu.co. ORCID: <https://orcid.org/0009-0003-3997-6828>

²**MSc. Sergio Alexander Castro Casadiego, Ingeniero Electrónico.** Programa de Ingeniería Electrónica. Facultad de Ingeniería. Universidad Francisco de Paula Santander. e-mail: sergio.castroc@ufps.edu.co. ORCID: <https://orcid.org/0000-0003-0962-9916>

³**Ing. Crhistian Zahir Diaz Garcia., Ingeniero Electrónico.** Programa de Ingeniería Electrónica. Facultad de Ingeniería. Universidad Francisco de Paula Santander. e-mail: crhastianzahirdiga@ufps.edu.co. ORCID: <https://orcid.org/0009-0003-3997-6828>

Entidad

Universidad Francisco de Paula Santander, Cúcuta, Norte de Santander, Colombia.

Tel: 60-7-5776655 Ext 202

E-mail: carlosvicentenr@ufps.edu.co

Recibido: 25/01/2024 / Aceptado: 10/05/2024

Resumen

El presente artículo tuvo como objetivo evaluar y comparar el rendimiento de tres modelos de detección de residuos: Faster R-CNN ResNet101, SSD MobileNet V2, SSD MobileNet V2 FPNLite. Se utilizó un conjunto de imágenes propio obtenido por medio de fotografías tomadas en un entorno urbano como lo es el Canal Bogotá de la ciudad de Cúcuta, este conjunto de imágenes se utilizó para el entrenamiento y evaluación de los modelos de detección. Se evaluaron aspectos clave como lo son la precisión, Recall, F1-Score, función de pérdida, tiempo de ejecución y tamaño de los modelos. el modelo que presento mejores características de rendimiento fue el modelo Faster R-CNN ResNet101, con un valor de F1-Score del 83.8%, una precisión del 89.3% y un 78.9% de Recall.

Palabras clave: Modelos de detección, procesamiento de imágenes, rendimiento, precisión, residuos.

Abstract

The objective of this paper was to evaluate and compare the performance of three debris detection models: Faster R-CNN ResNet101, SSD MobileNet V2, SSD MobileNet V2 FPNLite. An own set of images obtained from photographs taken in the Bogotá Channel in the city of Cúcuta was used to train the detection models. Key aspects such as accuracy, Recall, F1-Score, loss function, execution time and size of the models were evaluated. The model with the best performance characteristics was the Faster R-CNN ResNet101 model, with an F1-Score value of 83.8%, an accuracy of 89.3% and a Recall value of 78.9%.

Keywords: Detection models, image processing, performance, accuracy, waste.

1. INTRODUCCIÓN

La gestión de los residuos es uno de los retos al que se enfrenta la sociedad en general, en el que una mala gestión de residuos tiene un impacto negativo en el cuidado del medio ambiente y en la salud de las personas. La presencia y gestión defectuosa de residuos tiene una relación directa con la contaminación ambiental, la magnitud de esta contaminación está ligada a las fuentes que generan estos residuos y a las herramientas adoptadas por la sociedad para su correcto manejo y tratamiento. El daño generado por la presencia de residuos orgánicos e inorgánicos está relacionado con la atención que los humanos les dan, ya que comúnmente estos son arrojados en botaderos improvisados, concentrados o dispersos en el área urbana, o vertederos donde se queman al aire libre, impactando negativamente en recursos como lo son el agua, el aire y el suelo (Daroca-Capell, 2014).

Esta investigación se enfoca en entornos urbanos como lo es el Canal Bogotá de la ciudad de Cúcuta, Norte de Santander, esta es una infraestructura sumamente importante para el manejo de aguas lluvias

de la ciudad. Esta infraestructura fue creada para albergar las corrientes pluviales, pero este canal se ve afectado por una problemática que va más allá de su función inicial, como lo es la presencia de basuras y residuos de construcción y demolición, como lo señala El Tiempo en un artículo publicado en 2022, indica una realidad que demanda una atención inmediata (El Tiempo, 2022).

A partir de esta problemática ambiental, surge la posibilidad de abordar este tipo de problemáticas por medio de herramientas innovadoras y eficientes. En este contexto la detección automática de residuos por medio de sistemas basados en visión por computador e inteligencia artificial se presenta como una herramienta para la gestión de residuos en zonas urbanas de forma eficaz (Frutos *et al.*, 2023).

En el año 2022 se realizó una optimización de un modelo de detección de huecos y de bolsas de basura usando la tecnología de aprendizaje profundo, para la implementación se usó un prototipo utilizando una Raspberry Pi 4 Model B, en el cual se utilizaron los modelos de detección Faster R-CNN, SSD MobileNet V2 y SSD MobileNet V2 FPNLite, en el que

se obtuvo que el modelo SSD MobileNet V2 fue el más preciso de estos tres, Mientras que el valor de Recall más alto lo obtuvo el modelo SSD MobileNet V2 FPNLite (Hernández-Rodríguez, 2022).

Otra investigación realizada sobre esta temática se presentó en el año 2021, esta investigación consistió en utilizar técnicas de aprendizaje profundo para detectar objetos en áreas rurales, entre los modelos de aprendizaje profundo utilizados en esta investigación se encuentra el modelo Faster R-CNN ResNet101 y el modelo SSD MobileNet V2. En la investigación se obtuvo que, en cuanto a tiempo de ejecución, el modelo SSD MobileNet V2 fue el más rápido, mientras que, el modelo del tipo Faster R-CNN fue más lento (Barba-Guamán, 2021).

El presente artículo se realizó con el objetivo de evaluar y comparar el rendimiento de tres modelos de detección utilizados para detectar residuos: Faster R-CNN ResNet101, SSD MobileNet V2, SSD MobileNet V2 FPNLite por medio de aspectos clave como lo son la precisión, Recall, F1-Score, función de pérdida, tiempo de ejecución y tamaño de los modelos.

2. METODOLOGIA

La metodología de la presente investigación está compuesta por cuatro etapas: la primera de estas etapas consiste en una descripción del conjunto de datos empleado, en la segunda etapa se presentan los modelos de detección a utilizarse, en la tercera etapa se describen las configuraciones realizadas en estos modelos de detección y, por último, en la cuarta etapa se realiza la evaluación del

rendimiento de los modelos de detección de residuos.

A. Conjunto de datos: El conjunto de datos empleado para el desarrollo de la presente investigación está compuesto por 1340 imágenes, estas imágenes fueron obtenidas por medio de fotografías tomadas en zonas pertenecientes al Canal Bogotá de la ciudad de Cúcuta, Norte de Santander. Estas imágenes fueron sometidas a un proceso de etiquetado por medio de la herramienta LabelImg, este proceso de etiquetado se muestra en la Figura 1, dicho proceso permite seleccionar la variable objetivo (Residuos), indicando lo que se quiere predecir con el uso de los modelos de detección, este proceso de etiquetado genera un archivo “xml” para cada imagen, dicho archivo contiene la información de las etiquetas realizadas (Aljabri *et al.*, 2022).



Figura 1. Proceso de etiquetado.

Este conjunto de datos compuesto por imágenes y archivos “xml” se dividió en dos partes, un 80% de este corresponde al conjunto de datos para entrenamiento, mientras que, el 20% restante corresponde al conjunto de datos que se utilizará para evaluar el rendimiento de los modelos de detección (Vrigazova, 2021). En la Tabla 1,

se observa la composición del conjunto de datos utilizado.

Tabla 1. Conjunto de datos

Número total de imágenes	1340
Número total de archivos "xml"	1340
Imágenes de entrenamiento	1062
Imágenes de evaluación	278
Ancho de las imágenes	224 píxeles
Alto de las imágenes	224 píxeles

B. Arquitecturas de detección: Para realizar la actividad de identificar residuos, se utilizaron modelos de detección pre-entrenados de Tensorflow 2, se utilizaron los siguientes 3 modelos:

Faster R-CNN ResNet101: Este modelo de detección de objetos emplea una red neuronal convolucional profunda para generar regiones de interés en las imágenes y dividir estas regiones como objetos y no objetos. La fortaleza de este modelo es la precisión (Zhou *et al.*, 2021).

SSD MobileNet V2: Este modelo de detección de objetos la cuál es una arquitectura de aprendizaje profundo que permite detectar objetos en imágenes, este modelo tiene una alta velocidad de ejecución y al mismo tiempo no descuida la precisión para realizar las detecciones (Shamsuddin *et al.*, 2022).

SSD MobileNet V2 FPNLite: Este modelo es similar a la arquitectura anteriormente descrita, en la que se sacrifica un poco de precisión a fin de tener una velocidad de ejecución más optimizada, esta arquitectura es ideal para implementarse en sistemas con recursos de hardware limitados (Saiful Bahri *et al.*, 2023).

En la Tabla 2, se presenta la comparación de la velocidad y precisión de los modelos.

Tabla 2. Comparación de modelos

Nombre del modelo	Velocidad (ms)	COCO mAP
Faster R-CNN ResNet50	55	31.8
SSD MobileNet V2	19	20.2
SSD MobileNet V2 FPNLite	22	22.2

C. Configuraciones de entrenamiento:

En el entrenamiento de los modelos de detección se realizaron distintas configuraciones para adaptar estos modelos a las necesidades de la investigación.

La primera de estas configuraciones corresponde al conjunto de datos utilizado para el entrenamiento, en el que se estableció como única clase a detectar la clase "Residuos", a su vez, se determinaron las dimensiones de las imágenes con 224 píxeles de ancho y 224 píxeles de alto.

La segunda de estas configuraciones corresponde al aumento de datos, ya que, los modelos de detección de Tensorflow tienen un apartado para realizar un aumento de datos al conjunto de imágenes de entrenamiento, este aumento de datos consistió en aplicar una transformación que invierte la imagen de forma horizontal, esta transformación genera una imagen adicional para cada imagen del conjunto de datos de entrenamiento.

La tercera de estas configuraciones corresponde al número de pasos de entrenamiento, el número de pasos establecido para el entrenamiento fue de

13500. El resumen de estas configuraciones se presenta en la Tabla 3.

Tabla 3. Configuraciones de entrenamiento

Clase	Residuos
Ancho de la imagen	224 píxeles
Alto de la imagen	224 píxeles
Aumento de datos	Transformación horizontal
Número de pasos	13500

D. Evaluación del rendimiento de los modelos de detección: La ejecución del entrenamiento y evaluación de los modelos de detección se llevó a cabo a través de la herramienta Google Colab, esta herramienta es un producto de Google Research que permite desarrollar proyectos de automático a través del acceso a potentes opciones de hardware como lo es el uso de GPU y TPU de forma gratuita (Bisong, 2019).

Para evaluar los modelos de detección entrenados se tendrán en cuenta parámetros como lo es la función de pérdida, el tiempo de entrenamiento, el tiempo de ejecución y el tamaño de los modelos en megabytes. Además, se someterán los modelos al conjunto de datos de evaluación y se utilizará una matriz de confusión para determinar la capacidad de los modelos para identificar la presencia de residuo en imágenes y obtener la precisión con la que se realizan estas detecciones. La composición de la matriz de confusión se presenta en la Figura 1.

Esta matriz de confusión permite conocer el número de detecciones correctas e incorrectas (Corso, 2009) a través de cuatro categorías: Verdaderos positivos (TP): Esta categoría indica el número de residuos detectados correctamente.

Verdaderos negativos (TN): Esta categoría indica la identificación correcta de los materiales que no se consideran residuos, en el caso de los modelos de detección esta categoría toma el valor de cero debido a que, solo se busca detectar los objetos de interés (Residuos).

		Sí	No
Valores Reales	Sí	Verdaderos Positivos TP	Falsos Positivos FP
	No	Falsos Negativos FN	Verdaderos Negativos TN
		Valores Predicción	

Figura 1. Matriz de confusión.

Falsos positivos (FP): Esta categoría indica el número de residuos que se detectaron como residuos, pero en realidad no eran residuos.

Falsos negativos (FN): Esta categoría indica el número de residuos que no fueron detectados por el modelo de detección, pero si eran residuos en realidad.

Con la información que se obtiene a través de la implementación de la matriz de confusión, se hace posible obtener métricas que representan el rendimiento de los modelos de detección, dichas métricas se describen a continuación:

Precisión: La precisión está representada por la ecuación 1 y se utiliza para medir el número de residuos correctamente detectados respecto al total de residuos predichos (Corso, 2009).

$$Precisión = \frac{TP}{TP+FP} \quad (1)$$

Recall: Esta métrica determina la

proporción de los residuos correctamente detectados respecto al total de residuos reales, dicha métrica se calcula por medio de la ecuación 2 (Corso, 2009).

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

F1-Score: Esta métrica se emplea para obtener un valor de rendimiento que combina la precisión y el Recall, se calcula por medio de la ecuación 3 (Corso, 2009).

$$F1 - Score = \frac{2 \times (Precisión + Recall)}{Precisión + Recall} \quad (3)$$

3. ANALISIS Y DISCUSIÓN DE RESULTADOS

Del entrenamiento de los tres modelos a través de la herramienta Google Colab y el uso de GPU, se registró el tiempo de entrenamiento, este registro se muestra en la Tabla 4.

Tabla 4. Tiempo de entrenamiento

Modelo	Tiempo de entrenamiento
Faster R-CNN ResNet101	01h 21m 55s
SSD MobileNet V2	41m 30s
SSD MobileNet V2 FPNLite	23m 20s

De acuerdo a los resultados obtenidos en la Tabla 4 el modelo que tuvo un mayor tiempo de entrenamiento fue la arquitectura Faster R-CNN ResNet101, seguida de la arquitectura SSD MobileNet V2 y la arquitectura que tuvo un menor tiempo de entrenamiento fue la SSD MobileNet V2 FPNLite.

Para obtener un correcto funcionamiento de los modelos de detección se superviso el comportamiento de la función de pérdida, ya que, una correcta función de pérdida

evita un sobreajuste, en el cual, para un valor de función de pérdida más bajo el rendimiento del modelo es de mayor calidad de rendimiento.

El registro de la función de pérdida respecto a los pasos de entrenamiento del modelo Faster R-CNN ResNet101 se presenta en la Tabla 5.

Tabla 5. Función de pérdida del modelo Faster R-CNN ResNet101

Paso de entrenamiento	Función de pérdida
1	1.05
2000	0.6
4000	0.37
6000	0.25
8000	0.2
10000	0.17
12000	0.12
13500	0.1

El registro de la función de pérdida respecto a los pasos de entrenamiento del modelo SSD MobileNet V2 se presenta en la Tabla 6.

Tabla 6. Función de pérdida del modelo SSD MobileNet V2

Paso de entrenamiento	Función de pérdida
1	1.44
2000	0.9
4000	0.85
6000	0.83
8000	0.74
10000	0.78
12000	0.72
13500	0.7

El registro de la función de pérdida respecto a los pasos de entrenamiento del modelo SSD MobileNet V2 FPNLite se presenta en la Tabla 7.

Tabla 7. Función de pérdida del modelo SSD MobileNet V2 FPNLite

Paso de entrenamiento	Función de pérdida
1	1.15
2000	0.7
4000	0.54
6000	0.47
8000	0.52
10000	0.41
12000	0.39
13500	0.4

De los resultados obtenidos en la Tabla 5, Tabla 6, Tabla 7, se observó la evolución de la función de pérdida a medida que se incrementaba el valor del número de pasos de entrenamiento, en donde, la función de pérdida más baja se logró en el modelo Faster R-CNN ResNet101 con un valor de 0.1, mientras que, el modelo SSD MobileNet V2 FPNLite alcanzó un valor de 0.4 y la función de pérdida con un valor más alto se presentó en el modelo SSD MobileNet V2 con un valor de 0.7.

El tamaño de los tres modelos obtenidos se presenta a continuación por medio de la Tabla 8.

Tabla 8. Tamaño de los modelos entrenados

Modelo	Tamaño del modelo
Faster R-CNN ResNet101	369 MB
SSD MobileNet V2	41.9 MB
SSD MobileNet V2 FPNLite	26.8 MB

Los resultados obtenidos en la Tabla 8 permiten identificar que el modelo con un mayor tamaño fue el Faster R-CNN ResNet101 con un tamaño de 369 MB, seguido del modelo SSD MobileNet V2 con un tamaño de 41.9 MB, mientras que, el modelo más ligero en cuanto a tamaño se refiere, fue el SSD MobileNet V2 FPNLite con un tamaño de 26.8 MB.

Los modelos de detección se ejecutaron en el conjunto de datos de evaluación, este conjunto de datos tiene 278 imágenes, el tiempo que tardó cada modelo en procesar todas las imágenes y generar la matriz de confusión se presenta en la Tabla 9.

Tabla 9. Tiempo de ejecución

Modelo	Tiempo de ejecución
Faster R-CNN ResNet101	28 segundos
SSD MobileNet V2	4 segundos
SSD MobileNet V2 FPNLite	6 segundos

El modelo con mayor tiempo de ejecución fue el modelo Faster R-CNN con un valor de 28 segundos, seguido por el modelo SSD MobileNet V2 FPNLite con un tiempo de 6 segundos y por último, el modelo más rápido fue el SSD MobileNet V2 con un tiempo de 4 segundos.

El rendimiento de los modelos de detección de residuos Faster R-CNN, SSD MobileNet V2 y SSD MobileNet V2 FPNLite se muestran en la Figura 2, Figura 3 y Figura 4 respectivamente.

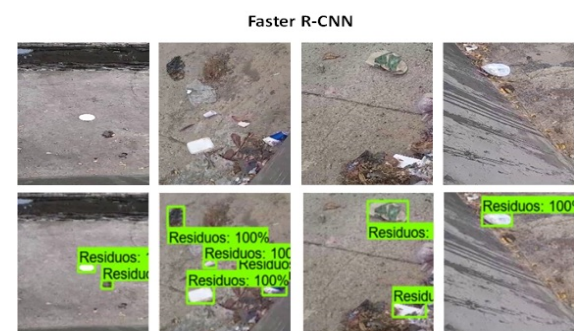


Figura 2. Rendimiento modelo Faster R-CNN ResNet101.



Figura 3. Rendimiento modelo SSD MobileNet V2.



Figura 4. Rendimiento modelo SSD MobileNet V2 FPNLite.

La matriz de confusión obtenida para el modelo Faster R-CNN se muestra en la Figura 5.

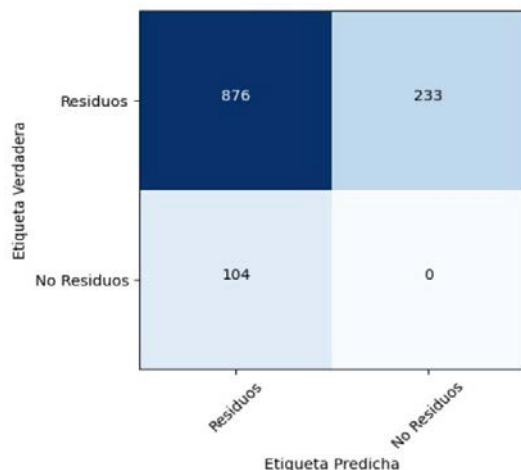


Figura 5. Matriz de confusión Faster R-CNN ResNet101.

Esta matriz de confusión del modelo Faster R-CNN ResNet101 indica que se realizaron

876 detecciones correctas y 337 detecciones incorrectas.

La matriz de confusión obtenida para el modelo SSD MobileNet V2 se muestra en la Figura 6.

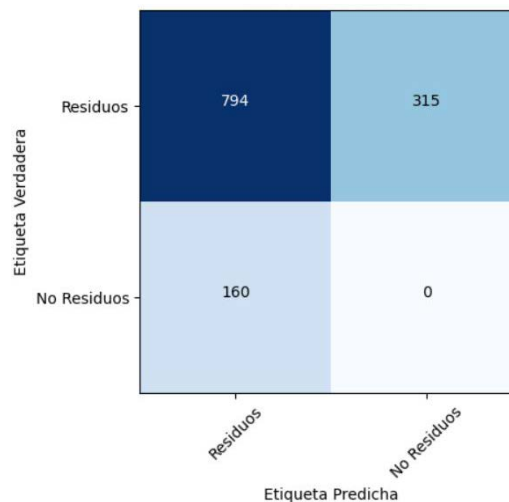


Figura 6. Matriz de confusión SSD MobileNet V2.

Esta matriz de confusión del modelo SSD MobileNet V2 indica que se realizaron 794 detecciones correctas y 475 detecciones incorrectas.

La matriz de confusión obtenida para el modelo SSD MobileNet V2 FPNLite se muestra en la Figura 7.

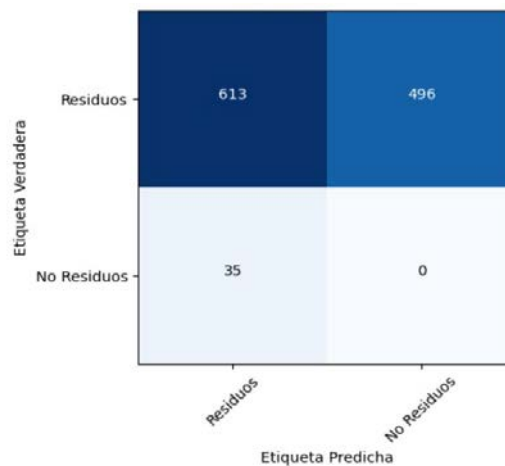


Figura 7. Matriz de confusión SSD MobileNet V2 FPNLite.

Esta matriz de confusión del modelo SSD MobileNet V2 FPNLite indica que se realizaron 613 detecciones correctas y 531 detecciones incorrectas.

A partir de las matrices de confusión obtenidas anteriormente se determinaron y compararon las métricas de precisión, Recall y F1-Score para cada uno de los modelos de detección de residuos utilizados. Esta información se presenta en la Tabla 10.

Tabla 10. Métricas de rendimiento

Modelo	Precisión	Recall	F1-Score
Faster R-CNN ResNet101	89.3%	78.9%	83.8%
SSD MobileNet V2	83.2%	71.6%	76.9%
SSD MobileNet V2 FPNLite	92.6%	62.2%	74.4%

De esta comparación de las métricas se obtuvo que, el modelo con mayor precisión fue el SSD MobileNet V2 FPNLite con un 92.6%, Mientras que el modelo Faster R-CNN obtuvo el mayor valor en cuanto a Recall y F1-Score, con valores de 78.9% y 83.8% respectivamente.

4. CONCLUSIONES

De los modelos entrenados, el modelo que tiene una mayor velocidad, es decir un menor tiempo de ejecución es el modelo SSD MobileNet V2 con un tiempo de 4 segundos, mientras que el modelo más lento es el Faster R-CNN ResNet101 con un tiempo de 28 segundos. A su vez, Faster R-CNN ResNet101 es el modelo con mayor tamaño con un tamaño de 369 MB. Por lo tanto, para implementaciones en dispositivos con recursos de hardware más limitados, como lo puede ser poca memoria RAM, se recomienda no implementarse el

modelo Faster R-CNN, en su lugar implemente los modelos SSD MobileNet, ya que, estos modelos tienen un tamaño en megabytes reducido y mayor velocidad para ejecutarse.

En cuanto a la función de pérdida, el modelo Faster R-CNN obtuvo el menor valor, con un 0.1, mientras que el valor más alto lo presentó el modelo SSD MobileNet V2 con un valor de 0.7.

Por otra parte, el modelo SSD MobileNet V2 FPNLite presentó el valor con mayor precisión (92.6%) y la menor precisión (83.2%) la presentó el modelo SSD MobileNet V2. A su vez, el modelo Faster R-CNN alcanzó un Recall más alto con un valor del 78.9% y el modelo SSD MobileNet V2 FPNLite presentó el Recall más bajo con un valor del 62.2%. Estas métricas indican que, el modelo que presenta mejores características de rendimiento es el modelo Faster R-CNN ResNet101, ya que este modelo obtuvo un valor de F1-Score del 83.8% y esta es la métrica que indica el equilibrio entre precisión y Recall. El modelo con los parámetros de rendimiento más bajos lo presentó el modelo SSD MobileNet V2 FPNLite debido a que su F1-Score tuvo un valor del 74.4%.

El desarrollo de la presente investigación permitió observar la relación que existe entre la función de pérdida y la métrica F1-Score, ya que, el modelo Faster R-CNN ResNet101 presenta el menor valor de función de pérdida (0.1) y también, este modelo fue el que obtuvo un mayor valor de F1-Score (83.8%).

5. REFERENCIAS BIBLIOGRÁFICAS

- Aljabri, M., AlAmir, M., AlGhamdi, M., Abdel-Mottaleb, M., & Collado-Mesa, F. (2022). Towards a better understanding of annotation tools for medical imaging: a survey. *Multimedia Tools and Applications*, 81(18), 25877–25911. <https://doi.org/https://doi.org/10.1007/s11042-022-12100-1>
- Barba-Guamán, L. R. (2021). *Uso de técnicas deep learning para reconocimiento de objetos en áreas rurales (Disertación Doctoral)*. Universidad Politécnica De Madrid. Madrid, España.
- Bisong, E. (2019). Google Colaboratory. En *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (pp. 59–64). Apress, Berkeley, CA. https://doi.org/https://doi.org/10.1007/978-1-4842-4470-8_7
- Corso, C. L. (2009). Aplicación de algoritmos de clasificación supervisada usando Weka. *Universidad Tecnológica Nacional, Facultad Regional Córdoba*.
- Daroca-Capell, T. (2014). *Dimensión y características de los factores que inciden en la contaminación ambiental por residuos sólidos (Tesis de maestría)*. Universidad de San Martín de Porres. Lima, Perú.
- El Tiempo. (2022). *Remueven 4 toneladas de basuras del Canal de Bogotá*. <https://diarioeltiempo.co/nacional/remueven-4-toneladas-de-basuras-del-canal-bogota/1797/>
- Frutos, V., Pérez, B., Campos, S., Barrera, A., García, O., García, F., & Al-Kaff, A. (2023). Detección de residuos urbanos mal ubicados mediante visión por ordenador e IA. *XLIV Jornadas de Automática*, 855–860. <https://doi.org/https://doi.org/10.17979/spudc.9788497498609.855>
- Hernández-Rodríguez, J. D. (2022). *Mejora del modelo de detección de huecos y bolsas de basura basado en deep learning implementado en una Raspberry Pi (en Bogotá, Colombia) (Proyecto final de grado)*. Universidad de los Andes. Bogotá, Colombia.
- Saiful Bahri, I. Z., Saon, S., Mahamad, A. K., Isa, K., Fadlilah, U., Ahmadon, M. A. Bin, & Yamaguchi, S. (2023). Interpretation of Bahasa Isyarat Malaysia (BIM) Using SSD-MobileNet-V2 FPNLite and COCO mAP. *Information (Switzerland)*, 14(6). <https://doi.org/https://doi.org/10.3390/info14060319>
- Shamsuddin, M. F., Azami, M. H., Mohd Zaki, H. F., & Abdullah, N. A. (2022). Effect of Color Contrast to the Accuracy of SSD-MobileNetV2. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3), 18–21.
- Vrigazova, B. (2021). The Proportion for Splitting Data into Training and Test Set for the Bootstrap in Classification Problems. *Business Systems Research*, 12(1), 228–242. <https://doi.org/https://doi.org/10.2478/bsrj-2021-0015>
- Zhou, Y., Wen, S., Wang, D., Mu, J., & Richard, I. (2021). Object detection in autonomous driving scenarios based on an improved faster-RCNN. *Applied Sciences (Switzerland)*, 11(24). <https://doi.org/https://doi.org/10.3390/app112411630>