

(Nombre Trabajo de Grado)
Detección, Georreferenciación y Cuantificación de Daños en Pavimentos Flexibles
a partir de Imágenes, utilizando Aprendizaje Profundo.

Manuel Alejandro Morales Acevedo

(Autor Trabajo de Grado)

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface,
en alcances y calidad, todos los requisitos que demanda
un Trabajo de Grado de Maestría.



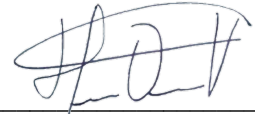
Julián Gil González

Director



Valentina Corchuelo

Jurado



Hernán Vargas

Jurado

Aprobado en cumplimiento de los requisitos exigidos por la
Pontificia Universidad Javeriana Cali, para optar el título de
Magister en Ciencia de Datos.



HERNÁN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias



JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Ciudad y fecha: Cali, 9 de febrero de 2024



Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, 09/02/24

Autor: Manuel Alejandro Morales Acevedo

Título del Trabajo de Grado: “Detección, Georreferenciación y Cuantificación de Daños en Pavimentos Flexibles a partir de Imágenes, utilizando Aprendizaje Profundo.”

Director: Julián Gil González

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

Firma del Director del Trabajo de Grado

Santiago de Cali, 9 de diciembre del 2023

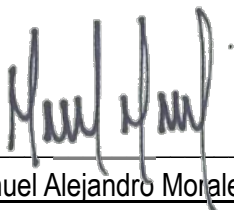
Ingeniero:

Juan Carlos Martínez Arias
Director Posgrados de Ingeniería
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana – Cali

Con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el Trabajo de Grado y posteriormente optar por el título de Magíster en Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto de Trabajo de Grado denominado “Detección, georreferenciación y cuantificación de daños en pavimentos flexibles a partir de imágenes, utilizando aprendizaje profundo”, el cual fue realizado por el (los) estudiante (s) Manuel Alejandro Morales Acevedo con código (s) 8974546 pertenecientes perteneciente al énfasis en: N/A, bajo la dirección del profesor Julián Gil González

El suscrito director del Trabajo de Grado autoriza para que se proceda a hacer la evaluación de este Proyecto ante el Tribunal que para el efecto se designe, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado oficialmente.

Atentamente,



Manuel Alejandro Morales Acevedo



Julián Gil González

C.C. 1127615202 de CON-CAR-VEN C.C. 1088286439 de Pereira



David Alejandro Jiménez Sierra

C.C. 1094934464 de Armenia

Documentación anexa:

Resumen del Proyecto Aplicado en formato digital (máximo 1 página).
Una copia digital (PDF) del documento del proyecto aplicado



**ALCALDÍA
DE SAN JOSÉ DE
CÚCUTA**



CARTA DE AUTORIZACIÓN DE USO DE DATOS DE EMPRESA PARA OBTENCIÓN DE GRADO DE MAGISTER EN CIENCIA DE DATOS

Yo, ELIANA CONSTANZA MEDINA PABUENCE, en mi calidad de secretaria de Despacho de la Secretaría de Infraestructura, del Municipio de San José de Cúcuta, Departamento Norte de Santander otorgo la Autorización, Al señor MANUEL ALEJANDRO MORALES ACEVEDO identificado con C.C. 1.127.615.202 estudiante de la Maestría en Ciencia de Datos de la Pontificia Universidad Javeriana Cali para que utilice la siguiente información de la organización:

- Datos Geográficos de segmentos y calzadas de la malla vial urbana.
- Códigos de Identificación Vial
- Bases de Datos espaciales y alfanuméricas del inventario vial urbano
- Uso del software de Sistema de Información Geográfica de la secretaria de Infraestructura.

Con la finalidad de que pueda desarrollar su Trabajo de Investigación para optar el grado de Magister en Ciencia de Datos.

Título del Proyecto: DETECCIÓN, GEORREFERENCIACIÓN Y CUANTIFICACIÓN DE DAÑOS EN PAVIMENTOS FLEXIBLES A PARTIR DE IMÁGENES, UTILIZANDO APRENDIZAJE PROFUNDO.

Cordialmente,

Eliana C. Medina P.

ELIANA CONSTANZA MEDINA PABUENCE
Secretaria de Despacho
SECRETARÍA DE INFRAESTRUCTURA



Firmado digitalmente por Eliana Medina
Fecha: 2022.12.12 12:11:45 -05'00'
Versión de Adobe Acrobat: 2022.003.20282

Copia: NA
Anexos: NA

José Edgar Caicedo Fonseca	Profesional Universitario - SIM	
Los arriba firmantes expresamos que hemos preparado y examinado este documento y lo encontramos preciso a las disposiciones legales y/o técnicas aplicables vigentes; por lo tanto, bajo nuestra responsabilidad, lo presentamos a la firma de la Secretaria de Despacho de Infraestructura.		



**Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias**

**FICHA RESUMEN
TRABAJO DE GRADO DE MAESTRÍA**

TITULO: “Detección, georreferenciación y cuantificación de daños en pavimentos flexibles a partir de imágenes, utilizando aprendizaje profundo”

1. ÉNFASIS: N/A
2. TIPO DE PROYECTO: Aplicado
3. ÁREA DE TRABAJO: Infraestructura
4. ESTUDIANTE (S): Manuel Alejandro Morales Acevedo
5. CORREO ELECTRÓNICO: manuel130489m@javerianacali.edu.co
6. DIRECCIÓN Y TELÉFONO: Torres de Zulima, Bloque B, Apartamento 202, Cúcuta, Norte de Santander, 3138097491
7. DIRECTOR: Julián Gil González
8. VINCULACIÓN DEL DIRECTOR (en la universidad): Planta
9. CORREO ELECTRÓNICO DEL DIRECTOR: julian.gil@javerianacali.edu.co
10. CO-DIRECTOR(ES) (Si aplica): David Alejandro Jiménez Sierra
11. GRUPO O EMPRESA QUE LO AVALA (Si aplica): Secretaría de Infraestructura del Municipio San José de Cúcuta, Norte de Santander e Ingeniería Geomática Smart S.A.S.
12. OTROS GRUPOS O EMPRESAS: N/A
13. PALABRAS CLAVE (al menos 5): Imágenes, Aprendizaje Profundo, Vialidad, Daños en Pavimento.
14. ODS QUE APLICA EL PROYECTO (Agenda 2030): N/A
15. FECHA DE INICIO (Desarrollo del proyecto): 23/01/2023
16. RESUMEN (máximo 400 palabras): Las técnicas a utilizar para llevar a cabo esta tarea, es el uso de aprendizaje profundo, Visión Artificial para detectar objetos en imágenes. El problema para abordar está orientado a la seguridad vial, teniendo en cuenta la gran cantidad de accidentes relacionados con el mal estado y deterioro de las vías, la finalidad es, que mediante el uso de aprendizaje profundo realizar un diagnóstico del estado de la malla vial, determinar sitios críticos y su respectiva ubicación, definir las áreas prioritarias, para ser utilizando en la determinación del monto de inversión para mantenimiento y rehabilitación de las vías.



Pontificia Universidad
JAVERIANA
Cali

Detección, georreferenciación y cuantificación de daños en pavimentos flexibles a partir de imágenes, utilizando aprendizaje profundo.

Manuel Alejandro Morales Acevedo

Código 8974546

*Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos*

Director(a)

Julián Gil González

Codirector(a)

David Alejandro Jiménez Sierra

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, DICIEMBRE 08 DE 2023

TABLA DE CONTENIDO

1. <i>DEFINICION DEL PROBLEMA</i>	8
2. <i>OBJETIVOS DEL PROYECTO</i>	10
3. <i>ALCANCE</i>	12
4. <i>JUSTIFICACION</i>	13
5. <i>MARCO TEORICO</i>	14
6. <i>MATERIALES Y METODOS</i>	44
7. <i>RESULTADOS</i>	57
8. <i>CONCLUSIONES Y TRABAJOS FUTUROS</i>	92
9. <i>REFERENCIAS BIBLIOGRAFICAS</i>	94

LISTA DE FIGURAS

<i>Figura 1. Fisura longitudinal</i>	14
<i>Figura 2. Fisura transversal</i>	14
<i>Figura 3. Fisuras en bloque</i>	15
<i>Figura 4. Piel de cocodrilo</i>	15
<i>Figura 5. Bache</i>	16
<i>Figura 6. Parche</i>	16
<i>Figura 7. Arquitectura YOLOv4 para la detección de objetos.</i>	21
<i>Figura 8. Arquitectura YOLOv8 para la detección de objetos.</i>	23
<i>Figura 9. Interfaz del software LabelImg</i>	26
<i>Figura 10. Implementación de Homografía en Imágenes</i>	27
<i>Figura 11. Cámara Digital</i>	29
<i>Figura 12. Cámara de Acción insta360 One X2.</i>	30
<i>Figura 13. Interfaz de software Insta360 Studio.</i>	30
<i>Figura 14. Video en Formato 360.</i>	31
<i>Figura 15. Metadatos de Imagen.</i>	32
<i>Figura 16. Constelación de satélites GPS.</i>	33
<i>Figura 17. Proyección Cartográfica CMT12.</i>	34
<i>Figura 18. a) Azimut y b) Rumbo.</i>	34
<i>Figura 19. Modelo Vectorial.</i>	36
<i>Figura 20. Modelo Ráster.</i>	37
<i>Figura 21. Software ArcGIS.</i>	37
<i>Figura 22. Software ArcGIS Pro.</i>	38
<i>Figura 23. Herramienta GeoTagged Photos To Points.</i>	388
<i>Figura 24. Herramienta Spatial Join.</i>	39
<i>Figura 25. Software ArcGIS Online.</i>	40
<i>Figura 26. Software ArcGIS Dashboard.</i>	40
<i>Figura 27. Vías arteriales de San José de Cúcuta.</i>	58
<i>Figura 28. Sistema de Captura de Imágenes, con cámara insta 360 X2.</i>	58
<i>Figura 29. Imagen sin Corrección de Color.</i>	58
<i>Figura 30. Imagen con Corrección de Color.</i>	48
<i>Figura 31. Imagen sin corrección de Perspectiva.</i>	58
<i>Figura 32. Imagen con corrección de Perspectiva.</i>	48
<i>Figura 33. Interfaz de software labelImg.</i>	58

<i>Figura 34. Etiquetado de daños con el software labelImg</i>	49
<i>Figura 35. Vías arteriales de san José de Cúcuta desplegadas en ArcGIS Pro.</i>	52
<i>Figura 36. Vías arteriales de san José de Cúcuta desplegadas en ArcGIS Online.</i>	52
<i>Figura 37. Líneas de Ejes de vías y polígonos de calzadas.</i>	53
<i>Figura 38.. Georreferenciación de las imágenes con daños.</i>	54
<i>Figura 39 Medición de distancia de objeto sobre el terreno.</i>	55
<i>Figura 40. Medición de distancia de objeto en pixeles.</i>	56
<i>Figura 41. Unión de Atributos de daños detectados y código de calzada.</i>	57
<i>Figura 42. Gráfico de entrenamiento experimento 1.</i>	60
<i>Figura 43. Gráfico de entrenamiento experimento 2.</i>	62
<i>Figura 44. Gráfico de entrenamiento experimento 3.</i>	64
<i>Figura 45. Gráfico de entrenamiento experimento 4. Yolov4.</i>	67
<i>Figura 46. Gráfico de entrenamiento experimento 4. Yolov8.</i>	68
<i>Figura 47. Detecciones utilizando modelo entrenado del experimento 1</i>	72
<i>Figura 48. Detecciones utilizando modelo entrenado del experimento 2</i>	74
<i>Figura 49. Detecciones utilizando modelo entrenado del experimento 3</i>	77
<i>Figura 50. Detecciones utilizando modelo entrenado con yolo v4 del experimento 4</i>	80
<i>Figura 51. Detecciones utilizando modelo entrenado con yolo v8 del experimento 4.</i>	82
<i>Figura 52. Coordenadas y Azimut de la Cámara en cada fotografía</i>	86
<i>Figura 53. Daños detectados y georreferenciados</i>	88
<i>Figura 54. Publicación de capas geográficas en ArcGIS Online</i>	90
<i>Figura 55. Cuadro de mando daños detectados</i>	91

LISTA DE ECUACIONES

<i>Ecuación 1. Calculo de Precision.</i>	24
<i>Ecuación 2. Cálculo de Recall</i>	24
<i>Ecuación 3. Calculo de F1 Score</i>	25
<i>Ecuación 4. Calculo de Mean Average Precision</i>	25
<i>Ecuación 5. Cálculo de coordenada x.</i>	35
<i>Ecuación 6. Cálculo de coordenada y.</i>	35
<i>Ecuación 7. Formula Para calcular Distancia por pixel.</i>	35

LISTA DE TABLAS

<i>Tabla 1. Relación de imágenes del dataset RDD2022.</i>	58
<i>Tabla 2. Cantidad de etiquetas del dataset DSCUCU1.</i>	58
<i>Tabla 3. Precisiones de ejecución de datos de experimento 1.</i>	60
<i>Tabla 4. Cantidad de etiquetas del dataset DSCUCU2.</i>	61
<i>Tabla 5. Precisiones de ejecución de datos de experimento 2.</i>	63
<i>Tabla 6. Cantidad de etiquetas del dataset DSCUCU3.</i>	63
<i>Tabla 7. Precisiones de ejecución de datos de experimento 3.</i>	65
<i>Tabla 8. Cantidad de etiquetas del dataset DSCUCU4.</i>	65
<i>Tabla 9. Precisiones de ejecución de datos de experimento 4 Yolo v4.</i>	67
<i>Tabla 10. Precisiones de ejecución de datos de experimento 4 Yolo v8.</i>	69
<i>Tabla 11. Precisiones de ejecución de datos de experimento 1.</i>	70
<i>Tabla 12. Resultados globales del modelo de experimento 1.</i>	71
<i>Tabla 13. Precisiones de ejecución de datos de experimento 2.</i>	72
<i>Tabla 14. Resultados globales del modelo de experimento 2.</i>	73
<i>Tabla 15. Precisiones de ejecución de datos de experimento 3.</i>	75
<i>Tabla 16. Resultados globales del modelo de experimento 3.</i>	76
<i>Tabla 17. Precisiones de ejecución de datos de experimento 4. Yolov4.</i>	78
<i>Tabla 18. Resultados globales del modelo de experimento 4.</i>	78
<i>Tabla 19. Precisiones de ejecución de datos de experimento 4. Yolov8.</i>	80
<i>Tabla 20. Precisiones mAP de Modelo entrenado yolov4.</i>	83
<i>Tabla 21. Precisiones mAP de Modelo entrenado yolov8.</i>	84
<i>Tabla 22. Cantidad de detecciones realizadas por el modelo entrenado con YOLO v8</i>	85
<i>Tabla 23 Tabla con distancias, rumbos y áreas de los daños detectados</i>	87
<i>Tabla 24 Distancias de Anchos y Largos de los bounding box de daños detectados.</i>	89

INTRODUCCIÓN

El deterioro de la superficie de la carretera debido a factores que incluyen la sobrecarga de los vehículos, la mala calidad de la construcción, el envejecimiento, los desastres naturales y condiciones climáticas pueden provocar daños en el pavimento de la carretera. Este deterioro produce lentitud en el tráfico generando atascos, daños en los vehículos y accidentes de tránsito. Esto también causa problemas a las autoridades de los gobiernos locales que necesitan identificar con precisión los daños y realizar el trabajo de reparación. Si estos daños no son reparados en las primeras etapas, el costo de restauración aumenta gradualmente, lo que genera una carga innecesaria para el presupuesto de la entidad gubernamental a cargo de las vías.

Mediante el aprendizaje profundo, es posible automatizar el proceso de detección, de daños en pavimentos flexibles a partir de imágenes, lo que facilita y agiliza el seguimiento del estado de las superficies de las carreteras. Esto podría conducir a reparaciones y rehabilitaciones más eficientes y rentables y, en última instancia, caminos más seguros para todos.

En este proyecto proponemos el uso de aprendizaje profundo desarrollando un modelo destinado a detectar con precisión diferentes tipos de daños en las imágenes, así como a estimar el alcance y la gravedad del daño. También implementaremos un sistema de georreferenciación de la ubicación de los daños en la red vial.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La infraestructura vial es un activo público crucial, ya que contribuye al desarrollo y crecimiento económico al mismo tiempo que brinda beneficios sociales críticos. Las vías conectan centros poblados, permiten la comercialización de productos y el acceso a la educación, empleo, servicios sociales y de salud. Sin embargo, la superficie de la carretera sufre desgaste y deterioro con el tiempo debido a factores relacionados con la ubicación, tiempo de construcción, volumen de tráfico, clima y materiales utilizados para la construcción; por lo tanto, el conocimiento de su grado de deterioro es fundamental para el mantenimiento eficiente y rentable con el objetivo de preservar su buen estado y aumentar la seguridad para los diferentes actores viales [1][2].

Existen diferentes causas potenciales que expliquen las malas condiciones de las carreteras en Colombia. Entre estas causas se identifican, la falta de financiamiento, la infraestructura, los estándares de construcción, y los desastres naturales. El mantenimiento y la reparación de las vías requieren de una inversión significativa, que puede no estar disponible en algunas áreas debido a presupuestos limitados u otras prioridades; como resultado, es posible que las carreteras no reciban el mantenimiento adecuado y se deterioren con el tiempo. Por otro lado, las vías mal diseñadas y construidas son más susceptibles a daños dado que no pueden soportar las cargas pesadas y el tráfico al que están sujetas. Lo anterior, puede provocar fallas prematuras y la necesidad de reparaciones frecuentes. Además, desastres naturales como terremotos, inundaciones y deslizamientos de tierra causan daños significativos a las carreteras y otras infraestructuras. Esto da lugar a la necesidad de reparaciones costosas y cierres temporales o permanentes de las carreteras. A medida que la población y el crecimiento económico crecen, también aumenta la cantidad de tráfico en las calles. Esto conduce a un mayor desgaste de las vías y dificultan la realización de trabajos de mantenimiento y reparación [3] [4] [5] [6].

Los pavimentos flexibles, como el asfalto, se utilizan comúnmente para superficies de carreteras. Estos pavimentos están sujetos a varios tipos de daños, como fisuras, piel de cocodrilo y baches [7]. Detectar, georreferenciar y cuantificar estos daños es importante para el mantenimiento y reparación de vías. En este trabajo seleccionamos pavimentos flexibles como superficie a inspeccionar, debido a que el mayor porcentaje de las vías, se encuentran construidas en este material [8]. La inspección manual de grandes redes de carreteras requiere largos periodos y considerables cantidades recursos económicos; la inspección manual consiste en realizar una inspección visual de la superficie de la carretera, tomar medidas de las dimensiones de los daños, esta tarea debe realizarse por profesionales especializados e incluye tomar precauciones adicionales por estar expuesto a situaciones de inseguridad alta, debido a que la inspección se realiza sobre la calzada de la vía.

Así, las técnicas de aprendizaje automatizado son una alternativa a la inspección manual. Específicamente, mediante el uso de aprendizaje profundo, es posible detectar y cuantificar daños en pavimentos flexibles a partir de imágenes [9]. Dado un conjunto de imágenes de pavimentos flexibles, necesitamos desarrollar un modelo de aprendizaje profundo que pueda detectar y clasificar con precisión diferentes tipos de daños, como fisuras, piel de cocodrilo y baches. Adicionalmente, cuantificar las dimensiones de los daños detectados, así como también la correcta georreferenciación y vinculación a los segmentos de la malla vial para establecer la severidad de los daños en los sectores seleccionados para este estudio.

Sin embargo, desde la perspectiva de la ciencia de datos, existen algunos desafíos que deben abordarse para mejorar la precisión y confiabilidad de esta tecnología. Estos desafíos incluyen la gestión de conjuntos de datos complejos con variaciones significativas, la falta de interpretabilidad en modelos de aprendizaje profundo, la necesidad de avanzar en métodos de aprendizaje no supervisado y la búsqueda de eficiencia en tiempo real y uso de memoria, especialmente en dispositivos con recursos limitados [10].

1.2. FORMULACIÓN DEL PROBLEMA

¿Cómo utilizar el aprendizaje profundo para automatizar el proceso de detección, georreferenciación y cuantificación de daños en pavimentos flexibles a partir de imágenes? ¿Qué características se deben considerar relevantes al momento de crear una base de datos de imágenes de pavimentos flexibles? ¿Qué técnicas deben ser seleccionadas para que aporten adecuadamente al entrenamiento del modelo de aprendizaje profundo para detectar y clasificar diferentes tipos de daños en las imágenes? ¿De qué manera Implementar un sistema de georreferenciación de la ubicación de los daños en la red vial? ¿Cómo valorar el nivel de desempeño del modelo seleccionado? ¿Cuáles son las aplicaciones y limitaciones del modelo?

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Desarrollar un sistema de visión por computador a partir de modelos de aprendizaje profundo para la detección y cuantificación de daños en pavimentos flexibles.

2.2. OBJETIVOS ESPECÍFICOS

1. Construir una base de datos con imágenes de pavimentos flexibles con diferentes tipos de averías.
2. Implementar un modelo de aprendizaje profundo para detectar y clasificar diferentes tipos de daños en las imágenes.
3. Evaluar el desempeño del modelo usando métricas apropiadas para el aprendizaje supervisado.
4. Implementar un sistema de georreferenciación de la ubicación de los daños en la red vial.

2.3. RESULTADOS ESPERADOS

A continuación, enumero los resultados esperados asociados a los objetivos específicos:

1. Construir una base de datos de imágenes de pavimentos flexibles.
 - Creación de la base de datos de imágenes de pavimentos flexibles con etiquetas que indiquen el tipo de avería.
2. Entrenar un modelo de aprendizaje profundo para detectar y clasificar diferentes tipos de daños en las imágenes.
 - Definición del modelo de aprendizaje profundo más adecuado y ejecución con los datos

de entrenamiento.

3. Evaluar el desempeño del modelo usando métricas apropiadas.

- Evaluación del modelo y uso en entorno de producción

4. Implementar un sistema de georreferenciación de la ubicación de los daños en la red vial.

- Definición del sistema de georreferenciación, localización y cuantificación de los daños identificados.

El resultado esperado de este proyecto es un modelo de aprendizaje profundo que puede detectar, georreferenciar y cuantificar con precisión el daño en pavimentos flexibles a partir de imágenes. El modelo debe poder detectar y clasificar diferentes tipos de daños, como fisuras, piel de cocodrilo y baches.

3. ALCANCE

Este proyecto se enfoca en el desarrollo de un modelo de aprendizaje profundo para detectar, georreferenciar y cuantificar daños en pavimentos flexibles a partir de imágenes. El alcance del proyecto incluye lo siguiente:

- Construir una base de datos de imágenes de pavimentos flexibles. El proyecto abarcará solo las vías arteriales y zonales de la ciudad, la longitud total de estos tramos es de 285 Km, de los cuales 240 Km están construidos en pavimento flexible, esto representa un 85% del total de las vías arteriales y zonales. Para este proyecto se estima que se obtengan alrededor de 56 000 imágenes geoetiquetadas en formato .jpg
- Entrenar un modelo de aprendizaje profundo para detectar y clasificar diferentes tipos de daños en las imágenes. Para entrenar el modelo se trabajará con etiquetas de 5 tipos de daños.
- Evaluar el desempeño del modelo usando métricas apropiadas para el aprendizaje supervisado. Se ejecutará y evaluará el modelo haciendo uso de equipos de cómputo de escritorio disponibles.
- Implementar un sistema de georreferenciación de la ubicación de los daños en la red vial. La precisión de las coordenadas de la georreferenciación de los datos geográficos será a nivel métrico 3-5 metros.

El alcance del proyecto no incluye el desarrollo de un sistema de captura de imágenes de la red vial, ni la reparación de los daños en la vía. Estas tareas están más allá del alcance de este proyecto.

4. JUSTIFICACIÓN

El desarrollo de una herramienta de detección, georreferenciación y cuantificación de daños en pavimentos flexibles ayudaría a las autoridades responsables del mantenimiento de las carreteras a la identificación temprana y reparación de problemas en el pavimento evitando que evolucionen hacia complicaciones más significativas. Contribuiría a realizar una mejor planificación de mantenimiento y reparación de las carreteras, lo que podría ahorrar tiempo y dinero. Además, beneficia a los diferentes actores viales mejorando la seguridad, al contribuir a la reducción de accidentes causados por el mal estado del pavimento.

Actualmente existe un gran interés en el uso de inteligencia artificial para resolver una gran variedad de problemas, incluyendo dificultades relacionadas con la detección y análisis de imágenes [11]. El desarrollo de una herramienta de detección, georreferenciación y cuantificación de daños en pavimentos flexibles contribuye en la precisión de la detección de daños, reducción significativa en el tiempo de ejecución y en los costos de la realización del proyecto [12]. Hoy en día hay una gran variedad de estudios en donde se realizaron detecciones del estado de la superficie de las vías a partir de imágenes, con alcances similares a los planteados en este proyecto [13] [14] [15].

Al implementar una herramienta de detección, georreferenciación y cuantificación de daños en pavimentos flexibles se busca mejorar la eficiencia del mantenimiento y la reparación de carreteras, lo que permite ahorrar tiempo y recursos financieros. Mejorar la seguridad de las diferentes vías al identificar y reparar problemas en el pavimento antes de que se conviertan en problemas mayores. Finalmente, podría contribuir a una mejor gestión del sistema de carreteras en general, lo que podría tener un impacto positivo en la economía y el bienestar de la sociedad.

5. MARCOTEÓRICO

5.1. MARCO TEÓRICO

5.1.1 Pavimentos.

Los pavimentos son diseñados para soportar cargas vehiculares proyectadas en un determinado tiempo o vida útil, pero existen algunos factores que pueden acelerar el deterioro de las vías, como los cambios drásticos de temperatura, humedad, errores en los diseños de las capas o de la mezcla asfáltica, falla de taludes, falla de obras de contención, falta de confinamiento lateral, entre otras. A partir de estos deterioros prematuros de la estructura vial nace la necesidad de realizar inspecciones rutinarias que permitan conocer el estado de la vía y determinar si se requiere algún tipo de intervención [16].

5.1.2. Daños en pavimentos

Existen diferentes tipos de daños y cada uno de ellos señala alguna dificultad que está presentando la carpeta asfáltica o la estructura vial, por esto es de gran importancia hacer una clasificación adecuada del tipo de falla, ya que de lo contrario se podría incurrir en sobrecostos por el tratamiento de algún daño de forma inadecuada [16].

Los daños que presenta una estructura de pavimento flexible pueden ser clasificados en las siguientes categorías:

- Fisuras
- Deformaciones
- Pérdida de capas estructurales

- Daños superficiales
- Otros daños

Dentro de cada categoría existen diferentes deterioros que se originan por diversos factores, algunos de los cuales se han establecido mediante la revisión bibliográfica, y otros mediante evaluación de campo y ensayos de laboratorio [17].

Para efectos de este proyecto, trabajaremos con las categorías de fisuras y deformaciones.

5.1.3. Tipos de daños

5.1.3.1. Fisuras.

5.1.3.1.1. Fisuras longitudinales y transversales (FL, FT). Corresponden a discontinuidades en la carpeta asfáltica, en la misma dirección del tránsito o transversales a él [17].

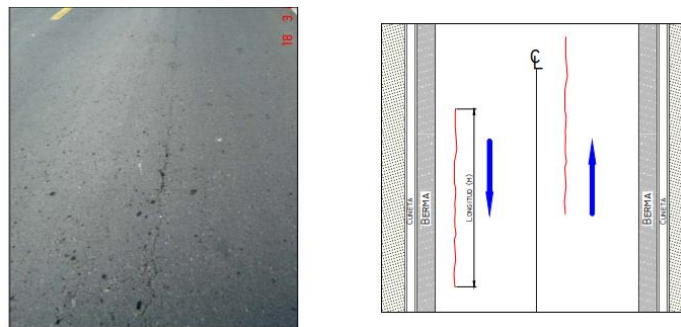


Figura 1. Fisura longitudinal (FL, Unidad de medida: m) [17].

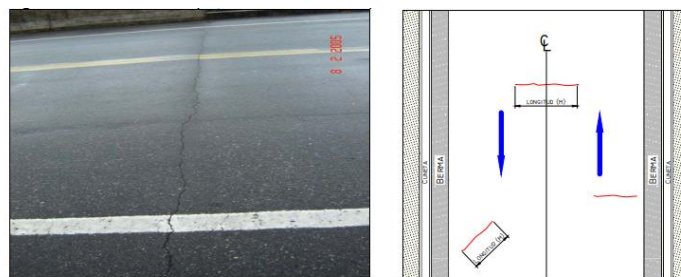


Figura 2. Fisura transversal (FT, Unidad de medida: m) [17].

5.1.3.1.2. Fisuras en bloque (FB). Cuando se presenta este tipo daño la superficie del asfalto es dividida en bloques de forma aproximadamente rectangular. Los bloques tienen lado promedio mayor que $0,30 \text{ m}^3$ [17].

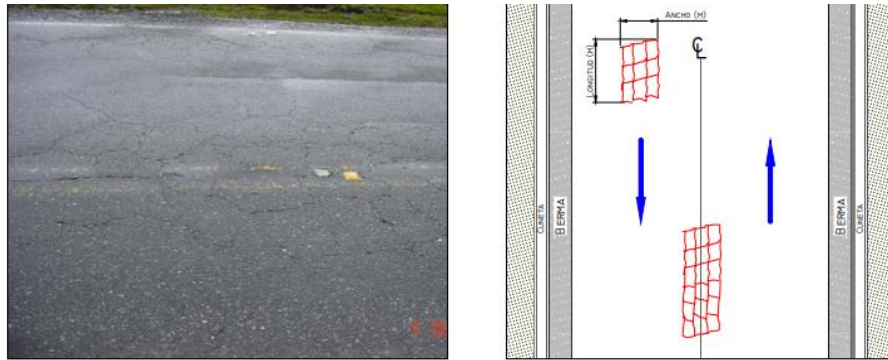


Figura 3. Fisuras en bloque (FB, Unidad de medida: m^2) [17].

5.1.3.1.3. Piel de cocodrilo (PC). Corresponde a una serie de fisuras interconectadas con patrones irregulares, generalmente localizadas en zonas sujetas a repeticiones de carga. La fisuración tiende a iniciarse en el fondo de las capas asfálticas, donde los esfuerzos de tracción son mayores bajo la acción de las cargas [17].



Figura 4. Piel de cocodrilo (PC, Unidad de medida: m^2) [17].

5.1.3.2. Pérdida de las capas de la estructura.

5.1.3.2.1 Baches (BCH). Desintegración total de la carpeta asfáltica que deja expuestos los materiales granulares lo cual lleva al aumento del área afectada y al aumento de la profundidad debido a la acción del tránsito [17].

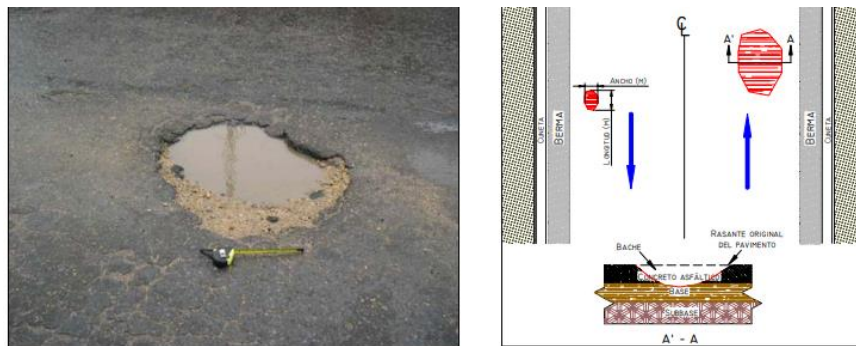


Figura 5. Bache (BCH, Unidad de medida: m^2) [3]

5.1.3.2.2. Parche (PCH). Los parches corresponden a áreas donde el pavimento original fue removido y reemplazado por un material similar o diferente, ya sea para reparar la estructura (a nivel de concreto asfáltico o hasta los granulares) o para permitir la instalación o reparación de alguna red de servicios (acueducto, gas, etc.) [17].

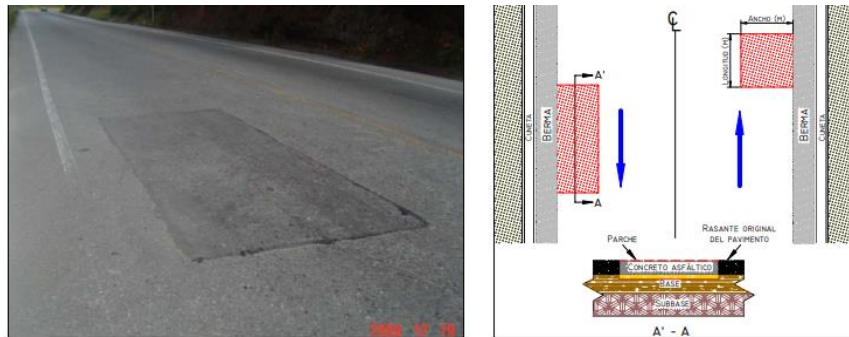


Figura 6. Parche (PCH, Unidad de medida: m^2) [17].

5.1.4. Aprendizaje Automático

Es una aplicación de Inteligencia Artificial que proporciona a los sistemas informáticos la capacidad de aprender y mejorar de la experiencia sin explícita programación. Una vez que se entrena un algoritmo informático en Aprendizaje Automático, puede aplicar la relación aprendida durante el entrenamiento para resolver problemas similares [18].

5.1.4.1. Aprendizaje Profundo.

Es una rama del aprendizaje automático, que es uno de los mayores avances y focos de investigación en aprendizaje automático en los últimos años [19]. El aprendizaje profundo es un tipo de aprendizaje automático que utiliza redes neuronales con varias capas de unidades de procesamiento para aprender patrones y características complejos a partir de los datos [20]. Es un subconjunto de la inteligencia artificial (IA) que es particularmente adecuado para manejar datos no estructurados de alta dimensión, como imágenes, texto y audio [21]. Los algoritmos de aprendizaje profundo se entrenan utilizando grandes conjuntos de datos y técnicas de optimización iterativa, y pueden aprender a hacer predicciones y decisiones basadas en los datos sin ser programados explícitamente [22].

5.1.4.1.2. Redes Neuronales Artificiales

Las redes neuronales son un tipo de algoritmo de aprendizaje automático que se inspiran en la estructura y función del cerebro [23]. Consisten en nodos o neuronas interconectados, que procesan y transmiten información mediante conexiones ponderadas. Las redes neuronales son capaces de aprender de los datos y hacer predicciones o decisiones basadas en esos datos [24]. Son ampliamente utilizados en aplicaciones tales como reconocimiento de imagen y voz, procesamiento de lenguaje natural y sistemas de recomendación [25].

5.1.5. PyTorch

PyTorch es un framework de aprendizaje automático de código abierto para entrenar redes neuronales profundas. Desarrollado por Meta en 2016, se ha convertido en uno de los frameworks más populares en la investigación. Se destaca por su accesibilidad, desarrollo rápido de modelos y tiempos de entrenamiento eficientes [26] [27] [28] [29] [30].

5.1.6. Google Colab y GPUs

Google Colab es una plataforma de código abierto basada en la nube que permite a los usuarios escribir y ejecutar código de Python en un navegador. Es ampliamente utilizado para aprendizaje automático, análisis de datos y educación. Colab proporciona acceso gratuito a recursos informáticos, incluyendo GPUs y TPUs, lo que facilita el trabajo con conjuntos de datos grandes y modelos computacionalmente intensivos [31].

En Google Colab, los usuarios de pago pueden elegir entre una GPU estándar o premium, lo que les permite mejorar su GPU cuando necesitan más potencia. Las GPU estándar son típicamente NVIDIA T4 Tensor Core GPUs, mientras que las GPU premium son generalmente NVIDIA V100 o A100 Tensor Core GPUs. La asignación de un tipo específico de chip GPU no está garantizada y depende de varios factores, incluyendo la disponibilidad y el saldo pagado con Colab [31].

5.1.7. You Only Look Once (YOLO)

El algoritmo You Only Look Once (YOLO) es un método de aprendizaje profundo para la detección y clasificación de objetos en tiempo real [32]. Es un detector de disparo único que utiliza una red neuronal convolucional (CNN) para predecir cuadros delimitadores y probabilidades de clase para objetos en imágenes [33]. El algoritmo YOLO procesa la imagen completa en un paso hacia adelante de la CNN y hace predicciones para múltiples objetos en un solo paso. Esto permite una detección de objetos rápida y eficiente, lo que lo hace adecuado para aplicaciones como vigilancia y vehículos autónomos [34].

5.1.8. Darknet

Darknet es una plataforma de código abierto para redes neuronales escrita en C y CUDA,

utilizada para implementaciones avanzadas de redes neuronales profundas, incluyendo YOLO para detección de objetos en tiempo real [35] [36] [37].

5.1.9. YOLOv4

YOLOv4 es un modelo avanzado para la detección de objetos en tiempo real, que se destaca por su velocidad y precisión óptimas. Es la cuarta versión de la serie de modelos YOLO (You Only Look Once) y fue desarrollada por Alexey Bochkovskiy, Chien-Yao Wang y Hong-Yuan Mark Liao [38].

La arquitectura de YOLOv4 es una combinación innovadora de varios componentes y técnicas que trabajan en conjunto para maximizar la precisión y eficiencia en la detección de objetos en tiempo real. La estructura general del modelo se divide en tres componentes principales: el backbone, el neck y el head. Cada uno de estos componentes desempeña un papel crucial en el proceso de detección de objetos [38].

- Backbone: CSPDarknet53

El backbone es la columna vertebral del modelo y se encarga de la extracción de características fundamentales de las imágenes de entrada. En YOLOv4, el backbone utilizado es CSPDarknet53, una red neuronal diseñada específicamente para la eficiencia y efectividad en la detección de objetos. CSPDarknet53 utiliza conexiones residuales ponderadas y una arquitectura optimizada para maximizar la velocidad y precisión en la extracción de características. La elección de CSPDarknet53 se basa en su capacidad para equilibrar el tamaño del campo receptivo, el número de parámetros y la cantidad de salidas de capa, lo que lo convierte en una opción ideal para la detección de objetos en diversos tamaños y contextos [38].

- Neck: SPP y PAN

El neck actúa como un intermediario entre el backbone y el head, realizando funciones adicionales de procesamiento y mejora de las características extraídas. En YOLOv4, el neck está compuesto por dos módulos clave: el Spatial Pyramid Pooling (SPP) y el Path Aggregation Network (PAN). El SPP amplía el campo receptivo y separa las características contextuales más significativas, mientras que el PAN agrega parámetros de diferentes niveles del backbone para una detección más efectiva. Esta combinación resulta en una mejora notable en la capacidad de capturar y procesar información relevante para la detección de objetos [38].

- Head: Basado en YOLOv3

El head es responsable de realizar las predicciones finales, incluyendo la clasificación de objetos y la localización de sus cajas delimitadoras. YOLOv4 utiliza un head basado en YOLOv3, que es conocido por su eficiencia y precisión en la generación de cajas delimitadoras y la clasificación de objetos. El head aprovecha las características procesadas por el backbone y el neck para realizar predicciones rápidas y precisas, lo que es esencial para la detección de objetos en tiempo real [38]. En la figura 7, podemos apreciar la arquitectura del modelo yolo v4.

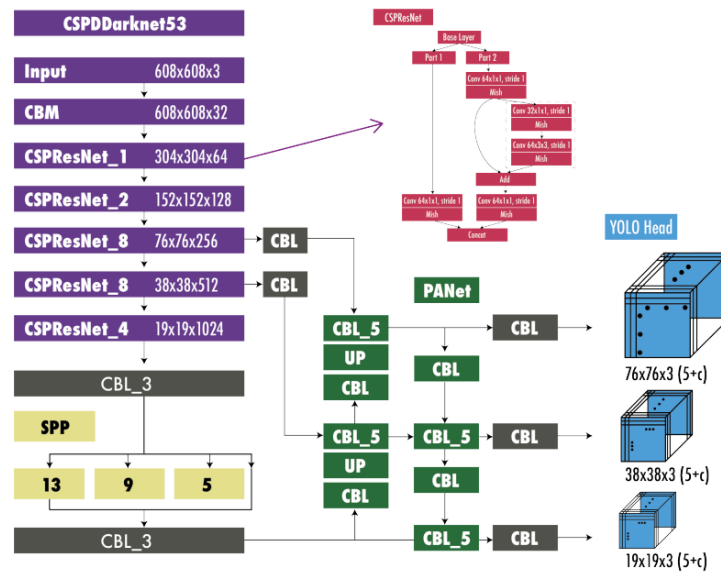


Figura 7. Arquitectura YOLOv4 para la detección de objetos. Los módulos en el diagrama son CMB : Convolución + Normalización por lotes + Activación Mish, CBL: Convolución + Normalización por lotes + Leaky ReLU, UP: upsampling, SPP: Agrupación Piramidal Espacial y PANet: Red de Agregación de Trayectorias [39].

5.1.9.1. Parámetros de entrenamiento del modelo YOLOv4

Para realizar el entrenamiento de un custom dataset utilizando el modelo yolo v4 se recomienda utilizar los siguientes parámetros de entrenamiento.

- batch: Determina cuántas imágenes se utilizarán en un solo paso durante el entrenamiento. Valor recomendado: batch=64
- subdivisions: Divide el batch en un número más manejable de imágenes para evitar problemas de memoria. Valor recomendado: subdivisions=16
- max_batches: Es el número total de pasos (batches) durante el entrenamiento. Es crucial tener suficientes batches para que el modelo pueda aprender adecuadamente. Valor recomendado: max_batches = clases * 2000.

- **steps:** Son los puntos durante el entrenamiento en los que la tasa de aprendizaje se reduce. Valor recomendado: $\text{steps} = 80\%$ y 90% de max_batches . Con un max_batches de 6000
- **width y height:** Establecen el tamaño de las imágenes que ingresan a la red. Deben ser múltiplos de 32. Valor recomendado: $\text{width}=416$ y $\text{height}=416$.
- **classes:** Indica el número de clases (objetos) que queremos detectar. Valor recomendado: $\text{classes}=\text{n}$, donde n es el número de clases.
- **filters:** Es el número de filtros en la capa convolucional antes de cada capa [yolo]. Se calcula como $(\text{classes} + 5) * 3$ [40].

5.1.10. YOLOv8

YOLOv8 fue lanzado en enero de 2023 por Ultralytics, la empresa que desarrolló YOLOv5. YOLOv8 ofrece cinco versiones escaladas: YOLOv8n (nano), YOLOv8s (pequeño), YOLOv8m (mediano), YOLOv8l (grande) y YOLOv8x (extra grande). Es compatible con múltiples tareas de visión como detección de objetos, segmentación, estimación de pose, seguimiento y clasificación [39].

La arquitectura de YOLOv8 utiliza un backbone similar al de YOLOv5 con algunas modificaciones en la CSPLayer, ahora llamada módulo C2f. Este módulo (cross-stage partial bottleneck con dos convoluciones) combina características de alto nivel con información contextual para mejorar la precisión de detección [39].

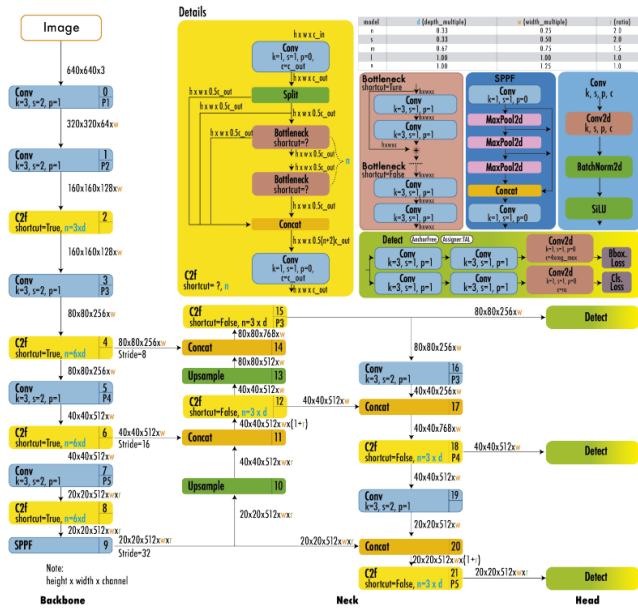


Figura 8. Arquitectura YOLOv8. Esta arquitectura utiliza una columna vertebral CSPDarknet53 modificada. El módulo C2f reemplaza la capa CSPLayer utilizada en YOLOv5. Una capa de agrupación piramidal espacial rápida (SPPF) acelera el cálculo al agrupar características en un mapa de tamaño fijo. Cada convolución cuenta con normalización por lotes y activación SiLU. La cabeza está desacoplada para procesar independientemente la detección de objetos, clasificación y tareas de regresión. Extraído de [39].

YOLOv8 usa un modelo sin anclas con una cabeza desacoplada para procesar independientemente la clasificación de la objetividad y las tareas de regresión. Este diseño permite que cada rama se concentre en su tarea y mejore la precisión general del modelo. En la capa de salida de YOLOv8, se utiliza la función de activación sigmoidea para la puntuación de objetividad, representando la probabilidad de que el cuadro delimitador contenga un objeto. Usa la función softmax para las probabilidades de clase, representando las probabilidades de los objetos de pertenecer a cada clase posible [39].

5.1.10.1. Parámetros de entrenamiento del modelo YOLOv8

Para realizar el entrenamiento de un custom dataset utilizando el modelo yolo v8 se emplean los siguientes parámetros de entrenamiento con sus respectivos valores recomendados.

- Epochs: Número de épocas para entrenar. Valor recomendado: 120
- Patience: Épocas de espera sin mejora observable para detener prematuramente el entrenamiento. Valor recomendado: Patience = 50
- Batch: Número de imágenes por lote. Valor recomendado: Batch =12
- Imgsz: Tamaño de las imágenes de entrada como entero. Valor recomendado: 640
- Device: Dispositivo para ejecutar, por ejemplo, cuda device=0 o device=0,1,2,3 o device=cpu. Valor recomendado: device=0
- Workers: Número de hilos de trabajadores para la carga de datos. Valor recomendado: Workers = 2 [41] [42].

5.1.11. Las métricas utilizadas en YOLO (You Only Look Once)

5.1.11.1. Precision (Precisión): Es una medida de la exactitud, que se calcula como la proporción de verdaderos positivos (objetos correctamente identificados) entre todos los objetos que el modelo ha clasificado como positivos (correctos e incorrectos) [43].

$$\text{Precision} = \frac{TP}{TP + FP}$$

Ecuación 1. Cálculo de Precision.

Donde TP son verdaderos positivos y FP son falsos positivos.

5.1.11. 2. Recall (Sensibilidad): Mide la capacidad del modelo para encontrar todos los casos relevantes dentro de un conjunto de datos. Se calcula como la proporción de verdaderos positivos dividida por la suma de verdaderos positivos y falsos negativos [43].

$$\text{Recall} = \frac{TP}{TP + FN}$$

Ecuación 1. Cálculo de Recall.

Donde TP son verdaderos positivos y FN son falsos negativos.

5.1.11.3. F1 Score: Es una medida que combina la precisión y el recall en un solo número. Es útil cuando necesitas un equilibrio entre precisión y recall. Se calcula como el promedio armónico de la precisión y el recall [44].

$$F1 \text{ Score} = 2 * \frac{\textit{presion} * \textit{recall}}{\textit{presion} + \textit{recall}}$$

Ecuación 3. Cálculo de F1 Score.

5.1.11.4. mAP.50 (mean Average Precision): Esta es una medida de la precisión promedio en múltiples umbrales de IoU (Intersection over Union). El ".50" se refiere a un umbral de IoU del 50%, lo que significa que una detección se considera correcta si el área de intersección sobre la unión con la verdad de tierra es al menos del 50% [45].

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Ecuación 4. Cálculo de mean Average Precision.

Donde N = el número de clases y AP_i = la precisión promedio de la clase i.

5.1.12. Software de Etiquetado LabelImg,

LabelImg es una herramienta de código abierto y gratuita para el etiquetado gráfico de imágenes, escrita en Python y que utiliza QT para su interfaz gráfica. Su facilidad de uso la convierte en una opción ideal para etiquetar cientos de imágenes y probar proyectos de detección de objetos. LabelImg admite el etiquetado en formatos de archivo VOC XML o YOLO text [46].

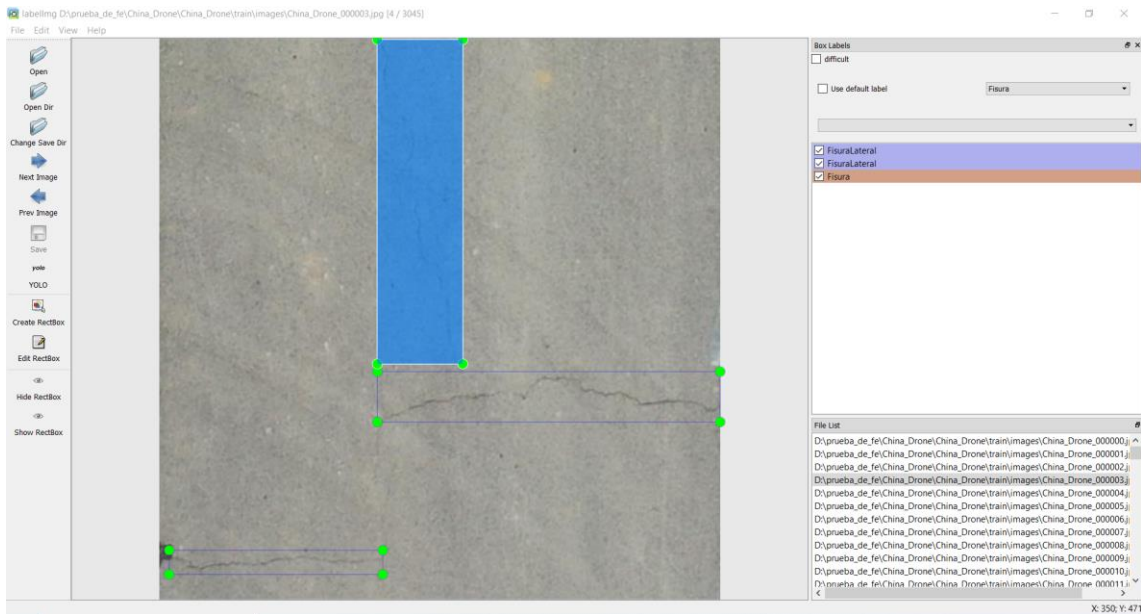


Figura 9. Interfaz del software LabelImg, Fuente: Elaboración propia.

5.1.13 Formato de etiquetas YOLO

El formato YOLO (You Only Look Once) se utiliza en la detección de objetos en imágenes. Las etiquetas en este formato se guardan en archivos .txt, con un archivo por imagen. Cada fila del archivo representa un objeto con la estructura: clase centro_x centro_y ancho alto, donde las coordenadas y dimensiones están normalizadas entre 0 y 1. No se necesita un archivo si la imagen no contiene objetos [47] [48] [49] [50].

5.1.14 Formato de etiquetas Pascal VOC

El formato Pascal VOC (Visual Object Classes), creado originalmente para el Visual Object Challenge (VOC), se utiliza ampliamente para etiquetas de detección de objetos. Este formato utiliza archivos XML, donde cada objeto se describe dentro de una estructura <object> que incluye detalles como el nombre del objeto, su posición (xmin, xmax, ymin, ymax en píxeles), entre otros. Esta información se encuentra dentro de una estructura mayor <annotation> que también incluye detalles del archivo de imagen [51].

5.1.15. Procesamiento Digital de Imágenes

El procesamiento de imágenes digitales es el uso de algoritmos y técnicas informáticas para manipular, analizar y comprender imágenes digitales [52]. Implica la aplicación de métodos matemáticos y computacionales para extraer información y características de las imágenes y para mejorar o modificar las imágenes para varios propósitos [53]. El procesamiento de imágenes digitales se puede aplicar a una amplia gama de campos, incluida la medicina, la ingeniería y la robótica, y tiene muchas aplicaciones prácticas, como imágenes médicas, sensores remotos y vigilancia [54].

5.1.16. Homografía

La homografía de imágenes es un concepto matemático utilizado en procesamiento de imágenes y visión por computadora para relacionar dos imágenes de la misma escena tomadas desde diferentes puntos de vista [55]. Es una transformación que asigna puntos en una imagen a los puntos correspondientes en la otra imagen y está representada por una matriz de 3×3 [56]. La homografía de imágenes se usa comúnmente para alinear imágenes, corregir distorsiones geométricas y realizar reconstrucciones 3D [57].

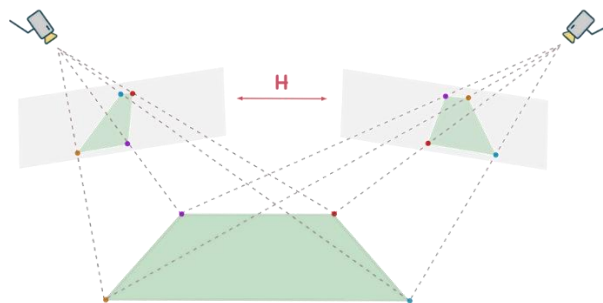


Figura 10. Implementación de Homografía en Imágenes Fuente: <https://towardsdatascience.com/>

5.1.17. Métodos Básicos de Aumentación de Datos

Las manipulaciones básicas de imágenes se enfocan en transformaciones de imagen, como rotación, volteo y recorte, entre otras. La mayoría de estas técnicas manipulan las

imágenes directamente y son fáciles de implementar. Los métodos considerados se presentan con Volteo, Rotación, Relación de Escala, Inyección de Ruido, Espacio de Color, Contraste, Afilado, Traslación, Recorte [58].

5.1.18. Clasificación supervisada

La clasificación supervisada en imágenes es una técnica utilizada en el aprendizaje automático y el procesamiento de imágenes para clasificar objetos o píxeles en una imagen en función de los datos de entrenamiento etiquetados [59]. En la clasificación supervisada, se entrena un modelo usando un conjunto de datos de imágenes con etiquetas conocidas, como el tipo de objeto o la cobertura terrestre en la imagen. El modelo aprende a asociar las características visuales de las imágenes con las etiquetas correspondientes y luego se puede utilizar para predecir las etiquetas de nuevas imágenes [60]. La clasificación supervisada se usa comúnmente en aplicaciones tales como mapeo de cobertura terrestre, imágenes médicas y reconocimiento de objetos [61].

5.1.18. Equipos y herramientas

5.1.18.1. Cámaras digitales

Las cámaras digitales se han convertido en una herramienta importante en el campo de la detección de objetos mediante el aprendizaje profundo. Ahora es posible recopilar grandes cantidades de imágenes de alta calidad para entrenar y evaluar modelos de detección de objetos.

Una de las principales ventajas de usar cámaras digitales para la detección de objetos es la capacidad de capturar imágenes de alta resolución. Esto permite la detección de objetos pequeños y de grano fino, como grietas en el pavimento o baches en las carreteras [62]. Además, las cámaras

digitales se pueden sincronizar fácilmente con otros sensores, como lidar y radar, para proporcionar un conjunto de datos multimodal para entrenamiento y evaluación [63].



Figura 11. Cámara Digital Fuente: <https://www.nikon.com.mx/>

5.1.18.2. Cámara Insta360 One X2

La Insta360 One X2, una cámara de bolsillo avanzada, permite la grabación en 360 grados mediante su doble lente, alcanzando una resolución máxima de 5,7K a 30 fotogramas por segundo. Incluye un giroscopio de seis ejes para una estabilización óptima de vídeo y una pantalla frontal para el monitoreo en tiempo real de las grabaciones. Presenta modos de grabación diversificados como 360, Steady Cam, InstaPano y MultiView, mejorando notablemente sus capacidades de estabilización y ofreciendo funciones avanzadas como seguimiento de sujetos y recuadre automático [64].



Figura 12. Cámara de Acción insta360 One X2, Fuente: <https://insta360.com>

5.1.18.3. Insta360 Studio

Insta360 Studio 2023 es un software de escritorio que permite a los usuarios editar videos y fotos tomadas con la ONE RS/R, Ace/Ace Pro, X3/ONE X2/ONEX, GO 3/2, Sphere, EVO, GO, ONE, NanoS, Nano y Air. Permite editar, reencuadrar y renderizar archivos de imágenes y video 360° crudos [65].

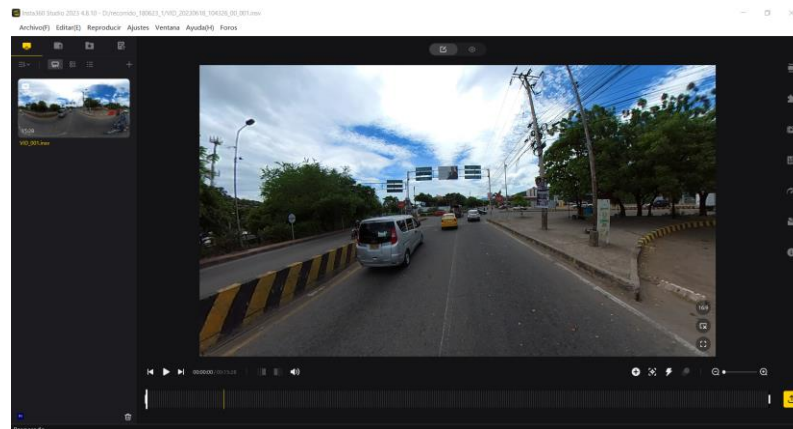


Figura 13. Interfaz de software Insta360 Studio, Fuente: Elaboración Propia.

5.1.18.4. Video Equirectangular 360°

Los videos equirectangulares 360° ofrecen una experiencia panorámica integral, permitiendo al espectador una visión omnidireccional. Se distinguen de los videos tradicionales al no requerir gafas de realidad virtual para su visualización. Su aplicación es amplia, incluyendo ferias comerciales y presentaciones empresariales, aunque su producción demanda una experiencia técnica y de producción considerable [66].



Figura 14. Video en Formato 360, Fuente: Elaboración Propia.

5.1.18.5. Geoetiquetado de Imágenes

El geoetiquetado consiste en agregar información geográfica a los metadatos de archivos de imágenes, vídeos, sonido y sitios web. Generalmente incluye coordenadas de longitud y latitud, y puede ampliarse a detalles como altitud y dirección. El geoetiquetado permite una georreferenciación y localización eficaz de archivos multimedia. Para su realización, se emplean diversas técnicas y herramientas, incluyendo cámaras con GPS integrado, software específico y posicionamiento manual en plataformas web y redes sociales [67].

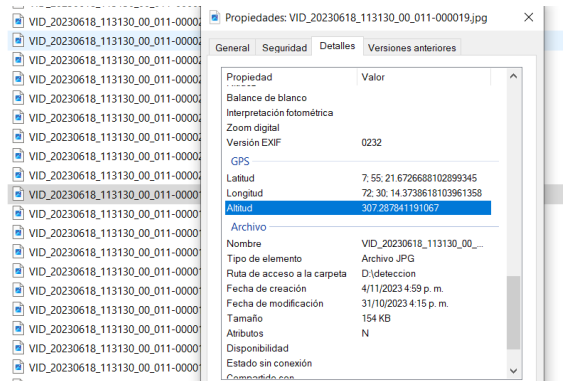


Figura 15. Metadatos de Imagen, Fuente: Elaboración Propia.

5.1.19. Conjunto de Datos de Daños en Carreteras, RDD2022

El Conjunto de Datos de Daños en Carreteras, RDD2022, comprende 47,420 imágenes de carreteras de seis países: Japón, India, la República Checa, Noruega, Estados Unidos y China. Las imágenes han sido anotadas con más de 55,000 instancias de daños en carreteras. Cuatro tipos de daños en carreteras, a saber, grietas longitudinales, grietas transversales, grietas de cocodrilo y baches, están incluidos en el conjunto de datos. El conjunto de datos anotado se ha concebido para desarrollar métodos basados en el aprendizaje profundo que permitan detectar y clasificar automáticamente los daños en carreteras [68].

5.1.20. Georreferenciación

5.1.20.1. Sistemas de información geográfica

Los sistemas de información geográfica (SIG) son sistemas informáticos para capturar, almacenar, analizar y gestionar datos espaciales. Se utilizan para representar y manipular información espacial, como mapas, imágenes de satélite y coordenadas geográficas, y para analizar

y visualizar patrones y relaciones espaciales [69]. Los SIG se usa ampliamente en una variedad de campos, incluidas las ciencias ambientales, la planificación urbana y la salud pública [70].

5.1.20.2. Sistemas de posicionamiento global

El Sistema de posicionamiento global (GPS) es un sistema de navegación basado en satélites que proporciona información sobre la ubicación y la hora en todas las condiciones meteorológicas, en cualquier lugar de la Tierra o cerca de ella [71]. Consiste en una red de satélites que orbitan alrededor de la Tierra, estaciones de control en tierra y unidades receptoras que pueden determinar la ubicación, la velocidad y la hora precisas del receptor. El GPS se usa ampliamente en aplicaciones tales como navegación, mapeo y topografía [72].

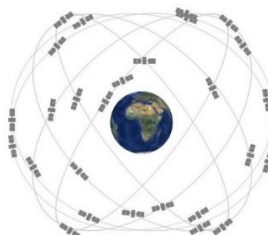


Figura 16. Constelación de satélites GPS, Fuente: <https://www.gps.gov/spanish.php>

5.1.20.3 Proyección Cartográfica CMT12 Origen Nacional para Colombia

Establecida por la Resolución 471 de 2020 del Instituto Geográfico Agustín Codazzi, esta proyección unifica el origen cartográfico de Colombia. Su implementación busca asegurar una representación cartográfica homogénea y continua del territorio colombiano y simplificar la gestión de coordenadas. El sistema tiene como objetivo reducir ambigüedades y errores en el manejo de la información cartográfica, tanto a nivel nacional como internacional [73].

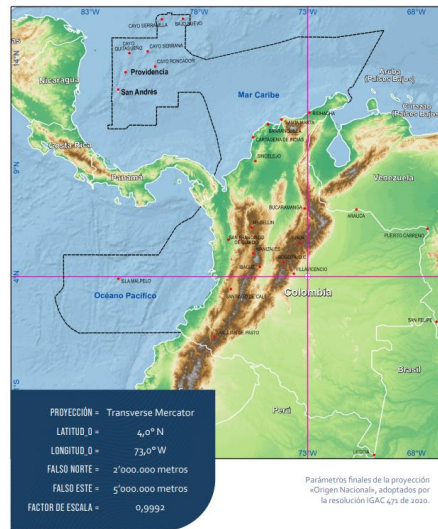


Figura 17. Proyección Cartográfica CMT12, Fuente: <https://origen.igac.gov.co/>

5.1.20.4. Rumbo y Acimut Topográfico

En el ámbito de la topografía, el acimut representa el ángulo horizontal medido desde un plano de referencia, típicamente el norte geográfico, en dirección horaria, y varía de 0° a 360° . Existen tres categorías de acimuts: verdadero, magnético y supuesto. El rumbo, por otro lado, es el ángulo horizontal agudo medido desde el norte o sur hasta un punto de observación, con valores de 0° a 90° . Se expresa mediante combinaciones de las direcciones cardinales (N, S, E, W) [74].

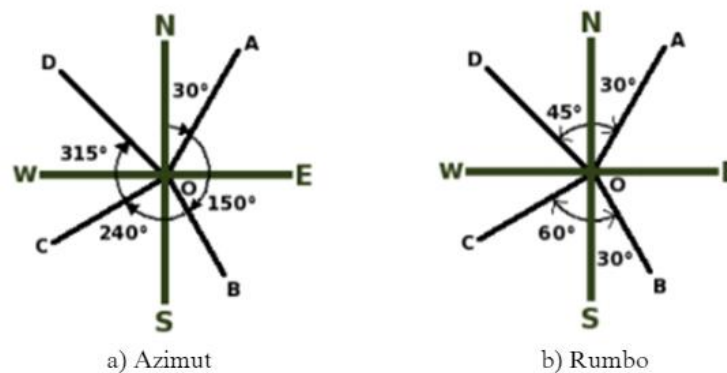


Figura 18. a) Azimut y b) Rumbo, Fuente: Elaboración Propia.

5.1.20.5. Conversión entre Coordenadas Cartesianas y Polares

Las coordenadas polares r y θ pueden convertirse en coordenadas cartesianas (x, y) Las fórmulas de conversión son:

$$x = r \cos(\theta)$$

Ecuación 5. Cálculo de coordenada x.

$$y = r \sin(\theta)$$

Ecuación 6. Cálculo de coordenada y.

Donde r es el radio y θ es el ángulo de acimut [75].

5.1.20.6. Formula Para calcular Equivalencia de Pixeles en Unidades Lineales.

Para conocer esta equivalencia, utilizamos la longitud total en píxeles de un objeto en la imagen y su longitud correspondiente en la realidad (en metros). Con estos dos datos, se calcula la distancia en metros que representa un píxel utilizando una simple regla de tres.

La fórmula para calcular la distancia que representa un píxel en metros es:

$$\text{Distancia por pixel} = \frac{\text{Longitud real en metros}}{\text{Longitud en pixeles}}$$

Ecuación 7. Calculo de distancia por pixel.

5.1.20.7. Modelos de Representación Vectorial

Los modelos vectoriales en Sistemas de Información Geográfica (GIS) representan la geografía mediante puntos, líneas y polígonos. Estos modelos son útiles para almacenar datos con límites discretos, como fronteras de países, parcelas de tierra y calles. Los puntos en un modelo vectorial se utilizan para modelar características discretas como edificios, pozos y postes de energía [76] [77].

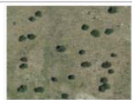





Primitiva	Entidad espacial	Representación	Atributos																		
Puntos			<table border="1"> <thead> <tr> <th>ID</th> <th>Area</th> <th>Distancia Normal</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>17.5</td> <td>35</td> </tr> <tr> <td>2</td> <td>22</td> <td>45.8</td> </tr> <tr> <td>3</td> <td>15</td> <td>27.2</td> </tr> <tr> <td>4</td> <td>18.7</td> <td>38.1</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </tbody> </table>	ID	Area	Distancia Normal	1	17.5	35	2	22	45.8	3	15	27.2	4	18.7	38.1	...		
ID	Area	Distancia Normal																			
1	17.5	35																			
2	22	45.8																			
3	15	27.2																			
4	18.7	38.1																			
...																					
Lineas			<table border="1"> <thead> <tr> <th>Ancho malla</th> <th>Cable malla</th> <th>Longitud</th> </tr> </thead> <tbody> <tr> <td>15</td> <td>4.3</td> <td>35</td> </tr> <tr> <td>6.3</td> <td>3.9</td> <td>5.2</td> </tr> </tbody> </table>	Ancho malla	Cable malla	Longitud	15	4.3	35	6.3	3.9	5.2									
Ancho malla	Cable malla	Longitud																			
15	4.3	35																			
6.3	3.9	5.2																			
Poligonos			<table border="1"> <thead> <tr> <th>Superficie</th> <th>Profundidad máxima</th> </tr> </thead> <tbody> <tr> <td>31434</td> <td>9.07</td> </tr> </tbody> </table>	Superficie	Profundidad máxima	31434	9.07														
Superficie	Profundidad máxima																				
31434	9.07																				

Figura 19. Modelo Vectorial, Fuente: https://volaya.github.io/libro-sig/chapters/Tipos_datos.html

5.1.20.8. Modelos de Representación Ráster

Los modelos ráster en GIS están compuestos por píxeles o celdas de cuadrícula. Son adecuados para representar datos que cambian continuamente a lo largo de un paisaje, como superficies. Un ráster consiste en una matriz de celdas donde cada una contiene un valor que representa información, como la temperatura. La resolución espacial del modelo ráster se determina por el área cubierta por cada píxel [78] [79].

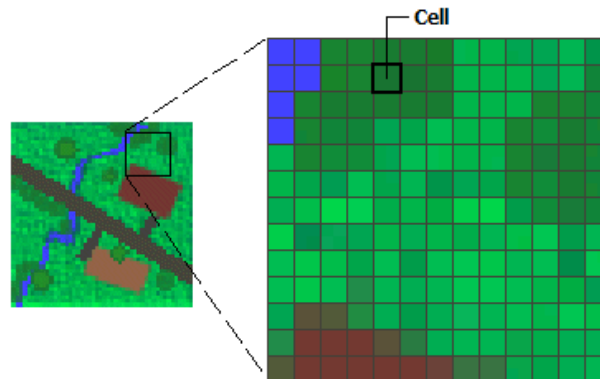


Figura 20. Modelo Ráster, Fuente: <https://desktop.arcgis.com/es/arcmap/latest/manage-data/raster-and-images/what-is-raster-data.htm>

5.1.20.7. ArcGIS

ArcGIS es un software desarrollado por Esri, utilizado para visualizar, editar, gestionar y analizar datos geográficos mediante la creación y visualización de mapas. Es una aplicación de un sistema de información geográfica (GIS) y se puede usar en plataformas de escritorio, móviles y web [80] [81] [82].

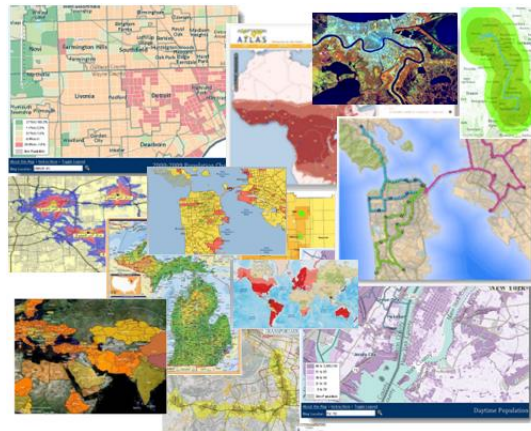


Figura 21. Software ArcGIS, Fuente: <https://resources.arcgis.com/es/help/getting-started/articles/026n00000014000000.htm>

5.1.20.8. ArcGIS Pro

ArcGIS Pro es una aplicación GIS de escritorio desarrollada por Esri que reemplaza a su software ArcMap. Soporta visualización en 2D y 3D, se integra con ArcGIS Online y utiliza Python 3. ArcGIS Pro permite crear, gestionar y compartir mapas a partir de fuentes de datos en 2D, 3D y 4D [83] [84] [85].

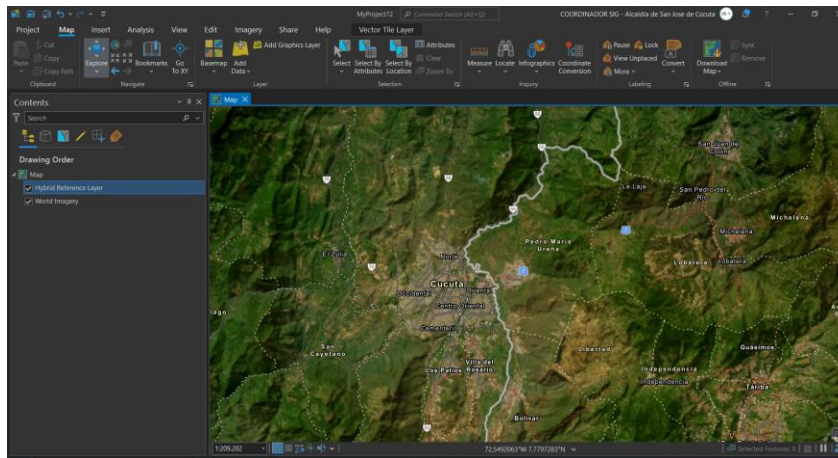


Figura 22. Software ArcGIS Pro, Fuente: Elaboración Propia.

5.1.20.9. Herramienta GeoTagged Photos To Points

Esta herramienta en ArcGIS Pro crea puntos a partir de las coordenadas x, y, z almacenadas en los metadatos de archivos de fotos geolocalizadas. Las fotos pueden agregarse como adjuntos a las características de salida en la geodatabase [86] [87] [88].

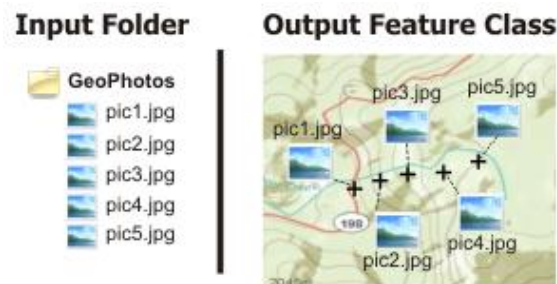


Figura 23. Herramienta GeoTagged Photos To Points, Fuente: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/geotagged-photos-to-points.htm>

5.1.20.10. Unión Espacial en ArcGIS

La herramienta de Unión Espacial en ArcGIS Pro permite unir atributos de una entidad a otra basándose en su relación espacial. Por defecto, todos los atributos de las entidades de unión se anexan a los atributos de las entidades objetivo y se copian a la clase de entidad de salida [89] [90] [91].



Figura 24. Herramienta Spatial Join, Fuente: <https://gisgeography.com/spatial-join/>

5.1.20.11. ArcGIS Online:

ArcGIS Online permite gestionar y compartir datos propios, publicándolos como capas web en la nube de Esri. Ofrece herramientas y aplicaciones para la eficiencia en operaciones de campo, análisis avanzado y mapeo, y permite comprender el mundo a través de mapas interactivos que conectan personas, lugares y datos [92] [93] [94] [95].

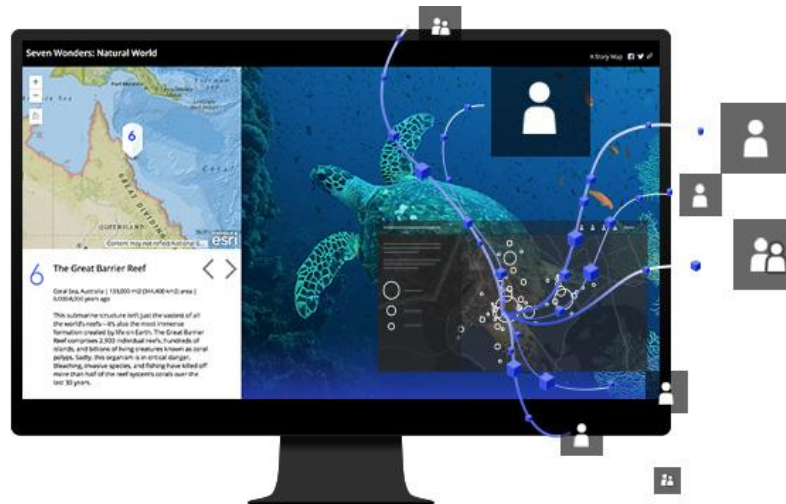


Figura 25. Software ArcGIS Online, Fuente: <https://www.esri.com/en-us/arcgis/products/arcgis-online/overview>

5.1.20.12. ArcGIS Dashboard

ArcGIS Dashboards permite a los usuarios transmitir información mediante visualizaciones de datos interactivas y analíticas basadas en ubicaciones en una sola pantalla. Facilita la toma de decisiones, la visualización de tendencias, el monitoreo en tiempo real y la informar a las comunidades. Ofrece una colección de elementos visuales para presentar los datos de manera significativa [96] [97] [98] [99].



Figura 26. Software ArcGIS Dashboard, Fuente: <https://www.esri.com/en-us/arcgis/products/arcgis-dashboards/overview>

5.2. Antecedentes

Detección y cuantificación de daños en la superficie de la carretera mediante técnicas de procesamiento de imágenes digitales de Mohan N, Bhavani N, Kasthuri K, R A Varun y Bhavya M.

Presenta un método para detectar y cuantificar daños en la superficie de la carretera mediante técnicas de procesamiento de imágenes digitales. El método propuesto consiste en adquirir imágenes de superficies de carreteras con una cámara digital, preprocesar las imágenes para eliminar el ruido y mejorar el contraste, y aplicar técnicas de segmentación de imágenes y extracción de características para identificar y clasificar diferentes tipos de daños. Los autores evalúan el rendimiento del método utilizando un conjunto de datos de imágenes de carreteras y demuestran su capacidad para detectar y cuantificar con precisión los daños en la superficie de la carretera. Este trabajo difiere en el hecho, que la forma de adquirir los datos fue a través de grabación en video a partir de un dron sin la implementación de un sistema de georreferenciación para mapear la severidad de daños [100].

Sistema de adquisición de datos y procesamiento de imágenes para la inspección de superficies de C. Banica, S. V. Paturca, S. D. Grigorescu y A. M. Stefan.

Este artículo presenta el hardware y el software estructuras de un sistema propuesto de adquisición y procesamiento de datos, que está designado para realizar una inspección automatizada del estado del pavimento de las carreteras. El sistema fue diseñado para recopilar información de múltiples fuentes: imágenes basadas en láser sistemas de perfilado de carreteras, cámaras para imágenes ambientales, sensores de proximidad, toda la información está espacialmente correlacionada por medios de odometría y geolocalización. Como trabajo adicional, pretenden aumentar el número de líneas transversales adquiridas por segundo. Aunque el uso de

costosas cámaras de alta velocidad sería una eficiente solución, tienen la intención de mantener un precio asequible del sistema de adquisición, mediante el uso de las mismas cámaras de baja velocidad, pero con un mayor número de proyectores de líneas láser. Los autores se centran en la generación de perfiles longitudinales para el mapeo detallado de la superficie de las vías, este trabajo difiere en el hecho de que no realizaron una clasificación, georreferenciación y cuantificación de los daños presentes en la superficie de la vía y no crearon un sistema de georreferenciación para mapear la severidad de daños [101].

Detección y clasificación de daños en la carretera basada en el aprendizaje profundo para múltiples países de Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Alexander Mraz, Takehiro Kashiyama, Yoshihide Sekimoto.

En este trabajo se toma como precursor a Japón, que han desarrollado métodos menos costosos y fáciles de realizar basados en teléfonos inteligentes para el monitoreo automático del estado de las vías, se manifiesta que otros países aún luchan por encontrar soluciones eficientes. En esta investigación, se evalúa la usabilidad del modelo japonés en otros países, se propone el uso de un conjunto de datos de daños viales heterogéneos a gran escala que comprende 26.620 imágenes recopiladas de varios países (India, Japón y la República Checa) utilizando teléfonos inteligentes. Se plantean modelos capaces de detectar y clasificar los daños viales en más de un país. Este estudio brinda recomendaciones para lectores, agencias locales y municipios naciones, cuando otro país publica sus datos y el modelo para la detección y clasificación automática de daños en las carreteras, se hace mención que los datos de entrada son proporcionados por varios países, los tipos de daños a identificar son ocho. En este trabajo no se realiza la cuantificación de daños y no se establece un sistema de georreferenciación para mapear la severidad de daños [102].

Deep Learning para la detección de fallas en pavimentos de una zona del distrito de Villa María del Triunfo 2022 de Zúñiga Guisado, Yenmy.

presenta un sistema de inteligencia artificial basado en aprendizaje profundo para la detección de fallas en pavimentos en una zona específica del distrito de Villa María del Triunfo. El objetivo principal es identificar grietas y huecos en pavimentos asfaltados o de cemento mediante arquitecturas de redes neuronales convolucionales, utilizando el algoritmo YOLOv5 y el lenguaje de programación Python con diversas bibliotecas. La investigación, de tipo aplicado y enfoque cuantitativo, se llevó a cabo en la zona de Jose Carlos Mariátegui, con una población de 420 imágenes capturadas de estructuras viales a través de un teléfono inteligente. Se aplicó un muestreo probabilístico por conveniencia y se analizaron 201 imágenes, con un test de 30 imágenes para evaluar la precisión y sensibilidad del sistema. Los resultados mostraron niveles de precisión y sensibilidad variables para grietas y huecos, con un promedio de confianza del 0.62% en 724 iteraciones. En resumen, la investigación busca desarrollar un sistema inteligente para la inspección de pavimentos con enfoque en la detección de fallas [103].

Detector de baches con deep learning de Manzanares González, Albert.

El mantenimiento deficiente de las carreteras, que puede representar peligros para los usuarios y daños a los vehículos, se aborda en este proyecto a través de la automatización de la detección de baches utilizando tecnología Deep Learning. La actual forma manual de reportar incidencias se considera costosa, y se propone reemplazarla con un sistema automatizado que reduzca costos y mejore la eficiencia. El detector de baches no solo busca agilizar el mantenimiento de las carreteras, sino también proporcionar una herramienta para que los usuarios mejoren su seguridad al recibir alertas sobre baches o ajustar la suspensión electrónica de sus vehículos para absorberlos mejor. El proyecto explora el estudio de Deep Learning, una tecnología en expansión y mejora continua, para la detección de objetos. Se detalla la selección de plataformas, el

entrenamiento de la red neuronal, la comparación de modelos y la implementación en una aplicación móvil como prototipo funcional del detector de baches. El contenido del proyecto está diseñado para ser comprensible y accesible para aquellos interesados en Deep Learning y redes neuronales, sin necesidad de experiencia previa en el tema [104].

6. MATERIALES Y METODOS

6.1. Construir una base de datos de imágenes de pavimentos flexibles con diferentes tipos de averías.

6.1.1. Definición del Área de Levantamiento de Datos

La primera etapa para la construcción de la base de datos, fue la definición del área de levantamiento de datos. Se seleccionó el área urbana de San José de Cúcuta en Norte de Santander, por cuestiones de limitaciones en espacio de almacenamiento y seguridad personal, se realizó el recorrido en las vías del subsistema vial arterial y la avenida principal del municipio de los patios. La elección de esta área se basó en la predominancia del pavimento flexible (asfalto), lo cual es representativo de muchas zonas urbanas y es crucial para el estudio de averías típicas en este tipo de material. Esta área ofrece una diversidad de condiciones de tráfico y exposición ambiental, proporcionando una gama representativa de daños comunes en pavimentos flexibles, como grietas, piel de cocodrilo y bache [17]. Además, la elección de una zona urbana densamente poblada y activa asegura que las imágenes capturadas reflejen una variedad de desafíos reales encontrados en el mantenimiento de carreteras.



Figura 27. Vías arteriales de San José de Cúcuta.

6.1.2. Planificación de las Rutas del Recorrido para el Levantamiento de Datos

La planificación de las rutas fue un paso crítico para garantizar una recolección de datos eficiente y exhaustiva. Se diseñaron rutas que cubrían integralmente las vías seleccionadas, minimizando la distancia total recorrida mientras se maximizaba la cobertura de diferentes tipos de pavimentos y condiciones de tráfico. Esta planificación meticulosa incluyó la identificación de tramos de carreteras con alta incidencia de averías y aquellos con mantenimiento reciente para obtener un equilibrio en la variedad de condiciones del pavimento. La estrategia de ruta también consideró factores como el tráfico esperado, las horas pico, y las condiciones de iluminación para optimizar las sesiones de captura de imágenes y garantizar la calidad y relevancia de los datos recogidos.

6.1.3. Captura de Imágenes Geoetiquetadas.

La captura de imágenes se realizó utilizando la cámara Insta360 modelo One X2 [64]. Acoplada en una motocicleta utilizando un bastón y a una altura de 2,15 metros en relación al pavimento. Se configuro la cámara para registrar posiciones cada segundo permitiendo un geoetiquetado preciso y detallado. Se mantuvo una velocidad constante de 30 km/h durante la captura de imágenes, asegurando que las fotos fueran claras y consistentes en calidad, mientras se cubría las rutas seleccionadas.

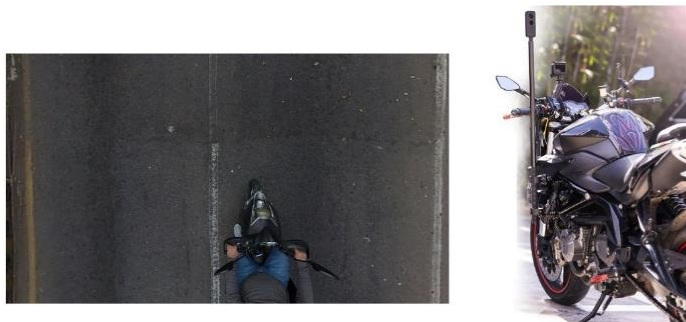


Figura 28. Sistema de Captura de Imágenes, con cámara insta 360 X2.

6.1.4. Almacenamiento y preprocesamiento de las Imágenes Capturadas.

Del recorrido realizado a través de las rutas, obtuvimos 27 000 fotografías de pavimentos flexibles. Este conjunto de datos le llamaremos RECOCUCUTA23, las imágenes fueron almacenadas digitalmente para su análisis y procesamiento posterior. Se empleó un sistema de almacenamiento estructurado, donde cada imagen se guardó con su correspondiente metadato, incluyendo la ubicación exacta, fecha y hora de captura. Este nivel de organización es crucial para facilitar el acceso rápido a imágenes específicas durante las etapas de análisis y para mantener un registro detallado del conjunto de datos. La elección del formato .jpg para las imágenes ofreció un equilibrio ideal entre calidad de imagen y eficiencia de almacenamiento.

El preprocesamiento de las imágenes incluyó la corrección de color, ajustes de luminosidad y corrección de perspectiva, con el fin de garantizar que los detalles del pavimento fueran claramente visibles y consistentes en todas las imágenes. Este paso fue esencial para mejorar la calidad de los datos y preparar las imágenes para el procesamiento posterior y el entrenamiento del modelo de aprendizaje profundo, este procedimiento lo realizamos utilizando el software de edición de datos Insta360 Studio [65]. Además, se realizó una revisión manual para descartar imágenes borrosas, con obstrucciones o que no aportaran información relevante sobre el estado del pavimento. Este proceso de filtrado aseguró que solo las imágenes más informativas y de alta calidad fueran incluidas en la base de datos.



Figura 29. Imagen sin Corrección de Color



Figura 30. Imagen con Corrección de Color.



Figura 31. Imagen sin corrección de Perspectiva

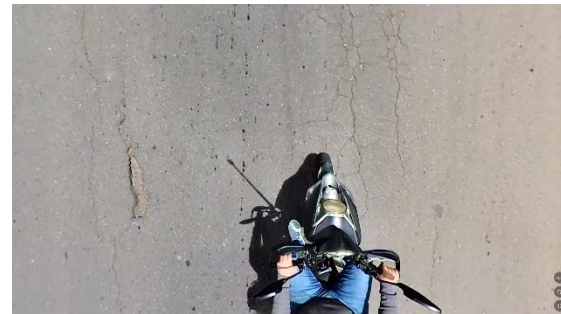


Figura 32. Imagen con corrección de Perspectiva.

6.1.5. Etiquetado de los Daños a Detectar

En esta etapa utilizamos el software labelImg para realizar el etiquetado en formato yolo, identificamos cuatro (4) tipos principales de daños en pavimentos: Fisura Lateral, Fisuras Longitudinal, Piel de Cocodrilo y Baches, por parte del Ingeniero Catastral y Geodesta Manuel Alejandro Morales Acevedo. Morales Acevedo, con su profundo conocimiento adquirido a través de su notable trayectoria en inventarios de vías e identificación de daños en pavimentos, especialmente durante su tiempo en la Secretaria de Infraestructura en la Alcaldía de Cúcuta, Norte de Santander, juega un papel crucial la realización de este proyecto. Su experiencia no solo se limita a la práctica, sino que también se extiende a la teoría y las mejores prácticas, como lo demuestra su conocimiento profundo del “Manual para la inspección visual de pavimentos flexibles” de INVÍAS [17]. Durante el etiquetado cada categoría fue cuidadosamente definida y etiquetada con un identificador

de clase único, asegurando la coherencia y precisión en el conjunto de datos de entrenamiento.



Figura 33. Interfaz de software labellmg.

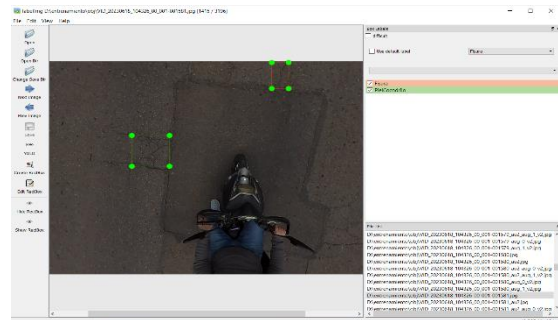


Figura 34. Etiquetado de daños con el software labellmg

6.1.6. Bases de Datos Existentes utilizadas en la literatura.

A partir del Conjunto de Datos de Daños en Carreteras, RDD2022 [68]. Utilizamos los dataset de imágenes correspondientes a china_drone y china_motorbike, en la tabla 1, se relacionan la cantidad de imágenes disponibles en estos dos datasets. Estas imágenes ya cuentan con etiquetado de daños en formato pascal [51], estas etiquetas fueron convertidas a formato yolo [47].

Dataset	Imágenes	Etiquetas
China_Drone	2401	2401
China_MotorBike	1977	1977

Tabla 1. Relación de imágenes del dataset RDD2022.

6.2 Entrenamiento del modelo de aprendizaje profundo.

6.2.1. Definición del Modelo de Aprendizaje Profundo más Adecuado.

En el núcleo de este proyecto se encuentra el algoritmo YOLO (You Only Look Once) [33]. Elegido por su reconocida eficiencia en la detección de objetos en tiempo real. Esta selección fue estratégica, considerando las necesidades específicas del proyecto que demandaba rapidez y precisión en la identificación de daños en pavimentos flexibles. YOLO, conocido por su capacidad para procesar imágenes con rapidez y ofrecer resultados precisos, se alineaba perfectamente con los objetivos. La habilidad de YOLO para analizar una imagen en su conjunto en un solo paso, en lugar de partes individuales, lo convierte en un algoritmo excepcionalmente rápido y efectivo para la detección de objetos [34].

Este algoritmo es especialmente adecuado para nuestro proyecto debido a su capacidad de generalizar a partir de un conjunto de datos diverso y su robustez en diferentes escenarios. La eficiencia de YOLO en la detección en tiempo real es crucial para la aplicación en el análisis de daños en pavimentos, donde la rapidez y precisión son esenciales. Además, la flexibilidad de YOLO para adaptarse a variados tamaños de imágenes y condiciones de iluminación lo hace ideal para nuestro conjunto de datos, que comprende imágenes tomadas bajo diferentes condiciones ambientales y de distintas fuentes.

6.2.2. Ejecutar el Modelo Seleccionado con los Datos de Entrenamiento.

Para el entrenamiento de los modelos, optamos por emplear tanto YOLO v4 [38]. Como YOLO v8 [39]. Para comparar y evaluar sus rendimientos en nuestro conjunto de datos específico. YOLO v4 fue nuestra elección inicial, dada su robustez y eficacia probada en múltiples aplicaciones [33][34]. Las imágenes para el entrenamiento provinieron de dos fuentes principales, el dataset DSCUCU1 y el dataset RD2022 [68]. Este enfoque mixto nos permitió utilizar una amplia variedad de tipos de daños, lo cual es vital para entrenar un modelo robusto y confiable. Las imágenes fueron divididas en un conjunto de

entrenamiento y prueba para YOLO v4, y otro conjunto de entrenamiento, validación y prueba para YOLO v8.

6.3 Evaluación del modelo.

6.3.1. Evaluación del Modelo para Entorno de Producción.

La evaluación de los modelos entrenados fue un paso crítico para determinar su idoneidad para su uso en entornos de producción. Utilizamos un conjunto de métricas estándar, incluyendo precisión, recall [43], F1-score [44], y mAP (mean Average Precision) [45]. Para evaluar el rendimiento de los modelos YOLO v4 y YOLO v8. Estas métricas proporcionaron una comprensión detallada del funcionamiento de cada modelo en términos de capacidad para la identificación correcta, los daños en las imágenes y minimizar los falsos positivos.

6.4 Implementación del sistema de georreferenciación.

6.4.1. Definición del Sistema de Georreferenciación a Usar.

La implementación del sistema de georreferenciación comenzó con la elección de una herramienta adecuada para la tarea. Se seleccionó ArcGIS [80]. Combinando ArcGIS Pro [83]. Para el uso de escritorio y ArcGIS Online [92]. Para aplicaciones basadas en la nube. Esta combinación ofrece una plataforma versátil y potente para la manipulación y análisis de datos geográficos. El sistema de coordenadas seleccionado fue MAGNA-SIRGAS, origen nacional [73]. Que es ampliamente utilizado en Colombia, proporcionando un marco de referencia geográfico preciso y confiable. Esta elección fue crucial para garantizar la precisión y la coherencia en la georreferenciación de los datos recopilados.

La implementación de ArcGIS Pro [83]. Permitió un análisis detallado y una visualización avanzada de los datos georreferenciados. Su capacidad para manejar grandes conjuntos de datos y realizar operaciones complejas de análisis espacial lo convirtió en una herramienta ideal para este proyecto. Por otro lado, ArcGIS Online [92]. Facilitó la colaboración y el acceso a los datos georreferenciados a través de una plataforma basada en la nube, permitiendo la visualización y el análisis de los datos desde cualquier lugar y en cualquier momento. La combinación de estas dos plataformas proporcionó una solución integral para la gestión de los datos geográficos relacionados con los daños en pavimentos flexibles.

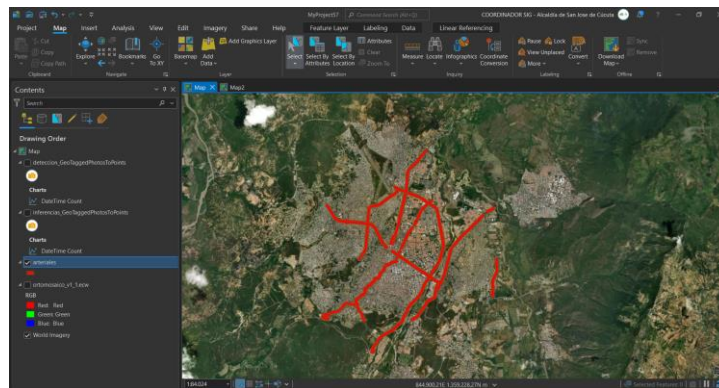


Figura 35. Vías arteriales de san José de Cúcuta desplegadas en ArcGIS Pro.

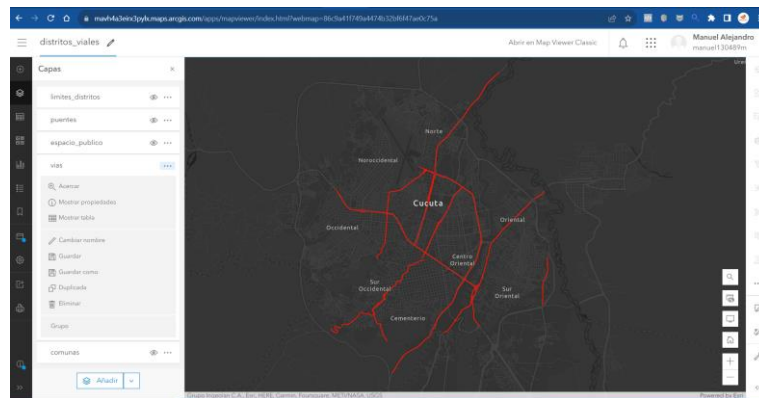


Figura 36. Vías arteriales de san José de Cúcuta desplegadas en ArcGIS Online.

6.4.2. Creación de Datos Geográficos de Segmentos Viales y Calzadas del Área de Estudio

El siguiente paso en la implementación del sistema de georreferenciación fue la creación de datos geográficos detallados de los segmentos viales y las calzadas en el área de estudio. Utilizando la ortofoto de la ciudad de Cúcuta, se procedió a crear dos capas vectoriales en ArcGIS Pro [83]. Una capa de tipo línea para representar los segmentos viales, y otra de tipo polígono para representar las calzadas. Esta diferenciación fue vital para un análisis exhaustivo, permitiendo una representación precisa de la infraestructura vial.

La capa de tipo línea fue diseñada para representar los segmentos viales en el sentido de circulación de la vía, divididos cada 100 metros, facilitando un análisis granular de las condiciones viales. Cada segmento se enriqueció con atributos como el nombre y el código de la vía, lo que facilitó la identificación y el análisis posterior. Por otro lado, la capa de tipo polígono, representando las calzadas, incluyó atributos de geometría como el área y el perímetro, proporcionando una comprensión detallada de las dimensiones físicas de las vías. Estas capas fueron fundamentales para analizar las condiciones de las vías arteriales de la ciudad de San José de Cúcuta y para vincular posteriormente los daños detectados en el pavimento.

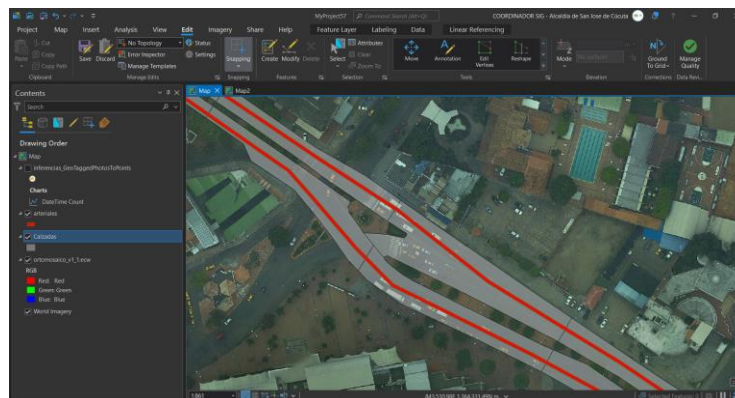


Figura 37. Líneas de Ejes de vías y polígonos de calzadas.

6.4.3. Georreferenciación de las Imágenes Geoetiquetadas

Las imágenes capturadas con la cámara Insta 360 X2 [64]. Se exportaron cada 4 metros e incluyeron metadatos valiosos como coordenadas [75]. Y orientación (azimut) [74]. Que fueron esenciales para la georreferenciación precisa. El proceso de georreferenciación se realizó mediante ArcGIS Pro [83]. Utilizando la herramienta GeoTagged Photos To Points [86]. Esta herramienta permitió leer los metadatos de las imágenes y georreferenciar las ubicaciones desde donde se tomaron, lo cual fue un paso crucial para vincular los datos visuales con su ubicación geográfica exacta.

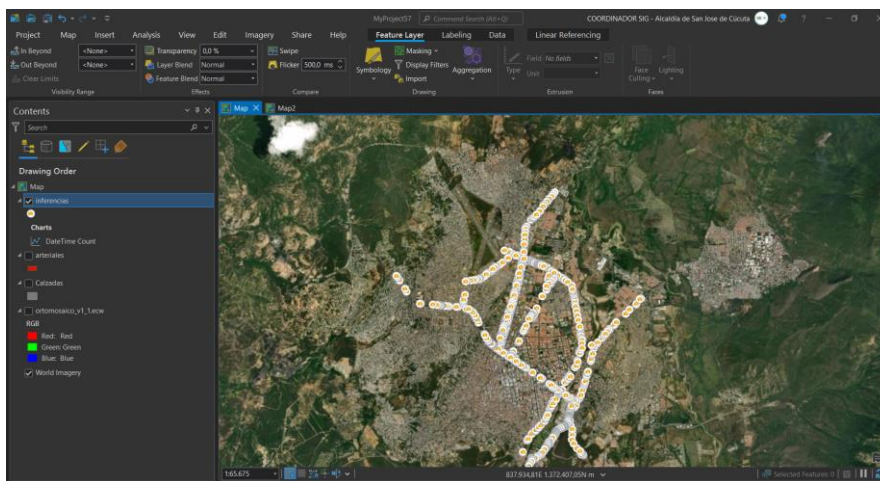


Figura 38. Georreferenciación de las imágenes con daños.

Una tarea importante fue comparar la equivalencia de la longitud de un píxel en las imágenes tomadas con la cámara Insta360 One X2 [64]. Con la longitud verdadera en el terreno. Para ello, se utilizó la ortofoto de Cúcuta como referencia, permitiendo establecer una relación precisa entre las imágenes y la realidad física. Esta equivalencia fue fundamental para garantizar que la interpretación y el análisis de los datos visuales fueran precisos y confiables, proporcionando una base sólida para la georreferenciación subsiguiente de los daños identificados en las imágenes.

En nuestra imagen encontramos un objeto que mide 1011 píxeles de longitud, y sabemos que en la realidad este objeto mide 3,04 metros, entonces la distancia que representa un píxel sería:

Aplicando la Ecuación 7. Tenemos:

$$\text{Distancia por píxel} = \frac{3,04 \text{ metros}}{1011 \text{ píxeles}} = 0,003\text{m/píxel}$$

Esto significa que cada píxel en nuestras imágenes representa 0.003 metros (o 3 milímetros) en la realidad.

Es importante tener en cuenta que este tipo de cálculos es específico para las condiciones bajo las cuales se tomaron las medidas (como la distancia de la cámara al objeto, la resolución de la cámara, etc.). Por lo tanto, esta fórmula no es universal y solo es aplicable bajo las mismas condiciones específicas para las que fue derivada.

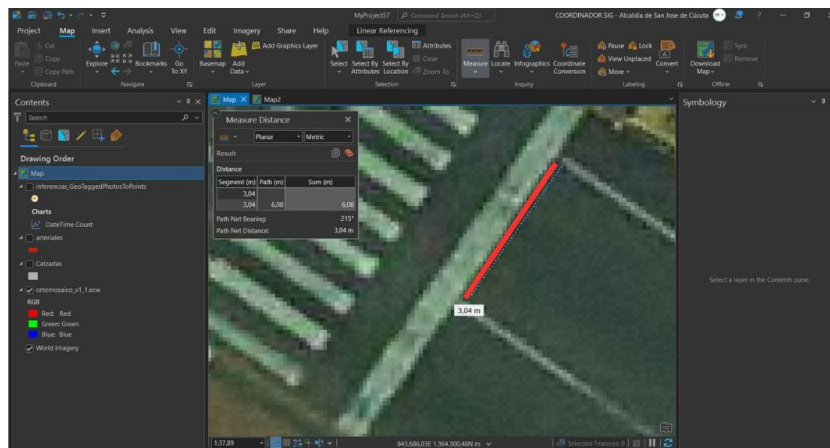


Figura 39. Medición de distancia de objeto sobre el terreno.

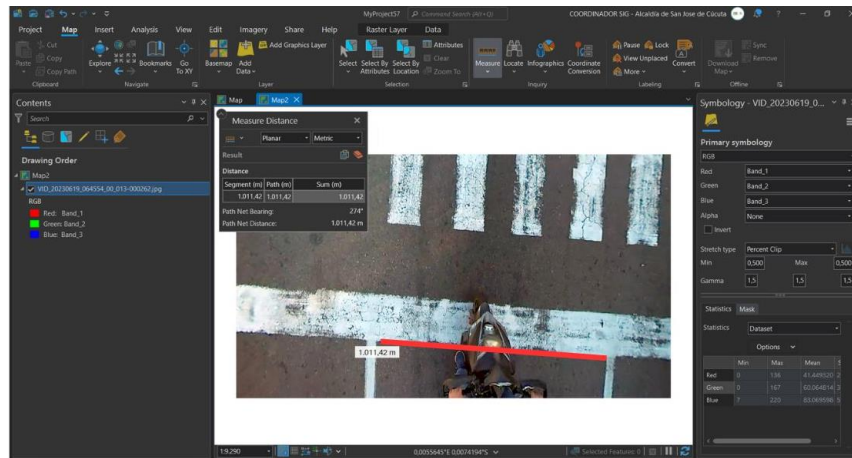


Figura 40. Medición de distancia de objeto en pixeles.

6.4.4. Unión de Atributos de las Calzadas y los Daños Identificados

Una vez cuantificados y georreferenciados los daños, el siguiente paso fue la unión de estos datos con los atributos de las calzadas. Mediante una intersección espacial [89]. Se pudo determinar cuántos y qué tipos de daños había en cada tramo de 100 metros de las vías. Esta información fue crucial para establecer la densidad de daños en las vías, proporcionando una visión clara de las áreas más afectadas y las que requerían atención prioritaria.

Este análisis espacial permitió no solo identificar las áreas con mayor concentración de daños, sino también correlacionar los tipos de daños con características específicas de las vías, como su uso, antigüedad y condiciones de tráfico. Esta correlación proporcionó insights valiosos para la toma de decisiones en cuanto a la planificación urbana y el mantenimiento vial, permitiendo una gestión más eficiente y efectiva de los recursos para la reparación y el mantenimiento de las vías.

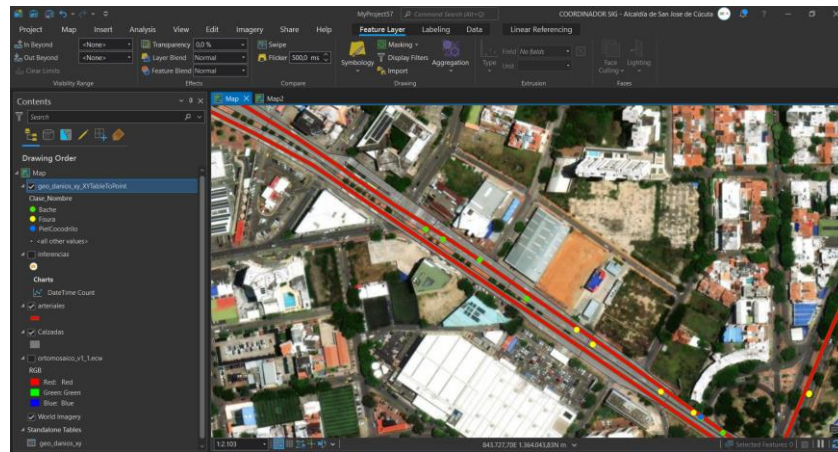


Figura 41. Unión de Atributos de daños detectados y código de calzada.

7. RESULTADOS

7.1. Construir una base de datos de imágenes de pavimentos flexibles con diferentes tipos de averías.

Una vez finalizado el etiquetado obtuvimos un total de 927 imágenes, en las cuales se identificaron 1 815 daños. La base de datos resultante la llamaremos DSCUCU1. En la tabla 2, se detalla la cantidad de etiquetas por tipo de daño.

Dataset	Tipo de Daño	Etiquetas
DSCUCU1	FisuraLateral	286
DSCUCU1	Fisura	908
DSCUCU1	PielDeCocodrilo	279
DSCUCU1	Bache	342

Tabla 2. Cantidad de etiquetas del dataset DSCUCU1.

El etiquetado meticuloso de estas imágenes fue fundamental para garantizar que el modelo aprenda a identificar correctamente los diferentes tipos de daños en pavimentos flexibles.

7.2. Entrenamiento del modelo de aprendizaje profundo.

A continuación, se detallan una serie de experimentos que se realizaron durante este proyecto, con el fin de comprender el funcionamiento del modelo y la configuración adecuada de sus parámetros. Para cada uno de los experimentos se empleó el backend de Google Colab [31]. Con Python 3 como entorno de ejecución.

Experimento 1

En este primer experimento nos permitió familiarizarnos con el uso del modelo, se empleó el backend de Google Colab [31]. Con Python 3 y GPU v100 de 16gb como entorno de ejecución, seleccionamos el modelo Yolov4 [38]. Para realizar el entrenamiento del modelo, utilizamos el dataset de imágenes etiquetadas DSCUCU1 con un total de 927 imágenes. El número de clases a detectar fueron 4, Fisura Lateral, Fisura Longitudinal, Piel de Cocodrilo y Baches.

Parámetros recomendados [40]. Utilizados en el entrenamiento del modelo del experimento 1.:

- batch=64
- subdivisions=16
- max_batches = 8 000.
- steps=6 400,7 200.
- width=416 y height=416.
- classes=4.
- filters = 27.

Se ejecutó el entrenamiento del modelo, correspondiente al experimento 1, se detuvo el entrenamiento en la iteración 6 400, es importante resaltar que Yolo v4 almacena los pesos de las iteraciones cada 1 000 iteraciones, también genera un archivo de la última iteración y un archivo de los mejores pesos calculados durante el entrenamiento, en la figura 42. se muestra el desempeño del modelo a través del gráfico del entrenamiento.

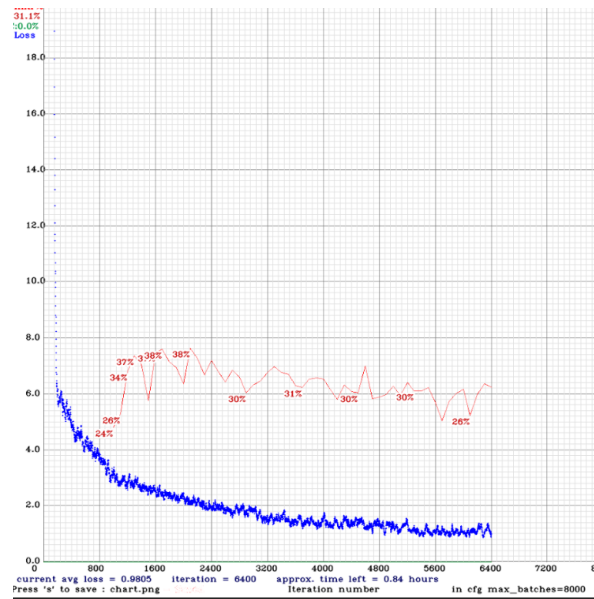


Figura 42. Gráfico de entrenamiento experimento 1.

Para conocer la precisión del modelo del experimento 1, se calculó el map (mean average precisión) [45]. Utilizando el dataset DSCUCU1 y los mejores pesos estimados en el modelo. En la tabla 3, se muestran los resultados.

Class_ID	Name	Ap (%)	Tp	Fp
0	FisuraLateral	31.91	38	46
1	Fisura	37.14	117	120
2	PielDeCocodrilo	43.57	55	50
3	Bache	41.09	39	28

Tabla 1. Precisiones de ejecución de datos de experimento 1. **Class ID:** Identificador de la clase. **Name:** Nombre de la clase. **AP (%):** Precisión media promedio en porcentaje. **TP:** Verdaderos positivos. **FP:** Falsos positivos.

Experimento 2

En este segundo experimento se empleó también el backend de Google Colab [31]. Con Python 3 y GPU v100 de 16gb como entorno de ejecución, seleccionamos nuevamente el modelo Yolov4 [38]. Con el fin de mejorar el desempeño en la detección de fisuras, y balancear el número de clases, realizamos un proceso de aumentación de datos para las imágenes de este dataset [58]. El resultado fue la inclusión de las imágenes aumentadas en el dataset DSCUCU1. Este nuevo dataset lo llamamos DSCUCU2 con un total de 2 113 imágenes. El número de clases a detectar también fueron 4, Fisura Lateral, Fisura Longitudinal, Piel de Cocodrilo y Baches. En la tabla 4 se muestran la cantidad de imágenes y etiquetas disponibles en este nuevo dataset.

Dataset	Tipo de Daño	Etiquetas
DSCUCU2	FisuraLateral	691
DSCUCU2	Fisura	908
DSCUCU2	PielDeCocodrilo	649
DSCUCU2	Bache	892

Tabla 2. Cantidad de etiquetas del dataset DSCUCU2.

Parámetros recomendados [40]. Utilizados en el entrenamiento del modelo del experimento 2:

- batch=64
- subdivisions=16
- max_batches = 8 000.
- steps=6 400,7 200.
- width=416 y height=416.

- classes=4.
- filters = 27.

Se ejecutó el entrenamiento del modelo, correspondiente al experimento 2, se detuvo el entrenamiento en la iteración 6 200, en la figura 43. se muestra el desempeño del modelo a través del grafico del entrenamiento.

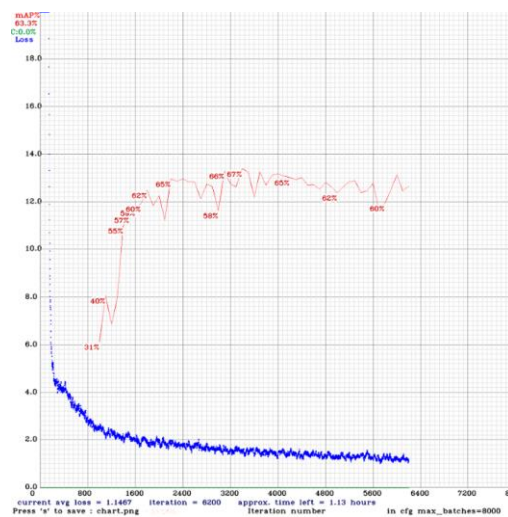


Figura 43. Gráfico de entrenamiento experimento 2.

Para conocer la precisión del modelo del experimento 2, se calculó el map (mean average precisión) [45]. Utilizando el dataset DSCUCU2 y los mejores pesos estimados en el modelo. En la tabla 5, se muestran los resultados.

Class_ID	Name	Ap (%)	Tp	Fp
0	FisuraLateral	59.44	125	58
1	Fisura	43.94	125	90
2	PielDeCocodrilo	73.50	149	59
3	Bache	90.86	270	40

Tabla 3. Precisiones de ejecución de datos de experimento 2. Class ID: Identificador de la clase. Name: Nombre de la clase. AP (%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Experimento 3

En este tercer experimento se empleó nuevamente el backend de Google Colab [31]. Con Python 3 y GPU v100 de 16gb como entorno de ejecución, seleccionamos nuevamente el modelo Yolov4 [38]. En este experimento incluimos un nuevo dataset llamado RDD2022 [68]. Con el fin de seguir mejorando la detección de fisuras, y balancear el número de clases, realizamos un proceso de filtrado de etiquetas y también una nueva aumentación de datos para las imágenes [58]. Del dataset DSCUCU2 y del dataset RDD2022 [68]. Este nuevo dataset lo llamamos DSCUCU3 con un total de 2 718 imágenes, que incluye imágenes del dataset DSCUCU2 y del dataset RDD2022 [68]. Implementamos un cambio en las clases, decidimos no tomar en cuenta la clase fisura lateral con el fin de evitar la confusión del modelo en la detección con respecto a la clase piel de cocodrilo, lo que redujo el número de clases a 3, Fisura, Piel de Cocodrilo y Baches. En la tabla 6 se muestran la cantidad de imágenes y etiquetas disponibles en este nuevo dataset.

Dataset	Tipo de Daño	Etiquetas
DSCUCU3	Fisura	1573
DSCUCU3	PielDeCocodrilo	1022
DSCUCU3	Bache	958

Tabla 4. Cantidad de etiquetas del dataset DSCUCU3.

Parámetros recomendados [40]. Utilizados en el entrenamiento del modelo del experimento 3:

- batch=64
- subdivisions=16
- max_batches = 6 000.
- steps=4 800,5 400
- width=416 y height=416.
- classes=3.
- filters = 24.

Se ejecutó el entrenamiento del modelo, correspondiente al experimento 3, se detuvo el entrenamiento en la iteración 5 900, en la figura 44. Se muestra el desempeño del modelo a través del grafico del entrenamiento.

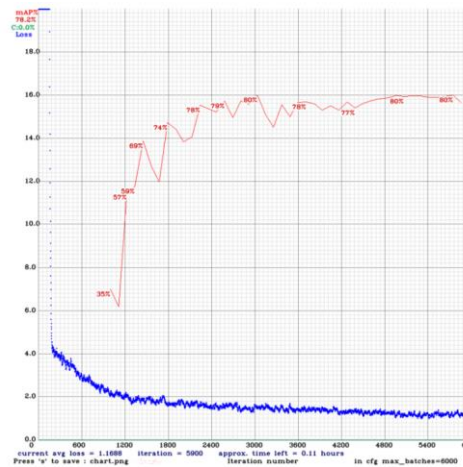


Figura 44. Gráfico de entrenamiento experimento 3.

Para conocer la precisión del modelo del experimento 3, se calculó el map (mean average precisión) [45]. Utilizando el dataset DSCUCU3 y los mejores pesos estimados en el modelo. En la tabla 7, se muestran los resultados.

Class_ID	Name	Ap (%)	Tp	Fp
0	Fisura	70.80	354	147
1	PielDeCocodrilo	79.98	260	66
2	Bache	89.10	292	49

Tabla 5. Precisiones de ejecución de datos de experimento 2. Class ID: Identificador de la clase. Name: Nombre de la clase. AP (%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Experimento 4

En este cuarto experimento se empleó nuevamente el backend de Google Colab [31]. Con Python 3, esta vez con GPU a100 de 40gb como entorno de ejecución. Para este experimento decidimos emplear el modelo Yolov4 [38]. Y también la versión más reciente de yolo, que es la versión 8 o Yolov8 [39]. Con el fin de perfeccionar de detección de los daños y balancear el número de clases, realizamos un nuevo proceso de filtrado de etiquetas y también una nueva aumentación de datos para las imágenes [58]. Del dataset DSCUCU3. Este nuevo dataset lo llamamos DSCUCU4 con un total de 3 196 imágenes. En este último experimento utilizamos 3 clases, Fisura, Piel de Cocodrilo y Baches. En la tabla 8 se muestran la cantidad de imágenes y etiquetas disponibles en este nuevo dataset. Para este entrenamiento se utilizaron tamaños más grandes para las imágenes de entrada, hasta el máximo que el hardware podía soportar en cada modelo.

Dataset	Tipo de Daño	Etiquetas
DSCUCU2	Fisura	2143
DSCUCU2	PielCocodrilo	1922
DSCUCU2	Bache	1361

Tabla 6. Cantidad de etiquetas del dataset DSCUCU4.

Este cuarto experimento se dividió en dos partes, la primera parte es el entrenamiento del dataset DSCUCU4 con el modelo Yolov4.

Parámetros recomendados [40]. Utilizados en el entrenamiento del modelo del experimento 4 con Yolov4:

- batch=64
- subdivisions=16
- max_batches = 6 000.
- steps=4 800, 5 400
- width=608 y height=608
- classes=3.
- filters = 24.

Se ejecutó el entrenamiento del modelo, correspondiente al experimento 4 con Yolov4 [38]. Se detuvo el entrenamiento en la iteración 4 200, en la figura 45. se muestra el desempeño del modelo a través del grafico del entrenamiento.

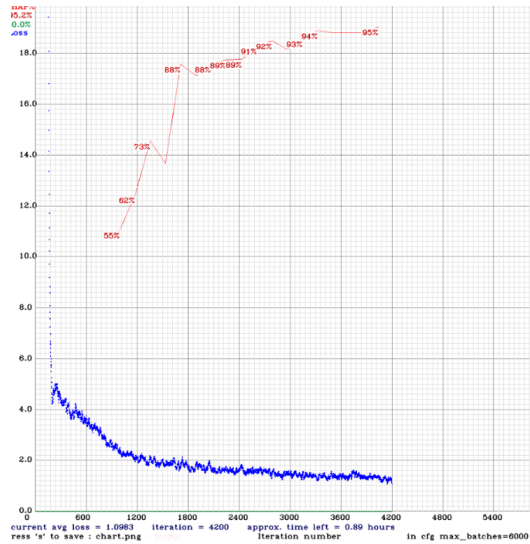


Figura 45. Gráfico de entrenamiento experimento 4 Yolo v4.

Para conocer la precisión del modelo del experimento 4, se calculó el map (mean average precisión) [45]. Utilizando el dataset DSCUCU4 y los mejores pesos estimados en el modelo. En la tabla 9, se muestran los resultados.

Class_ID	Name	Ap (%)	Tp	Fp
0	Fisura	88.17	185	42
1	PielCocodrilo	97.92	190	10
2	Bache	99.25	126	10

Tabla 7. Precisiones de ejecución de datos de experimento 4 Yolo v4.. *Class ID:* Identificador de la clase. *Name:* Nombre de la clase. *AP (%):* Precisión media promedio en porcentaje. *TP:* Verdaderos positivos. *FP:* Falsos positivos.

La segunda parte es el entrenamiento del dataset DSCUCU4 con el modelo Yolov8 [39]. Hay que destacar que en Yolov8 se debe dividir el dataset originales en 3 partes, entrenamiento, prueba y validación.

Parámetros recomendados [41][42]. Utilizados en el entrenamiento del modelo del experimento 4 con Yolov8:

- workers=2
- batch=12
- device=0
- epochs=120
- patience=50
- imgsz=1 216

Se ejecuto el entrenamiento del modelo, correspondiente al experimento 4 con Yolov8 [39]. Se detuvo el entrenamiento en época 60, en la figura 46. se muestra el desempeño del modelo a través del grafico del entrenamiento.

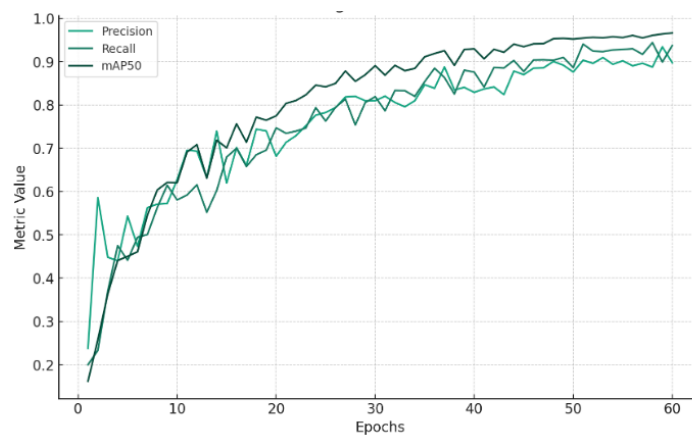


Figura 46. Gráfico de entrenamiento experimento 4. Yolov8.

Para conocer la precisión del modelo del experimento 4 con Yolov8, se calculó el map (mean average precisión) [45]. Utilizando el dataset de validación de del dataset DSCUCU4 y los mejores pesos estimados en el modelo. En la tabla 10, se muestran los resultados.

Class	Images	Instances	Box Precision (P)	Recall (R)	mAP50	mAP50- 95
all	319	546	0.906	0.939	0.968	0.730
Fisura	319	212	0.894	0.915	0.954	0.739
PielCocodrilo	319	189	0.885	0.968	0.974	0.654
Bache	319	145	0.938	0.933	0.976	0.798

Tabla 8. Precisiones de ejecución de datos de experimento 4.Yolov8. *Class*: Clase evaluada. **Images**: Número de imágenes. **Instances**: Número de instancias detectadas. **Box(P)**: Precisión de la caja delimitadora. **Recall** o tasa de verdaderos positivos. **mAP50**: Precisión media promedio al umbral del 50%. **mAP50-95**: Es similar a mAP50, pero en lugar de considerar un único umbral de IoU del 50%, esta métrica calcula el mAP en varios umbrales desde el 50% hasta el 95% en incrementos del 5%.

7.3. Evaluación del modelo.

A continuación, se realizará la evaluación de los modelos utilizados en cada uno de los experimentos.

Experimento 1

Se realizó la evaluación del modelo del experimento 1, a través del map (mean average precision) [45]. Utilizando el dataset DSCUCU1 y los mejores pesos estimados en el modelo. En la tabla 11, se muestran los datos del cálculo del mAP.

Name	Ap (%)	Tp	Fp
FisuraLateral	31.91	38	46
Fisura	37.14	117	120
PielDeCocodrilo	43.57	55	50
Bache	41.09	39	28

Tabla 9. Precisiones de ejecución de datos de experimento 1. Class ID: Identificador de la clase. Name: Nombre de la clase. AP (%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Teniendo en cuenta los resultados de experimento 1 con Yolov4. Se observa lo siguiente:

- Los resultados muestran que la clase "PielDeCocodrilo" tuvo la mayor precisión promedio (ap) con un 43.57%, seguida por "Bache" con un 41.09%, "Fisura" con un 37.14%, y "FisuraLateral" con un 31.91%.
- La clase "Fisura" tuvo el mayor número de verdaderos positivos (TP) con 117, pero también el mayor número de falsos positivos (FP) con 120, lo que sugiere que esta categoría es la más comúnmente detectada pero también la que más errores de clasificación tiene.
- "PielDeCocodrilo" y "Bache" muestran un equilibrio más cercano entre TP y FP, lo que indica una mejor precisión en la detección y clasificación de estos tipos de daños en comparación con "FisuraLateral" y "Fisura".

Medida	Valor	Descripción
Precisión	51%	Porcentaje de predicciones positivas que son correctas.
Recall	40%	Porcentaje de positivos reales que son correctamente identificados.
Puntuación F1	45%	Media armónica de precisión y recall, que equilibra ambos.

Verdaderos positivos (TP)	249	Objetos correctamente identificados como positivos.
Falsos positivos (FP)	244	Objetos incorrectamente identificados como positivos.
Falsos negativos (FN)	375	Objetos positivos que no fueron identificados.
mAP@0.50	38.43%	Precisión media promedio a un umbral de IoU de 0.50.

Tabla 10. resultados globales del modelo de experimento 1.

De los resultados globales del modelo del experimento 1, se observa lo siguiente.

- El modelo tiene una precisión (51%) y un recall (40%) moderados, lo que significa que es capaz de identificar correctamente una cantidad limitada de objetos.
- El puntaje F1 del 45% indica que el modelo tiene un equilibrio aceptable entre precisión y recall.
- El modelo identifica correctamente una cantidad considerable de objetos positivos (249 TP), pero también hay un número considerable de falsos positivos (244 FP) y falsos negativos (375 FN).
- El mAP@0.50 del 38.43% indica que el modelo tiene una capacidad moderada para detectar objetos correctamente.

Prueba de detecciones de daños en imágenes utilizando modelo del experimento 1.

Como parte del proceso de evaluación del modelo, se realizaron detecciones utilizando el modelo entrenado en cada experimento, En el experimento 1, las precisiones fueron altas para la clase bache, pero hubo una confusión del modelo a la hora de detectar piel de cocodrilo, en la detección de fisuras, las mismas se realizan con baja precisión y además ocurre un solapamiento entre los bounding boxes, en la figura 47, podemos apreciar las precisiones de cada una de las detecciones, del modelo correspondiente al experimento 1.

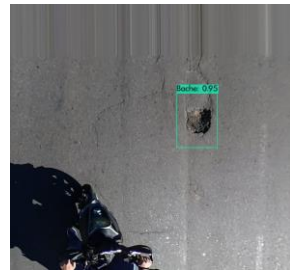


Figura a.



Figura b.

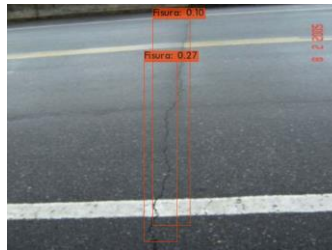


Figura c.

Figura 47. Detecciones utilizando modelo entrenado del experimento 1 Figura a: Detección de baches con precisión del 95% Figura b: Detección de piel de cocodrilo con precisión máxima del 42%, Figura c: Detección de Fisuras con precisión máxima del 27%

Experimento 2

Se realizó la evaluación del modelo del experimento 2, a través del map (mean average precisión) [45]. Utilizando el dataset DSCUCU2 y los mejores pesos estimados en el modelo. En la tabla 13, se muestran los datos del cálculo del mean average precisión mAP.

Name	Ap (%)	Tp	Fp
FisuraLateral	59.44	125	58
Fisura	43.94	125	90
PielDeCocodrilo	73.50	149	59
Bache	90.86	270	40

Tabla 13. Precisiones de ejecución de datos de experimento 2. Class ID: Identificador de la clase. Name: Nombre de la clase. AP (%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Teniendo en cuenta los resultados de experimento 2 con Yolov4. Se observa lo siguiente:

- En total, hubo 4 482 detecciones con 1 036 casos únicos verdaderos (unique_truth_count).
- La clase "Bache" tuvo el mayor porcentaje de precisión promedio (ap) con un 90.86%, seguida por "PielDeCocodrilo" con un 73.50%, "FisuraLateral" con un 59.44%, y "Fisura" con un 43.94%.
- La clase "Bache" también tuvo el mayor número de verdaderos positivos (TP) con 270, mientras que "Fisura" y "FisuraLateral" tuvieron cada uno 125 TP.
- En términos de falsos positivos (FP), la clase "Fisura" tuvo la mayor cantidad con 90, mientras que "Bache" tuvo la menor con 40.

Medida	Valor	Descripción
Precisión	0.73	Porcentaje de predicciones positivas que son correctas.
Recall	0.65	Porcentaje de positivos reales que son correctamente identificados.
Puntuación F1	0.69	Media armónica de precisión y recall, que equilibra ambos.
Verdaderos positivos (TP)	669	Objetos correctamente identificados como positivos.
Falsos positivos (FP)	247	Objetos incorrectamente identificados como positivos.
Falsos negativos (FN)	367	Objetos positivos que no fueron identificados.
mAP@0.50	66.94%	Precisión media promedio a un umbral de IoU de 0.50.

Tabla 11. resultados globales del modelo de experimento 2.

De los resultados globales del modelo del experimento 2, se observa lo siguiente.

- El modelo tiene una precisión decente (0.73) y un recall moderado (0.65), lo que significa que es capaz de identificar correctamente una cantidad razonable de objetos.

- El puntaje F1 del 0.69 indica que el modelo tiene un equilibrio aceptable entre precisión y recall.
- El modelo identifica correctamente una cantidad considerable de objetos positivos (669 TP), pero también hay un número considerable de falsos positivos (247 FP) y falsos negativos (367 FN).
- El mAP@0.50 del 66.94% indica que el modelo tiene un rendimiento decente en la detección de objetos.

Prueba de detecciones de daños en imágenes utilizando modelo del experimento 2.

Continuando con la evaluación del modelo, se realizaron detecciones utilizando el modelo entrenado en el experimento 2, las precisiones son buenas para la clase bache y la clase piel de cocodrilo, en la detección de fisuras, las detecciones se realizan con baja precisión y además ocurre un solapamiento entre los bounding boxes, además hay una detección incompleta de fisuras cuando la longitud de la fisura abarca más del 50% de la imagen en la figura 48, podemos apreciar las precisiones de cada una de las detecciones, del modelo correspondiente al experimento 2.

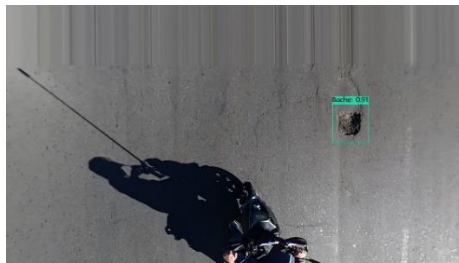


Figura a.



Figura b.

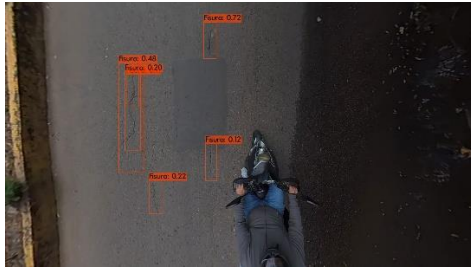


Figura c.



Figura d.

Figura 48. Detecciones utilizando modelo entrenado del experimento 2 Figura a: Detección de baches con precisión del 91%
Figura b: Detección de piel de cocodrilo con precisión del 76%, Figura c: Detección de Fisuras con precisión máxima de 72%,
Figura d. Detección de Fisuras con precisión máximo de 38%.

Experimento 3

Se realizó la evaluación del modelo del experimento 3, a través del map (mean average precisión) [45]. Utilizando el dataset DSCUCU3 y los mejores pesos estimados en el modelo. En la tabla 15, se muestran los datos del cálculo del mAP.

Name	Ap (%)	Tp	Fp
Fisura	70.80	354	147
PielDeCocodrilo	79.98	260	66
Bache	89.10	292	49

Tabla 12. Precisiones de ejecución de datos de experimento 3. Class ID: Identificador de la clase. Name: Nombre de la clase. AP(%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Teniendo en cuenta los resultados de experimento 3 con Yolov4. Se observa lo siguiente:

- Los resultados muestran que la clase "Bache" obtuvo la mayor precisión promedio (ap) con un 89.10%, seguida por "PielDeCocodrilo" con un 79.98%, y "Fisura" con un 70.80%.
- La clase "Fisura" tuvo la mayor cantidad de verdaderos positivos (TP) con 354, mientras que "Bache" y "PielDeCocodrilo" tuvieron 292 y 260 TP respectivamente.

- En términos de falsos positivos (FP), "Fisura" tuvo la mayor cantidad con 147, indicando un mayor número de detecciones incorrectas en comparación con las otras dos clases.

Medida	Valor	Descripción
Precisión	0.78	Porcentaje de predicciones positivas que son correctas.
Recall	0.78	Porcentaje de positivos reales que son correctamente identificados.
Puntuación F1	0.78	Media armónica de precisión y recall, que equilibra ambos.
Verdaderos positivos (TP)	906	Objetos correctamente identificados como positivos.
Falsos positivos (FP)	262	Objetos incorrectamente identificados como positivos.
Falsos negativos (FN)	256	Objetos positivos que no fueron identificados.
mAP@0.50	79.96%	Precisión media promedio a un umbral de IoU de 0.50.

Tabla 13. resultados globales del modelo de experimento 3.

De los resultados globales del modelo del experimento 3, se observa lo siguiente.

- El modelo tiene un buen equilibrio entre precisión y recall, con un puntaje F1 alto de 0.78.
- El modelo identifica correctamente una gran cantidad de objetos positivos (906 TP), pero también hay algunos falsos positivos (262 FP) y falsos negativos (256 FN).
- El mAP@0.50 del 79.96% indica que el modelo tiene un alto rendimiento en la detección de objetos.

Prueba de detecciones de daños en imágenes utilizando modelo del experimento 3.

En la evaluación del modelo desarrollado en el experimento 3, se observó un rendimiento destacado en la detección de clases específicas como baches, piel de cocodrilo y fisuras. Aunque el modelo demostró una buena precisión en general, se identificaron

algunos desafíos: en particular, tendió a confundir las grietas en las señalizaciones del pavimento con la clase "piel de cocodrilo". Además, se notó una limitación en la detección completa de fisuras, especialmente en casos donde estas abarcaban más del 50% de la imagen, como se evidencia en la Figura 49.



Figura a.

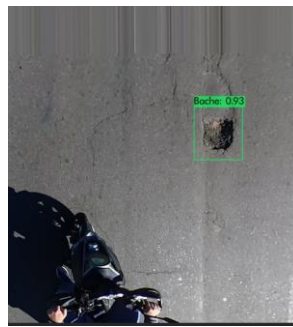


Figura b.

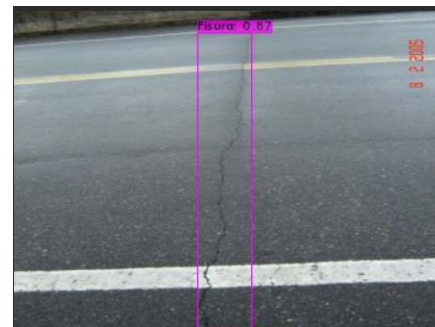


Figura c.



Figura d.

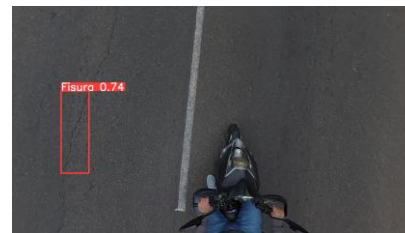


Figura e.

Figura 49. Detecciones utilizando modelo entrenado del experimento 3 Figura a: Detección de fisuras con precisión del 87%
Figura b: Detección de piel de baches con precisión del 93%, Figura c: Detección de piel de cocodrilo con precisión de 72%,
Figura d. Detección de piel de cocodrilo con precisión de 76%. Figura e. Detección de fisura con precisión de 74%

Experimento 4

Se realizó la evaluación del modelo del experimento 4 con Yolov4, a través del map (mean average precisión) [45]. Utilizando el dataset DSCUCU4 y los mejores pesos estimados en el modelo. En la tabla 17, se muestran los datos del cálculo del mAP.

Name	Ap (%)	Tp	Fp
Fisura	88.17	185	42
PielCocodrilo	97.92	190	10
Bache	99.25	126	10

Tabla 14. Precisiones de ejecución de datos de experimento 4 con Yolov4. Class ID: Identificador de la clase. Name: Nombre de la clase. AP (%): Precisión media promedio en porcentaje. TP: Verdaderos positivos. FP: Falsos positivos.

Teniendo en cuenta los resultados de experimento 4 con Yolov4. Se observa lo siguiente:

- La clase "Bache" obtuvo la mayor precisión promedio (ap) con un 99.25%, seguida de "PielCocodrilo" con un 97.92% y "Fisura" con un 88.17%.
- En términos de verdaderos positivos (TP), "PielCocodrilo" lideró con 190, seguida de "Fisura" con 185 y "Bache" con 126.
- La clase "Fisura" tuvo la mayor cantidad de falsos positivos (FP) con 42, mientras que "PielCocodrilo" y "Bache" tuvieron significativamente menos con 10 cada uno.

Medida	Valor	Descripción
Precisión	89%	Porcentaje de predicciones positivas que son correctas.
Recall	93%	Porcentaje de positivos reales que son correctamente identificados.
Puntuación F1	91%	Media armónica de precisión y recall, que equilibra ambos.
Verdaderos positivos (TP)	501	Objetos correctamente identificados como positivos.
Falsos positivos (FP)	62	Objetos incorrectamente identificados como positivos.
Falsos negativos (FN)	35	Objetos positivos que no fueron identificados.
mAP@0.50	95.11%	Precisión media promedio a un umbral de IoU de 0.50.

Tabla 15. Resultados globales del modelo de experimento 4 con Yolov4.

De los resultados globales del modelo del experimento 4 con Yolov4, se observa lo siguiente.

- El modelo tiene un alto nivel de precisión (89%) y recall (93%), lo que indica que es muy bueno para identificar correctamente los objetos.
- El puntaje F1 del 91% confirma que el modelo tiene un buen equilibrio entre precisión y recall.
- El modelo identifica correctamente una gran cantidad de objetos positivos (501 TP), con un número relativamente bajo de falsos positivos (62 FP) y falsos negativos (35 FN).
- El mAP@0.50 del 95.11% es un resultado excelente, lo que significa que el modelo es muy eficaz para detectar objetos.

Prueba de detecciones de daños en imágenes utilizando modelo del experimento 4.

En la evaluación de los modelos del experimento 4, se observaron resultados significativos. Para el modelo entrenado con YOLO v4 [38]. Las detecciones mostraron altas precisiones en las clases bache, piel de cocodrilo y fisuras. Este modelo fue capaz de identificar fisuras más delgadas. Sin embargo, se detectaron algunas limitaciones, como la confusión entre grietas de señalizaciones en el pavimento y la clase piel de cocodrilo, y una detección incompleta de fisuras cuando estas ocupan más del 50% de la imagen, como se muestra en la Figura 50.



Figura a.

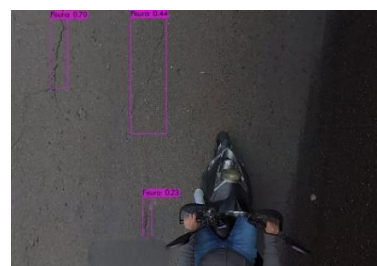


Figura b.



Figura c.



Figura d.

Figura 50. Detecciones utilizando modelo entrenado con yolo v4 del experimento 4 Figura a: Detección de piel de cocodrilo y bache con precisión del 94% y 99% respectivamente Figura b: Detección de fisura con precisión máxima del 70%, Figura c: Detección de fisura con precisión máxima del 37%, Figura d. Detección de piel de cocodrilo con precisión máxima de 59%.

Ahora realizaremos la evaluación del modelo del experimento 4 con Yolov8 [39]. A través del map (mean average precisión) [45]. Utilizando el dataset DSCUCU2 y los mejores pesos estimados en el modelo. En la tabla 19, se muestran los datos del cálculo del Map.

Class	Images	Instances	Box Precision	Recall	mAP50	mAP50-95
all	319	546	0.906	0.939	0.968	0.730
Fisura	319	212	0.894	0.915	0.954	0.739
PielCocodrilo	319	189	0.885	0.968	0.974	0.654
Bache	319	145	0.938	0.933	0.976	0.798

Tabla 16. Precisiones de ejecución de datos de experimento 4 con yolo v8. **Class:** Clase evaluada. **Images:** Número de imágenes. **Instances:** Número de instancias detectadas. **Box(P):** Precisión de la caja delimitadora. **Recall** o tasa de verdaderos positivos. **mAP50:** Precisión media promedio al umbral del 50%. **mAP50-95:** Similar A mAP50, pero en lugar de considerar un único umbral de IoU del 50%, esta métrica calcula EL mAP en varios umbrales desde el 50% hasta el 95% en incrementos del 5%.

Teniendo en cuenta los resultados de experimento 4 con Yolov8, de la tabla 19. Se observa lo siguiente:

- La precisión de la caja delimitadora (Box Precision) fue del 90.6%, y el recall del 93.9%.

- La precisión media promedio (mAP) a un umbral de IoU del 50% (mAP50) fue del 96.8%, mientras que para un umbral de IoU entre 50% y 95% (mAP50-95) fue del 73.0%.
- Para la clase "Fisura" (212 instancias):
- La precisión fue del 89.4%, con un recall del 91.5%.
- mAP50 fue del 95.4% Y mAP50-95 fue del 73.9%.
- Para la clase "PielCocodrilo" (189 instancias):
- La precisión fue del 88.5%, con un recall del 96.8%.
- mAP50 fue del 97.4% Y mAP50-95 fue del 65.4%.
- Para la clase "Bache" (145 instancias):
- La precisión fue del 93.8%, con un recall del 93.3%.
- mAP50 fue del 97.6% Y mAP50-95 fue del 79.8%.

El modelo entrenado con YOLO v8 [39]. También exhibió un alto grado de precisión en todas las clases, incluyendo una mejora notable en la detección de fisuras delgadas. Al igual que con YOLO v4 [38]. Se presentó la confusión entre las grietas de señalizaciones y la clase piel de cocodrilo, y se observó una detección incompleta de fisuras de gran longitud, evidenciado en la Figura 51.

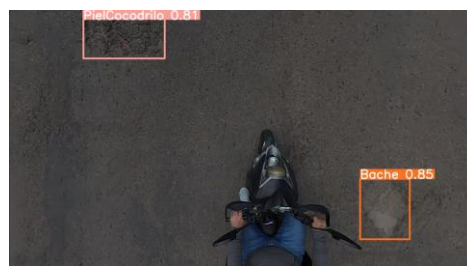


Figura a.

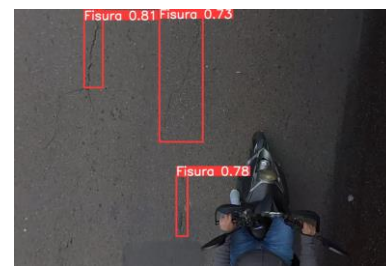


Figura b.

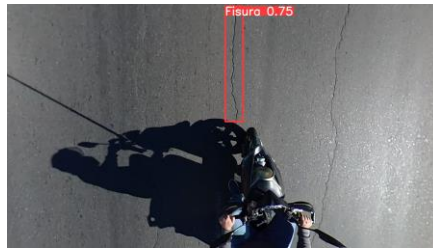


Figura c.

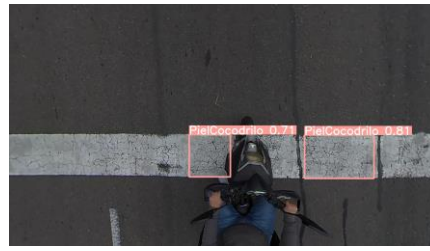


Figura d.

Figura 51. Detecciones utilizando modelo entrenado con yolo v8 del experimento 4. Figura a: Detección de piel de cocodrilo y baches con precisión del 81% y 85% respectivamente. Figura b: Detección de fisuras con precisión máxima del 81%. Figura c: Detección de fisura con precisión de 75%. Figura d: Detección de piel de cocodrilo con precisión máxima de 81%. Figura e: Detección de fisura con precisión de 74%

Los resultados de la valoración demostraron que ambos modelos son altamente eficientes en la detección de daños. Sin embargo, YOLO v8 [39.] Mostró una ligera ventaja sobre YOLO v4 [38]. En términos de eficiencia computacional y precisión, especialmente en la detección de fisuras más pequeñas y menos obvias. Esto fue un hallazgo significativo, ya que indicó que YOLO v8 [39]. Podría ser más apropiado para aplicaciones en entornos de producción donde se requiere una alta precisión y eficiencia.

Es importante resaltar que el principal desafío identificado a lo largo de todo el proceso de entrenamiento estuvo relacionado con el tamaño de las imágenes de entrada. Inicialmente, se adoptó YOLOv4 [38]. Con un tamaño máximo de entrada de 416x416 y, en algunos casos, 608x608, aprovechando una GPU de 40 GB. Posteriormente, al migrar a YOLOv8 [39]. Se logró incrementar significativamente el tamaño de la imagen de entrada hasta 1216, manteniendo la misma GPU de 40 GB. Este cambio marcó una diferencia notable, especialmente debido a las dimensiones reducidas de las fisuras y la baja cantidad de píxeles en estas áreas. La dificultad que presentaba YOLOv4 [38]. Para detectar fisuras en imágenes pequeñas se superó de manera sustancial con las mejoras implementadas en YOLOv8 [39]. Este último no solo posibilita entrenamientos más rápidos, sino que también brinda detecciones más precisas, todo ello utilizando una menor cantidad de recursos hardware.

7.3.2. Evaluación de Equidad de las Predicciones del Modelo

En esta sección nos enfocamos en la evaluación comparativa de las predicciones de los modelos YOLOv4 [38]. Y YOLOv8 [39]. Ambos entrenados con el mismo conjunto de datos, DSCUCU4, que comprende imágenes de daños en pavimento. La métrica clave utilizada para esta evaluación es la precisión media promedio (mAP) [45]. Un estándar en la detección de objetos que proporciona una evaluación integral de la precisión y recall [43]. A varios umbrales de IoU (Intersección sobre Unión).

Se empleó el conjunto de datos DSCUCU4, enfocado en tres tipos de daños en pavimento: Fisura, PielCocodrilo y Bache. Los modelos YOLOv4 [38] Y YOLOv8 [39]. Fueron entrenados bajo condiciones idénticas para garantizar una comparación equitativa. El rendimiento de cada modelo se evaluó utilizando el Map [45]. En la tabla 20. Podemos apreciar las precisiones mAP de Modelo entrenado yolov4 y en la tabla 21. Los resultados para Yolov8.

Resultados YOLOv4.

- Detecciones Totales: 1 710
- Casos de Verdad Únicos: 536
- Precisión Media Promedio (mAP@0.50): 95.11%

Clase	AP (%)
Fisura	88.17
PielCocodrilo	97.92
Bache	99.25

Tabla 20. Precisiones mAP de Modelo entrenado yolov4.

- Precision: 89%
- Recall: 93%

- F1-score: 91%
- IoU promedio: 69.87%

Resultados YOLOv8.

- Imágenes Evaluadas: 319
- Instancias Detectadas: 546

Clase	Box(P)	R	mAP50	mAP50-95
Todas	0.906	0.939	0.968	0.73
Fisura	0.894	0.915	0.954	0.739
PielCocodrilo	0.885	0.968	0.974	0.654
Bache	0.938	0.933	0.976	0.798

Tabla 21. Precisiones mAP de Modelo entrenado yolov8.

La comparación de YOLOv4 [38] Y YOLOv8 [39]. muestra diferencias significativas en sus capacidades de detección. YOLOv8 demostró una mayor precisión en la detección de clases específicas, como se evidencia en sus valores mAP superiores para cada clase. Sin embargo, es esencial considerar que, aunque YOLOv8 parece superar a YOLOv4 en términos de Map [45]. Ambos modelos presentan fortalezas únicas. Por ejemplo, YOLOv4 mostró un F1-score [44]. Ligeramente superior, indicando un equilibrio más cercano entre precisión y recall [43].

7.3.3. Uso en un Entorno de Producción

La implementación de los modelos en entornos de producción es el objetivo final de este proyecto. La evaluación exhaustiva y la interpretación detallada de los modelos indicaron que están listos para ser implementados en sistemas de mantenimiento y reparación de carreteras. Los modelos pueden ser utilizados para analizar automáticamente imágenes de carreteras y detectar daños, proporcionando así una herramienta valiosa para

los ingenieros y planificadores urbanos. Esta capacidad puede mejorar significativamente la eficiencia en la identificación de áreas que necesitan mantenimiento, lo que a su vez puede conducir a una mejor asignación de recursos y una gestión más eficaz de las infraestructuras viales.

Se seleccionó el modelo entrenado con YOLO v8 [39]. Para la detección de daños en pavimentos flexibles. Se utilizó un conjunto de datos de 19 300 imágenes, tomadas a intervalos de 4 metros, para realizar inferencias utilizando tres clases de daños en los pavimentos. El modelo se configuró para hacer predicciones con un umbral de precisión del 70%, lo que significa que sólo se consideraron las detecciones que el modelo calculó con una confianza de al menos el 70%.

En la tabla 22. Se muestra la distribución de instancias para cada clase de daño:

Clase ID	Descripción de la Clase	Número de Instancias Detectadas
0	Fisura	1127
1	Piel de Cocodrilo	104
2	Bache	435
Total		1666

Tabla 22. Cantidad de detecciones realizadas por el modelo entrenado con YOLO v8

Estos resultados proporcionan una visión cuantitativa clara de las instancias detectadas para cada tipo de daño en los pavimentos. La Clase 0, correspondiente a 'Fisura', presentó la mayor cantidad de detecciones, seguida por la Clase 2, 'Bache', y la Clase 1, 'Piel de Cocodrilo'. Estos hallazgos son esenciales para comprender la prevalencia y distribución de diferentes tipos de daños en el conjunto de datos analizado y ser fundamentales para la planificación de estrategias de mantenimiento y reparación de carreteras.

7.4 Implementación del sistema de georreferenciación.

7.4.1. Georreferenciación de los Daños Identificados en las Imágenes

La georreferenciación de los daños identificados en las imágenes fue un proceso complejo que implicó el uso de datos obtenidos del modelo de aprendizaje profundo. Cada daño detectado se representó en forma de bounding box, proporcionado por el modelo Yolov8 [39]. Con coordenadas precisas del punto medio. Estas coordenadas se utilizaron junto con la ubicación desde donde se tomó la imagen (punto medio inferior) para calcular la dirección y la distancia entre estos dos puntos. Además, el azimut de la cámara proporcionó la orientación necesaria para calcular las coordenadas reales de los daños en el sistema de coordenadas MAGNA-SIRGAS, Origen Nacional [73].

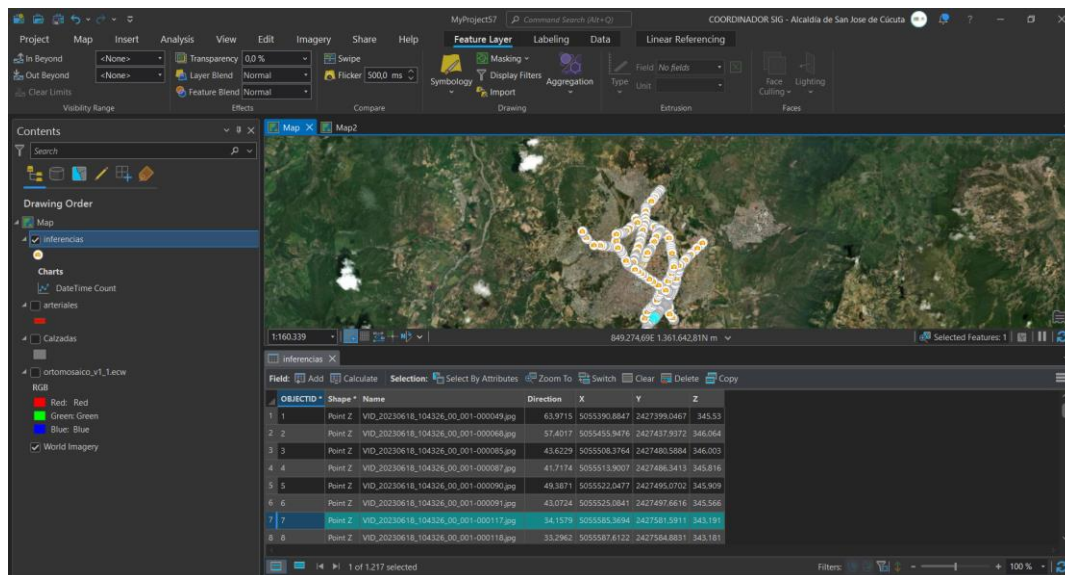


Figura 52. Coordenadas y Azimut de la Cámara en cada fotografía

Este proceso fue clave para vincular los datos visuales con su ubicación geográfica precisa, permitiendo no solo la identificación de los daños, sino también su posicionamiento exacto en el mapa. La precisión en esta etapa fue crucial, ya que cualquier

error en la georreferenciación podría haber llevado a conclusiones incorrectas sobre la ubicación y la extensión de los daños en las vías. La combinación de las coordenadas de los bounding boxes con los datos de orientación de la cámara aseguró que la georreferenciación de los daños fuera lo más precisa posible.

En la tabla 23. Se muestra los datos de distancias, rumbos y áreas de los daños detectados.

Archivo	Clase Nombre	Distancia	Distancia m	Rumbo deg
VID_20230618_104326_00_001-000049.jpg	Fisura	1312,33	3,94	43,74
VID_20230618_104326_00_001-000068.jpg	Bache	639,19	1,92	65,89
VID_20230618_104326_00_001-000068.jpg	PielCocodrilo	1092,65	3,28	-26,57
VID_20230618_104326_00_001-000085.jpg	Fisura	558,49	1,68	-31,16
VID_20230618_104326_00_001-000087.jpg	Bache	1087,05	3,26	-48,89
VID_20230618_104326_00_001-000090.jpg	Fisura	776,19	2,33	-21,00
VID_20230618_104326_00_001-000091.jpg	Bache	1161,78	3,49	-37,64
VID_20230618_104326_00_001-000117.jpg	Fisura	789,23	2,37	-48,02
VID_20230618_104326_00_001-000118.jpg	Fisura	1083,82	3,25	-32,54
VID_20230618_104326_00_001-000118.jpg	Fisura	343,98	1,03	-39,35

Tabla 23. Tabla con distancias, rumbos y áreas de los daños detectados

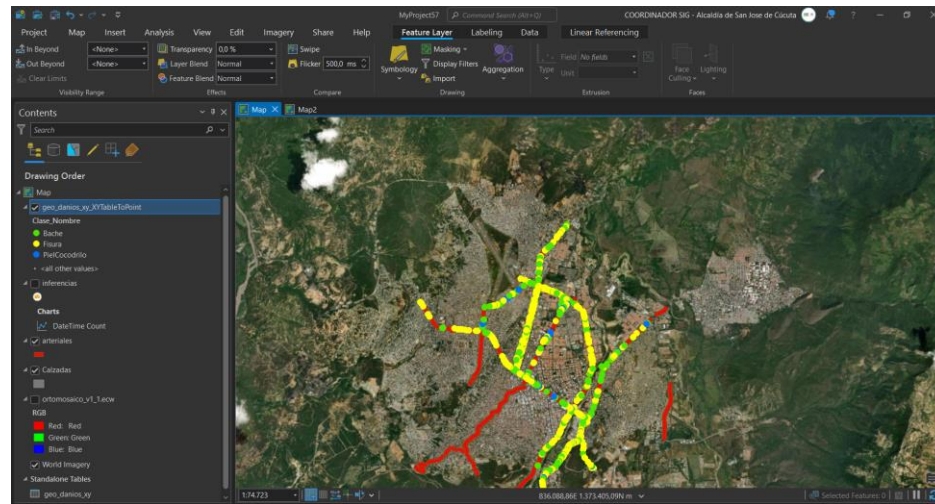


Figura 53. Daños detectados y georreferenciados

7.4.2. Cuantificación de las Longitudes y Áreas de los Daños Identificados en las Imágenes

El siguiente paso fue la cuantificación de las longitudes y áreas de los daños identificados en las imágenes. Conociendo la equivalencia entre la longitud de un píxel en las imágenes y la realidad en terreno, se pudo calcular las dimensiones reales de los daños. Para los daños de tipo piel de cocodrilo y baches, se calculó el área del rectángulo del bounding box, mientras que para las fisuras se consideró la longitud del lado mayor.

Este proceso de cuantificación fue esencial para entender la gravedad y el impacto de los daños en las vías. La capacidad de medir con precisión el tamaño de los daños permitió una evaluación más detallada de su severidad, lo que es crucial para la planificación de las reparaciones y el mantenimiento de las vías. Además, la cuantificación exacta de los daños proporcionó una base de datos valiosa para análisis estadísticos y comparativos, facilitando la identificación de patrones y tendencias en los daños en pavimentos.

En la tabla 24. Se muestra los datos de Distancias de Anchos y Largos de los bounding box de daños detectados.

Archivo	Clase Nombre	Ancho BB px	Ancho BB m	Largo BB px	Largo BB m	Area m2
VID_20230618_104326_00_001-000049.jpg	Fisura	103,40	0,31	246,76	0,74	0,23
VID_20230618_104326_00_001-000068.jpg	Bache	204,41	0,61	246,53	0,74	0,45
VID_20230618_104326_00_001-000068.jpg	PielCocodrilo	334,27	1,00	199,28	0,60	0,60
VID_20230618_104326_00_001-000085.jpg	Fisura	37,42	0,11	218,35	0,66	0,07
VID_20230618_104326_00_001-000087.jpg	Bache	127,21	0,38	220,53	0,66	0,25
VID_20230618_104326_00_001-000090.jpg	Fisura	50,91	0,15	451,16	1,35	0,21
VID_20230618_104326_00_001-000091.jpg	Bache	201,85	0,61	206,70	0,62	0,38
VID_20230618_104326_00_001-000117.jpg	Fisura	42,30	0,13	227,12	0,68	0,09
VID_20230618_104326_00_001-000118.jpg	Fisura	77,91	0,23	319,05	0,96	0,22
VID_20230618_104326_00_001-000118.jpg	Fisura	46,14	0,14	249,22	0,75	0,10

Tabla 17. Distancias de Anchos y Largos de los bounding box de daños detectados.

7.4.4. Representación Cartográfica y Creación de Cuadro de Mando de Daños

La representación cartográfica se ha establecido como un componente fundamental en la visualización y análisis de los daños detectados en las vías. Utilizando el robusto software ArcGIS Online [92]. Hemos desarrollado mapas detallados que incorporan la ubicación precisa de cada daño identificado. Estos mapas no solo resaltan la ubicación de los daños, sino que también proporcionan un contexto visual del entorno en el que se encuentran. Esto es crucial para entender la distribución geográfica y la concentración de

daños en diferentes áreas de la ciudad.

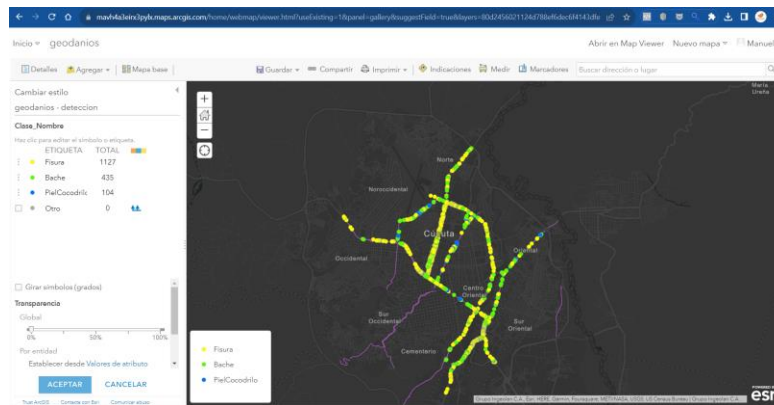


Figura 54. Publicación de capas geográficas en ArcGIS Online

En el proceso de creación de estos mapas, se prestó especial atención a asegurar que la información fuese tanto precisa como intuitiva. Se utilizaron simbologías y colores específicos para diferenciar entre los distintos tipos de daños - fisuras, piel de cocodrilo y baches. Esto permite a los usuarios, desde técnicos de campo hasta administradores ordenadores de gasto público, obtener una comprensión rápida y clara del estado actual de las vías. Además, la interactividad de los mapas online permite a los usuarios explorar diferentes áreas con facilidad, acercarse para obtener detalles más precisos o alejarse para obtener una visión general.

El cuadro de mando (dashboard) [96]. Se ha diseñado como una herramienta dinámica e interactiva para la visualización y análisis de los datos recopilados. Este cuadro de mando, alojado en ArcGIS Online [92]. Presenta una variedad de indicadores y gráficos que reflejan la cantidad y tipos de daños, así como su distribución geográfica. El objetivo de este cuadro de mando es proporcionar a los usuarios una herramienta de análisis y toma de decisiones en tiempo real, que sea tanto informativa como fácil de usar.

Los indicadores incluyen, pero no se limitan a, el número total de daños detectados, la distribución porcentual de cada tipo de daño y la densidad de daños por cada segmento de vía. Los gráficos y tablas complementan estos indicadores, ofreciendo un desglose más detallado de los datos. Por ejemplo, los gráficos de barras y los diagramas circulares muestran la proporción de cada tipo de daño en diferentes sectores de la ciudad. Esto es particularmente útil para identificar patrones o áreas que requieren atención prioritaria.



Figura 55. Cuadro de mando daños detectados

El cuadro de mando [96]. Es más que una simple herramienta de visualización; es una plataforma integral para la gestión y planificación estratégica del mantenimiento de carreteras. Al proporcionar acceso instantáneo a datos actualizados y relevantes, permite a los responsables de la toma de decisiones implementar acciones correctivas de manera más eficiente. Por ejemplo, los equipos de mantenimiento de carreteras pueden utilizar esta información para planificar sus rutas de inspección y reparación, asegurando que los recursos se asignen de manera efectiva.

8. CONCLUSIONES Y TRABAJOS FUTUROS

7.1. CONCLUSIONES

El proyecto "Detección, Georreferenciación y Cuantificación de Daños en Pavimentos Flexibles a Partir de Imágenes Utilizando Aprendizaje Profundo" ha demostrado ser un avance significativo en la aplicación de tecnologías de visión por computadora y aprendizaje profundo en el ámbito de la infraestructura vial y la movilidad urbana.

El uso del modelo YOLO v8 para la detección de daños en pavimentos flexibles representa un cambio significativo respecto a los métodos tradicionales de inspección visual. La automatización del proceso de detección y cuantificación de daños ofrece una forma más eficiente, rápida y objetiva de evaluar el estado de las carreteras.

La implementación de YOLO v8 demostró ser eficaz en la detección de una variedad de daños en pavimentos, incluyendo fisuras, piel de cocodrilo y baches. Con un alto grado de precisión y un umbral de confianza del 70%, el modelo identificó con éxito 1,127 instancias de fisuras, 435 de baches y 104 de piel de cocodrilo. Esta precisión fue esencial para una evaluación fiable del estado de las carreteras y para la toma de decisiones informadas en cuanto a mantenimiento y reparaciones.

La capacidad de identificar y cuantificar con precisión los daños en las carreteras tiene implicaciones directas en la seguridad vial. Al detectar y abordar los problemas en las etapas iniciales, se pueden prevenir accidentes y mejorar la seguridad de los usuarios de la carretera. Además, el modelo facilita la planificación y priorización de las labores de mantenimiento, lo que puede resultar en un uso más eficiente de los recursos.

La incorporación de sistemas SIG y GPS para la georreferenciación de los daños detectados agregó un valor considerable al proyecto. La capacidad de ubicar con precisión los daños en un contexto geográfico facilitó una planificación y ejecución de las estrategias de mantenimiento vial.

En resumen, este proyecto ha demostrado cómo la combinación de técnicas avanzadas de aprendizaje profundo y visión por computadora, junto con sistemas de georreferenciación sofisticados, puede mejorar significativamente la forma en que se gestionan y mantienen las infraestructuras viales. Estos avances no solo optimizan los procesos existentes, sino que también abren nuevas posibilidades para mejorar la seguridad vial y la eficiencia en la gestión de carreteras.

7.2. TRABAJOS FUTUROS

Este proyecto no solo aborda los desafíos actuales en la gestión de infraestructuras viales, sino que también sienta las bases para futuras investigaciones y aplicaciones. La metodología y las tecnologías empleadas en este proyecto pueden adaptarse y ampliarse para abordar otras necesidades en el campo de la ingeniería civil y la planificación urbana.

Además, este proyecto no solo está a la vanguardia de las innovaciones actuales, sino que también se posiciona como un pionero en el desarrollo de futuras tecnologías de detección y análisis espacial. Con la ambición de integrar la detección en 3D y la utilización de nubes de puntos y fotogrametría, el proyecto se expandirá para enfrentar desafíos aún más complejos en la gestión de infraestructuras. Una evolución clave será la inclusión de nuevos tipos de daños a detectar, ampliando significativamente el alcance de nuestra investigación más allá de los tres tipos de daños abordados inicialmente. Esta expansión no solo enriquecerá la precisión y eficiencia en la evaluación de las infraestructuras viales, sino que también abrirá nuevas avenidas en el análisis y planificación urbana. Con estos avances, ingenieros y planificadores urbanos estarán equipados para abordar los desafíos emergentes con una comprensión más profunda, promoviendo el desarrollo sostenible y la innovación en el diseño y mantenimiento de infraestructuras urbanas.

9. REFERENCIAS BIBLIOGRAFICAS

- [1] Miller, T. R., & Zaloshnja, E. (2009). On a crash course: The dangers and health costs of deficient roadways. Technical Report, The Transportation Construction Coalition, United States. Retrieved from: <http://www.infrastructureusa.org/on-a-crash-course-the-dangers-and-health-costs-of-deficient-roadways/>
- [2] Rojo, M., Gonzalo-Orden, H., Linares, A., et al. (2018). Impact of a lower conservation budget on road safety indices. *Journal of Advanced Transportation*, 2018, 1-12. <https://doi.org/10.1155/2018/9570465>
- [3] Razón Pública. (2022, December 2). Así están las carreteras colombianas. Retrieved from <https://razonpublica.com/asi-estan-las-carreteras-colombianas/>
- [4] El Tiempo. (2023, January 25). Las principales razones por las que se rompen frecuentemente las calles en Bogotá. Retrieved from <https://www.eltiempo.com/archivo/documento/CMS-4489577>
- [5] Maquinter. (2023, December 1). Daños en las carreteras, tipos más habituales y posibles causas. Retrieved from <https://maquinter.es/danos-en-las-carreteras-tipos-y-posibles-causas/>
- [6] El País. (2022, July 12). Las carreteras de Colombia, entre las más lentas del mundo por la corrupción y las construcciones de mala calidad. Retrieved from <https://elpais.com/america-colombia/2022-07-12/las-carreteras-de-colombia-entre-las-mas-lentas-del-mundo-por-la-corrupcion-y-las-construcciones-de-mala-calidad.html>
- [7] Ministerio de Transportes y Comunicaciones del Perú. (2014). Manual de Carreteras: Mantenimiento o Conservación Vial. Resolución Directoral N° 08-2014 MTC/14. Lima, Perú: Ministerio de Transportes y Comunicaciones del Perú.
- [8] Secretaría de Infraestructura Municipal de Cúcuta. (2022). Inventario Vial Urbano. Cúcuta, Colombia: Recuperado de <https://alccucuta.maps.arcgis.com/apps/dashboards/e17e150c546148b48adc0a845e2724e8>
- [9] Tello-Cifuentes, L., Aguirre-Sánchez, M., Díaz-Paz, J. P., & Hernández, F. (2021). Damage Evaluation in Flexible Pavement Using Terrestrial Photogrammetry and Neural Networks. *TecnoLógicas*, 24(50), 1-14
- [10] Quirita, V. H., Achanccaray Diaz, P. M., Arauco C, S., & Soto Vega, P. J. (2022, December). Desafíos del Aprendizaje Profundo en la Visión por Computador. En Congreso Internacional de Ingeniería de Sistemas (CIIS 2022).
- [11] Repsol. (2023, January 31). ¿Qué es la inteligencia artificial y cómo nos ayuda? [Página web]. Repsol. <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/inteligencia-artificial/index.cshtml>
- [12] Reyes-Ortiz, O. J., Mejía, M., & Useche-Castelblanco, J. S. (2019). Técnicas de inteligencia artificial utilizadas en el procesamiento de imágenes y su aplicación en el análisis de pavimentos. *Revista EIA*, 16(31), 189-207.
- [13] González-Marco, C., García-Segura, T., López-Marco, J. R., Mansanet, J., & Sánchez-Robles,

- J. (2023, July). Integrated Management System for the Predictive Maintenance of Interurban Roads Using Artificial Intelligence Techniques. En 27th International Congress on Project Management and Engineering (pp. 1-10). Donostia-San Sebastián, España: AEIPRO
- [14] Zaera Villanueva, I. (2023). Propuesta de plan de mantenimiento del pavimento basado en herramientas predictivas y técnicas de inteligencia artificial. Aplicación a la carretera A-7 desde PK 313+ 000 hasta PK 326+ 000 (provincia de Valencia) (Tesis doctoral, Universitat Politècnica de València).
- [15] Mayorga Ballesteros, D. C. (2023). Diseño de un modelo de aprendizaje de máquina para reconocimiento y clasificación de daños en pavimento para vías de Sogamoso (Tesis doctoral, Universidad Nacional de Colombia).
- [16] S. Alzate, "Sistema de Clasificación de Severidad de Daños en Pavimentos Flexibles para Determinar Posibles Intervenciones," Trabajo de grado para optar al título de Ingeniería Civil, Universidad EIA, Envigado, 2019.
- [17] "Manual para la inspección visual de pavimentos flexibles". Portal INVÍAS - Colombia. [Online]. Disponible en: <https://www.invias.gov.co/index.php/archivo-y-documentos/documentos-tecnicos/manuales-de-inspeccion-de-obras/974-manual-para-la-inspeccion-visual-de-pavimentos-flexibles> [Accedido el 7 de diciembre de 2022].
- [18] A. Basavaraju, J. Du, Z. Fujie, & J. Jim, "A Machine Learning Approach to Road Surface Anomaly Assessment Using Smartphone Sensors," *IEEE SENSORS JOURNAL*, 20(5), 2020.
- [19] A. M. A.-S. Ahmed & H. Tao, "Review of Deep Convolution Neural Network in Image Classification". En *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications*, 2017.
- [20] I. Goodfellow, Y. Bengio, & A. Courville, "Deep learning," MIT Press, 2016.
- [21] Y. LeCun, Y. Bengio, & G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
- [22] J. Schmidhuber, "Deep learning in neural networks: *An overview*," *Neural Networks*, vol. 61, pp. 85-117, 2015.
- [23] Y. LeCun, Y. Bengio, & G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
- [24] C. M. Bishop, "Pattern recognition and machine learning," Springer Science & Business Media, 2006.
- [25] I. Goodfellow, Y. Bengio, & A. Courville, "Deep learning," MIT Press, 2016.
- [26] Built In, 'What is PyTorch?,' Built In, [En línea]. Disponible: <https://builtin.com/data-science/what-is-pytorch>."
- [27] Built In, 'PyTorch in the industry,' Built In, [En línea]. Disponible: <https://builtin.com/data-science/pytorch-industry>."
- [28] Built In, 'Reasons to use PyTorch,' Built In, [En línea]. Disponible: <https://builtin.com/data-science/reasons-use-pytorch>."
- [29] Built In, 'Building blocks of PyTorch,' Built In, [En línea]. Disponible: <https://builtin.com/data-science/building-blocks-pytorch>."
- [30] Built In, 'PyTorch ecosystem,' Built In, [En línea]. Disponible: <https://builtin.com/data-science/pytorch-ecosystem>."

[science/pytorch-ecosystem."](#)

[31] TensorFlow, "Paid Colab users can now choose between a standard or premium GPU in Colab," in The TensorFlow Blog, 2023.

[32] J. Redmon, S. Divvala, R. Girshick, & A. Farhadi, "You only look once: Unified, real-time object detection." *En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, Las Vegas, NV, USA, 2016.

[33] J. Redmon & A. Farhadi, "YOLO9000: Better, faster, stronger." *En Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263-7271, Honolulu, HI, USA, 2017.

[34] J. Redmon & A. Farhadi, "YOLOv3: An incremental improvement". arXiv preprint arXiv:1804.02767, 2018.

[35] Open Source For You, 'Darknet framework overview,' Open Source For You, [En línea]. Disponible: <https://www.opensourceforu.com/tag/darknet-framework-overview/>

[36] Open Source For You, 'Working with Darknet,' Open Source For You, [En línea]. Disponible: <https://www.opensourceforu.com/tag/working-with-darknet/>

[37] Open Source For You, 'Real-time object detection in Darknet,' Open Source For You, [En línea]. Disponible: <https://www.opensourceforu.com/tag/real-time-object-detection-in-darknet/>

[38] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[39] J. R. Terven and D. M. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," Instituto Politecnico Nacional, Oct. 10, 2023.

[40] "How to train (to detect your custom objects)," GitHub. [En línea]. Disponible en: <https://github.com/AlexeyAB/darknet>. [Accedido: 08-Dic-2023].

[41] "Arguments" Ultralytics YOLOv8 Docs. [Online]. Available: <https://docs.ultralytics.com>

[42] P. Hidayatullah, "Train the YOLOv8 Model with Default Data Augmentation," en YOLOv8 Performance Improvement Masterclass 2023, Udemy, 2023. [Online]. Available: <https://www.udemy.com/course/yolo-performance-improvement-masterclass/learn/lecture/40418758#overview>

[43] Lebidzinski, P. (2021, June 24). A Single Number Metric for Evaluating Object Detection Models. Towards Data Science. [Online]. Available: <https://towardsdatascience.com/a-single-number-metric-for-evaluating-object-detection-models-c97f4a98616d>

[44] Deval Shah (2022). Mean Average Precision (mAP) Explained: Everything You Need to Know. V7 Labs. [Online]. Available: <https://www.v7labs.com/blog/mean-average-precision>

[45] Shah, D. (2022, March 7). Mean Average Precision (mAP) Explained: Everything You Need to Know. V7 Labs. [Online]. Available: <https://www.v7labs.com/blog/mean-average-precision>

[46] Roboflow, 'What is LabelImg?,' en Roboflow Blog, [En línea]. Disponible: <https://blog.roboflow.com/labelimg>

[47] "Object Detection Datasets Overview," Ultralytics YOLOv8 Docs. [Online]. Available: <https://docs.ultralytics.com>

[48] "Collect and Label Images to Train a YOLOv5 Object Detection Model," wandb.ai. [Online]. Available: <https://wandb.ai>

- [49] "How to Create a Dataset for Object Detection using the YOLO Labeling Format," Cogniflow Docs. [Online]. Available: <https://docs.cogniflow.ai>
- [50] "Train YOLOv8 Instance Segmentation on Custom Data," LearnOpenCV. [Online]. Available: <https://learnopencv.com>
- [51] "What is the Pascal VOC XML Annotation Format?," Roboflow. [Online]. Available: <https://roboflow.com/formats/pascal-voc-xml>
- [52] R. C. Gonzalez & R. E. Woods, "Digital image processing," 3rd ed., Pearson Education, 2008.
- [53] A. K. Jain, R. Kasturi, & B. G. Schunck, "Machine vision," vol. 1, McGraw-Hill, 1995.
- [54] W. K. Pratt, "Digital image processing," 4th ed., Wiley, 2001.
- [55] R. I. Hartley & A. Zisserman, "Multiple view geometry in computer vision," 2nd ed., Cambridge University Press, 2004.
- [56] Y. Ma, S. Soatto, J. Kosecka, & S. S. Sastry, "An invitation to 3-d vision: From images to geometry," Springer Science & Business Media, 2004.
- [57] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000.
- [58] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, y F. Shen, "Image Data Augmentation for Deep Learning: A Survey", arXiv:2204.08610v2 [cs.CV], 5 de Noviembre de 2023. Disponible en: <https://ar5iv.labs.arxiv.org/html/2204.08610>
- [59] M. Pal & P. M. Mather, "Supervised classification of remotely sensed data: A review," *Progress in Physical Geography*, vol. 30, no. 2, pp. 157-190, 2006.
- [60] C.-W. Hsu, C.-J. Lin, & C.-C. Lee, "A practical guide to support vector classification," Technical Report, Department of Computer Science and Information Engineering, National Taiwan University. Disponible en: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [61] "Weka 3: Data mining software in Java." (s.f.). Recuperado de <https://www.cs.waikato.ac.nz/ml/weka/>
- [62] T. R. Hasan, A. Z. M. Salem, & M. F. A. Rasid, "Detection and classification of pavement cracks using deep learning." *En Proceedings of the International Conference on Computer and Information Science*, pp. 456-461, Kuala Lumpur, Malaysia, 2017.
- [63] Y. L. Chen, R. Mohammad, R. Jahanshahi, & M. Preetham, "Inexpensive multimodal sensor fusion system for autonomous data acquisition of road surface conditions," *IEEE Sensors Journal*, vol. 16, no. 22, pp. 7731-7743, 2016.
- [64] Xataka. (2020). "Nueva Insta360 One X2: características, precio y ficha técnica". [En línea]. Disponible en: <https://www.xataka.com> [Consultado el 3 de diciembre de 2023].
- [65] insta360. (2023). "Insta360 STUDIO 2023 Software de edición 360". [En línea]. Disponible en: <https://www.insta360.com> [Consultado el 3 de diciembre de 2023].
- [66] Ryte Digital Marketing Wiki. (2023). "¿Qué es un Video 360?". [En línea]. Disponible en: <https://es.ryte.com> [Consultado el 3 de diciembre de 2023].
- [67] Wikipedia. (2023). "Geoetiquetado". [En línea]. Disponible en: <https://es.wikipedia.org> [Consultado el 3 de diciembre de 2023].
- [68] D. Arya, H. Maeda, S. Ghosh, D. Toshniwal y. Sekimoto, "RDD2022: A multi-national image dataset for automatic Road Damage Detection," 2022.

- [69] P. A. Burrough, "Principles of geographical information systems for land resources assessment," Oxford University Press, 1986.
- [70] M. F. Goodchild & S. Gopal (Eds.), "Accuracy of spatial databases," Taylor & Francis, 1989.
- [71] P. A. Longley, M. F. Goodchild, D. J. Maguire, & D. W. Rhind, "Geographic information systems and science," 4th ed., John Wiley & Sons, 2015.
- [72] Sistema de Posicionamiento Global (2023, Febrero 14). [En Línea]. Gobierno de los Estados Unidos. Disponible en: <https://www.gps.gov/spanish.php> [Consultado el 3 de diciembre de 2023].
- [73] Instituto Geográfico Agustín Codazzi. (2020). "Las nuevas especificaciones técnicas para la cartografía oficial de Colombia". [En línea]. Disponible en: <https://www.igac.gov.co> [Consultado el 3 de diciembre de 2023].
- [74] Espacioje. (2023). "¿Qué es Azimut o Acimut y Rumbo en topografía?". [En línea]. Disponible en: <https://www.espacioje.com> [Consultado el 3 de diciembre de 2023].
- [75] LibreTexts Español, '25.1: Sistemas de coordenadas'. Disponible en: <https://espanol.libretexts.org>
- [76]"Vector vs Raster in GIS: What's the Difference?," GIS Geography, [Online]. Available: gisgeography.com.
- [77]"Vector Data Model Definition," Esri Support, [Online]. Available: support.esri.com.
- [78]"Raster data," Introduction to image and raster data—ArcGIS Pro, Esri, [Online]. Available: pro.arcgis.com.
- [79]"What is raster data?," ArcMap, Esri, [Online]. Available: desktop.arcgis.com.
- [80]"What is ArcGIS?," GIS Geography, [Online]. Available: gisgeography.com.
- [81]"About ArcGIS," Esri, [Online]. Available: www.esri.com.
- [82]"ArcGIS," Wikipedia, [Online]. Available: en.wikipedia.org.
- [83]"ArcGIS Pro Overview," Esri, [Online]. Available: www.esri.com.
- [84]"Introduction to ArcGIS Pro," Esri, [Online]. Available: pro.arcgis.com.
- [85]"2D, 3D & 4D GIS Mapping Software | ArcGIS Pro," Esri, [Online]. Available: www.esri.com.
- [86]"GeoTagged Photos To Points (Data Management)," ArcGIS Pro, Esri, [Online]. Available: pro.arcgis.com.
- [87]"GeoTagged Photos To Points (Data Management)," ArcMap, Esri, [Online]. Available: desktop.arcgis.com.
- [88]"How To: Add Geotagged Photos as Point Features in ArcGIS Pro," Esri Support, [Online]. Available: support.esri.com.
- [89]"Spatial Join (Analysis)," ArcGIS Pro, Esri, [Online]. Available: pro.arcgis.com.
- [90]"Spatial Join (Analysis)," ArcMap, Esri, [Online]. Available: desktop.arcgis.com.
- [91]"Add Spatial Join (Data Management)," ArcGIS Pro, Esri, [Online]. Available: pro.arcgis.com.
- [92] An introduction to ArcGIS Online," ArcGIS Online Help, Esri, [Online]. Available: doc.arcgis.com.
- [93] Web GIS Mapping Software | ArcGIS Online," Esri, [Online]. Available: www.esri.com.
- [94] What's new in ArcGIS Online (December 2021)," Esri, [Online]. Available: www.esri.com.
- [95] ArcGIS Online Resources," Esri, [Online]. Available: www.esri.com.
- [96] ArcGIS Dashboards," Esri, [Online]. Available: www.esri.com.

- [97] Introduction to ArcGIS Dashboards," Esri, [Online]. Available: developers.arcgis.com.
- [98]"Create your first dashboard using ArcGIS Dashboards," Esri, [Online]. Available: www.esri.com.
- [99]"Build an interactive dashboard," Learn ArcGIS, [Online]. Available: learn.arcgis.com.
- [100] N. Mohan, "Detection and Quantification of Road Surface Damage using Digital Image Processing Techniques," *International Journal of Engineering Research & Technology*, vol. 6, no. 06, pp. 185–188, 2017.
- [101] C. Banica, S. V. Paturca, & S. D. Grigorescu, "Data Acquisition and Image Processing System for Surface Inspection," en *ROmania, IEEE Xplore*, pp. 28-33, 2017.
- [102] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *IEEE Transactions on Computers*, vol. 46, pp. 695-701, Jun. 1997.
- [103] Zúñiga Guisado, Y. (2022). Deep Learning Para la Detección de Fallas en Pavimentos de una Zona del Distrito de Villa María del Triunfo 2022 LIMA – PERÚ. Tesis de grado, Universidad César Vallejo.
- [104] Manzanares González, A. (2019). Detector de baches con Deep Learning. Trabajo Final de Grado, Universidad Pompeu Fabra, Barcelona.