

Detección automática de averías en el pavimento mediante visión por computadora y cómputo móvil

María de Jesús Galindo-López, Arturo Salazar-Segundo,
José Alberto Hernández-Aguilar

Universidad Autónoma del Estado de Morelos,
Facultad de Contaduría, Administración e Informática,
México

{mariadejesus, arturo.salazar, jose.hernandez}@uaem.mx

Resumen. En esta investigación se discute la detección automática de averías en el pavimento mediante visión por computadora. La detección de éstas se realiza combinando tecnologías computacionales y móviles, lo que puede ser crucial para el mantenimiento y seguridad de las vialidades de México, ya que son vistas como uno de los principales problemas en las ciudades. En esta investigación se discute las imperfecciones del pavimento previas a que se conviertan en baches, que pueden resultar peligrosos sobre todo de noche y en temporada de lluvias, ya que el agua estancada camufla la dimensión del bache, provocando daños que van de leves a graves no solo en los vehículos, sino que pueden llegar a causar accidentes con consecuencias desastrosas. Metodología: se recopilaron imágenes de imperfecciones en el asfalto utilizando teléfonos inteligentes, las cuales son procesadas para identificar sus características como forma, latitud y longitud, las cuales se procesan mediante visión por computadora utilizando Python, Scikit-learn, Opencv, y Tensorflow. En esta investigación se utilizó el modelo preentrenado resnet152 para identificar averías en el pavimento y se probó con el conjunto de datos propio. Los resultados preliminares indican que se tiene una precisión del 70 % para identificar averías en el pavimento.

Palabras clave: Detección de averías, pavimento, visión por computadora, cómputo móvil.

Automatic Pavement Fault Detection Using Computer Vision and Mobile Computing

Abstract. This research discusses the automatic detection of pavement defects using computer vision. Detecting these defects is done by combining computer and mobile technologies, which can be crucial for the maintenance and safety of Mexico's roads since they are seen as one of the main problems in cities. This research discusses the imperfections in the pavement before they become potholes, which can be dangerous, especially at night and during the rainy season, since stagnant water camouflages the size of the pothole, causing damage ranging from minor to severe not only to vehicles but can also cause accidents with disastrous consequences. Methodology: Images of asphalt imperfections were collected using smartphones, which are processed to identify their characteristics,

such as shape, latitude, and longitude, which are processed through computer vision using Python, Scikit-learn, Opencv, and Tensorflow. This research used the pre-trained model resnet152 to identify pavement damage and was tested with the in-house dataset. Preliminary results indicate a 70% accuracy in identifying pavement damage.

Keywords: Fault detection, pavement, computer vision, mobile computing.

1. Introducción

1.1. Antecedentes

Los baches se han convertido en uno de los problemas más comunes en las calles de México al grado de que la población los considera un foco de inseguridad vial, el 81 % de la población de 18 años y más, manifestó que los baches en calles y avenidas son uno de los problemas más importantes en su ciudad de acuerdo al Instituto Nacional de Estadística y Geografía en México [11]. Autores como Dhiman y Kettle (2019) [8], y Guo y Zhang (2022) [10], discuten el uso de diferentes algoritmos para la detección de anomalías basados en los datos de modelos pre-entrenados a fin de detectar grietas, escamas o baches. En Azar et al. (2019) [6] se discute la combinación de estas técnicas con el Internet de las cosas para ese mismo propósito.

La Inteligencia Artificial de las Cosas (del inglés Artificial Intelligence of Things - AIoT), se define como una extensión del internet hacia el ámbito físico, mientras que otros lo describen como un colectivo de sensores colocados en “cosas” y dentro de infraestructuras cibernéticas de acuerdo a Azar et al. (2019) [6]. En ella se unen las capacidades de IoT (Internet of Things), Big data e Inteligencia artificial, todas necesarias para la identificación de averías en el pavimento.

1.2. Problema de investigación

La principal línea de investigación de este proyecto se concentra en el desarrollo de un sistema que permita la detección y geolocalización de averías (anomalías) en el pavimento mediante visión por computadora y cómputo móvil.

1.3. Justificación

En Azar et al. (2020) [5] se describe una creciente demanda a nivel global de productos y servicios tanto en el campo Internet de las cosas como en el de Inteligencia Artificial (IA), tan solo para el año 2020 se esperaban inversiones de hasta 1.5 billones de dólares. Estas tecnologías hacen posible el desarrollo de sistemas que permiten la identificación de averías en el pavimento de manera muy precisa. Este problema no solo lo enfrentan países latinoamericanos como México sino también países como India, Turquía, Corea del Sur, Japón por mencionar algunos.



Fig. 1. Ejemplo de avería tipo piel de cocodrilo en asfalto, en esta figura se muestra un conjunto de grietas acumuladas en el pavimento que asemejan la piel de reptil.

1.4. Objetivo

Diseñar un sistema adaptable a automóviles que permita detectar averías en el pavimento de calles y ciudades mediante visión por computadora, el desarrollo de algoritmos inteligentes, y el uso de sensores cámaras y GPS (del inglés Global Position System - Sistema de Posicionamiento Global) en entornos de IoT mediante dispositivos móviles.

1.5. Hipótesis

Ho. Es posible la implementación de un sistema de detección de averías en el pavimento mediante el análisis de imágenes por visión por computadora y las tecnologías móviles actuales.

1.6. Contribución

La principal contribución de esta investigación es: la creación de un sistema que permita prevenir la formación de baches en el Estado de Morelos, mediante la detección de imperfecciones (averías) en la carpeta asfáltica mediante visión por computadora y cómputo móvil, así mismo se aporta una base de datos real de averías en el asfalto en el Estado de Morelos. Lo que proporciona información valiosa que permite la planificación del mantenimiento preventivo y correctivo, pudiendo reducir los costos de reparaciones tardías, contribuyendo con ello a la mejora de la seguridad vial.



Fig. 2. Detalle de lecturas de los sensores de giroscopio de un teléfono inteligente. En la pantalla se muestran las lecturas en un punto en particular, basado en [7].

1.7. Alcances y limitaciones

Se utilizará el modelo preentrenado `resnet152_rdd_19_best8140_infer` para identificar las averías en el pavimento mediante transferencia de aprendizaje (transfer learning). Para la etapa de prueba se utilizará un conjunto de datos propio, que consta de 75 imágenes de averías en el pavimento registradas con dispositivos móviles en el Estado de Morelos. Los principales desafíos técnicos son: la posición de la cámara, la calidad de la imagen, la velocidad del vehículo en el que va montada la cámara, y las condiciones meteorológicas en las que se prueba el desempeño del modelo.

1.8. Estructura del documento

En la primera sección se discute la problemática que se quiere resolver, el objetivo general, hipótesis y principal contribución. En la segunda sección se analiza el trabajo relacionado. En la tercera sección se presenta la metodología propuesta. En la cuarta sección se discuten los resultados obtenidos. En la quinta sección se presentan conclusiones y trabajos futuros. Finalmente, se listan las referencias utilizadas.

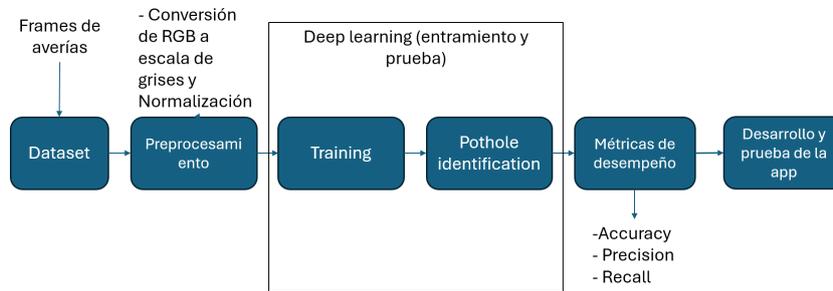


Fig. 3. Metodología propuesta (Fuente propia).

2. Trabajo relacionado

Según Manzanares-González (2019) [13] existen diferentes tipos de averías e imperfecciones en las carreteras y calles de México, éstas pueden ir desde simples grietas - ya sean transversales de corto, mediano y largo alcance-, las que ocurren por fatiga del asfalto también llamadas piel de cocodrilo, hasta hoyos también llamados baches. Los desperfectos mostrados en la (Fig. 1) son una clara señal de un futuro bache, que dependiendo del tamaño de la imperfección, el tamaño y la forma del bache podrían variar, aunque la profundidad dependerá de diversos factores como el clima, nivel de tránsito vehicular, tiempo de la avería, y la temporada del año, entre los más comunes.

2.1. Tipos de pavimentación

De acuerdo Sandstone Global (2024), en México existen diversos tipos de imperfecciones en las calles y carreteras. La pavimentación consiste en la colocación de capas de materiales en el suelo o nivel superior de la terracería, a la que posteriormente se le aplica el asfalto, losa, piedra, ladrillos, concreto, concreto hidráulico, por mencionar algunos, que conformará la superficie de rodamiento, los tipos de pavimento más utilizados son:

Pavimento flexible. Tiene como superficie de rodadura una capa de mezcla asfáltica comúnmente llamada asfalto. Las cargas de los vehículos hacia las capas inferiores se distribuyen mediante las características de fricción y cohesión de las partículas de los materiales, mientras que la carpeta asfáltica se pliega a pequeñas deformaciones de las capas inferiores sin que su estructura se rompa.

Pavimento semirrígido. La carpeta asfáltica se apoya en una base asfáltica o en una base estabilizada con cemento Portland. Este material es un tipo de cemento hidráulico compuesto principalmente de clinker -mezcla de roca caliza y arcilla- más yeso. Comúnmente se llama pavimento tipo adoquín.

Pavimento rígido. Tiene como superficie de rodadura una losa de concreto apoyada en capas de diversos materiales. Este tipo de pavimento no puede plegarse a las deformaciones de las capas inferiores sin que se presente una falla estructural. Este tipo de pavimento se caracteriza por tener en su capa superior concreto hidráulico, y es el que se empleó en el paso exprés de la ciudad de Cuernavaca Morelos. En este trabajo nos enfocaremos en el análisis de averías en pavimento flexible y pavimento rígido.

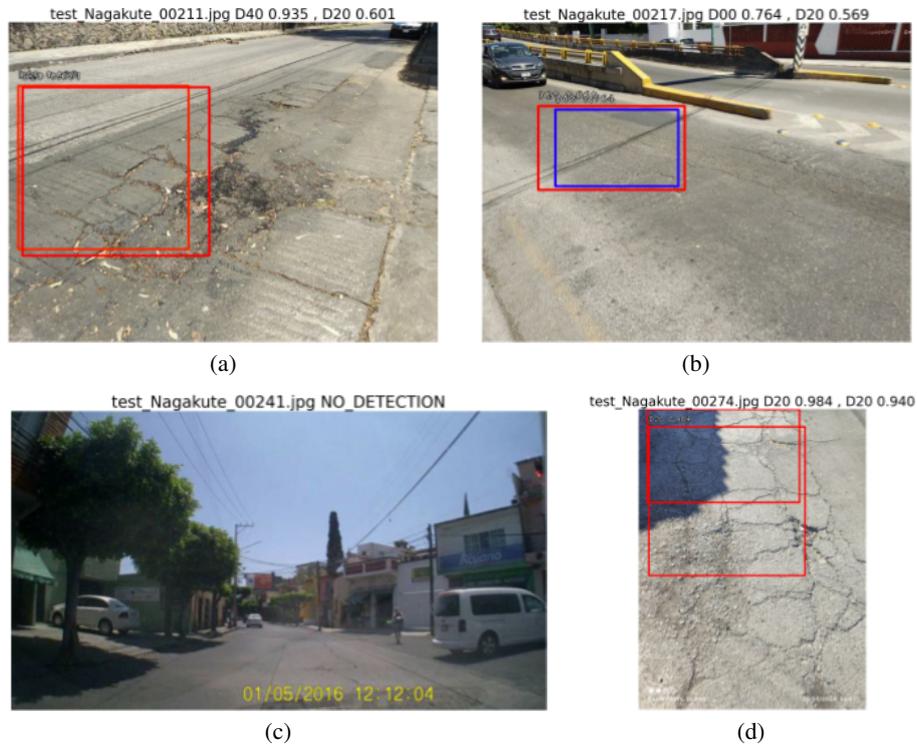


Fig. 4. Predicción de averías en diferentes escenarios.

2.2. Detección de averías en el pavimento

Existen proyectos que plantean diferentes enfoques para la detección de baches, algunos utilizan machine learning, otros deep learning, o una combinación de ellos, así como sensores GPS, giroscopios y acelerómetros para recopilar datos; otros utilizan bases de datos preexistentes de anomalías en pavimentos que posteriormente analizan y prueban con sus propios algoritmos, para verificar su funcionamiento en la identificación de averías con métricas como Accuracy, Precision o Recall.

Muchos de los datos que se utilizan en este tipo de estudios se recopilan con teléfonos inteligentes, también llamados smartphones, los cuales cuentan con muchos sensores, desde los más básicos como los que detectan luz ambiental, temperatura ambiental, o proximidad para apagar la pantalla en una llamada y evitar tocar botones de forma accidental, hasta los más complejos como los que detectan el rostro o voz en tiempo real, además incluyen cámaras de alta resolución y GPS útiles para la geolocalización de personas y cosas.

Para la Detección de anomalías en la carpeta asfáltica, Ramírez-Venegas (2017) [14] propuso el uso de sensores de teléfonos inteligentes y dispositivos de diagnóstico a bordo (del inglés OnBoard Diagnostic version II - OBDII), además utilizó redes neuronales artificiales (ANN). Usando 2 métodos de análisis, uno que procesa imágenes 2D y otro que procesa vídeos, se obtuvo una precisión en la detección de averías de aproximadamente el 90 %.

Tabla 1. Número de detecciones en algunas imágenes del conjunto de prueba.

Imagen	D00	D01	D10	D11	D20	D40	D43	D44	D30	N. detec.
00206					0,601	0,935				2
00210					0,999					1
00211		0,626			1,000					2
00212	0,764				0,569					2
00224										0
00236										0
00269	0,882	0,712								2
00272					0,984	0,940				2

Cabe señalar que el puerto OBDII se tiene hoy en día en el 100 % del parque vehicular del mundo. Un módulo OBDII es un dispositivo con un puerto simple que almacena datos en tiempo real de los autos, este módulo se usa con el ELM327, el cual cuenta con un módulo Bluetooth habilitado para emparejarse con un teléfono inteligente según Ashwini, Bhagwat y Sharma (2020) [4]. Todas las marcas ensambladoras de vehículos están obligadas desde el año 1996 a instalarlo en sus autos nuevos.

En Azar y Tapia (2018) [7] se presenta un trabajo que emplea una aplicación móvil, que además del acelerómetro de un teléfono inteligente utiliza el giroscopio para la identificación de baches. A partir de los datos recolectados se entrenan modelos con Máquinas de Soporte Vector - (del inglés Support Vector Machines - SVM) logrando una precisión superior al 90 %. En la Fig. 2 se muestran las lecturas obtenidas en un punto en particular mediante un teléfono móvil.

En Azar et al. (2019) [6] se discute la Inteligencia Artificial de las Cosas “móviles”, se menciona que no se necesita de un equipo de cómputo convencional o portátil más que los sensores conectados a un teléfono inteligente para recopilar los datos. Estos sensores hacen referencia a la cámara y al GPS, los cuales se activan al tomar una fotografía, en donde no solo se registra la imagen tomada con una alta resolución, sino también se registra el lugar donde fue tomada, incluyendo su latitud y longitud.

Los autores señalan que al tratarse de tecnologías móviles que al día de hoy tienen recursos limitados, herramientas como Tensorflow, YOLO (You Only Look Once) v5, y APIs (Applications) de alto nivel son esenciales en el desarrollo de una aplicación para la detección de las averías viales, las cuales deben ser optimizadas para su funcionamiento, haciendo además uso de lenguajes de programación como JAVA y Kotlin (Android), Objective C (IOS), C++ y Python que pueden proporcionar flexibilidad y distintos entornos de desarrollo.

En Manzanares-González (2019) [13] se recopilan datos de teléfonos inteligentes, y se utiliza Deep Learning (red neural profunda) para la detección de anomalías en el pavimento basados en modelos pre-entrenados a fin de detectar grietas, escamas o bien baches directamente. En este trabajo, se realizó una selección de la plataforma tecnológica, la cual permitió implementar una red neural en la que se entrenaron diferentes modelos.

Tabla 2. Resumen de la detección de averías.

Tipo	Número
Imágenes en las que sí se detectó averías	42
Imágenes en las que no se detectó averías	33
Total de imágenes procesadas	75

Con el mejor de estos modelos se creó una app para la detección de baches. En Escobar, Flores y Fernández (2023) [9] se discute la estimación de averías en el pavimento utilizando técnicas de procesamiento digital de imágenes, para lo cual se utilizaron técnicas de transformación de perspectiva, umbrales, y filtros, que permiten estimar el área afectada. Los resultados se evaluaron utilizando el método de intersección sobre unión (IOU); se obtuvieron valores de accuracy de 0.69 para piel de cocodrilo, 0.87 para baches, y 0.79 para grietas.

2.3. Bases de datos para la detección de baches

Con respecto a la detección de objetos basada en Deep Learning y aplicada a vehículos autónomos, en Arriola (2018) [1] se menciona que existen varias bases de datos de baches en distintas condiciones meteorológicas, pero algunos investigadores también se apoyan de imágenes propias, ya que el ángulo de visión de muchas imágenes no es el mismo que se lleva dentro de la cabina del vehículo. De acuerdo a Arriola (2018) [1], para entrenar una inteligencia artificial se necesita una base de datos robusta, en este trabajo la prueba consistió de 5874 imágenes las cuales tenían una resolución de 3680×2760 píxeles, que es una resolución aproximada a 4K.

Para acortar tiempos de entrenamiento es viable usar bases de datos ya hechas que se pueden encontrar en internet. De acuerdo a Arya et al. (2021) [2] una de las bases de datos más utilizada para el entrenamiento, prueba y evaluación de averías en el pavimento es el dataset RDD2020: An Image Dataset for Smartphone-based Road Damage Detection and Classification.

Este conjunto de datos contiene 6336 imágenes de caminos de la India, Japón y República Checa con más de 31000 ejemplos de daños en el piso. El conjunto de datos tiene cuatro categorías de daños: Roturas lineales (D00), roturas transversales (D10), roturas de piel de cocodrilo (D20), y baches (D40); y fue desarrollada para probar métodos basados en deep learning que permitan detectar y clasificar daños en el camino automáticamente. Estas imágenes se capturaron utilizando teléfonos inteligentes montados en el vehículo, lo que hace útil a los municipios y agencias de movilidad desarrollar métodos de monitoreo de bajo costo.

El reto global para la detección de daños en el camino (GRDDC'2020), fue organizado por la IEEE Big Data Cup en 2020, y utilizó el dataset RDD2020 para evaluar los modelos de detección de daños en el camino propuestos por los participantes. Una visión general del evento, puede ser consultada en Arya, Maeda y Sekimoto (2024) [3]. Así mismo, hay repositorios donde además de la base de datos, se encuentran los programas que pueden ser utilizados para identificar las averías en el pavimento y que pueden ser utilizados para fines de comparación, como por ejemplo el github de Khokhar (2023) [12].

3. Metodología

La metodología propuesta para esta investigación consta de 6 etapas véase la Fig. 3.

Base de datos. Para este proyecto se utilizaron 75 imágenes de averías en el pavimento con una resolución de 600 x 600 píxeles. Estas imágenes fueron tomadas por los autores en distintos puntos del Estado de Morelos.

Preprocesamiento. Las imágenes se convirtieron a escala de grises y se normalizaron para disminuir la complejidad de procesamiento.

Entrenamiento y prueba del modelo. Estas etapas se llevaron a cabo utilizando redes de convolución CNN - Convolutional Neural Network, para el entrenamiento se utilizó el modelo preentrenado `resnet152_rdd_19_best8140_infer` descrito en Manzanares-González (2019) [13], y para la prueba se utilizaron las 75 imágenes preprocesadas. El entrenamiento y la prueba del modelo se llevaron a cabo en una computadora de escritorio ADM Ryzen 5, modelo 2600 de 6 núcleos y 12 hilos a 3.40GHz de velocidad, 16GB de RAM DDR4 a 3200 MHz, una tarjeta gráfica AMD Radeon RX 460 de 4GB con interfaz PCIe 3.0x8.

Configuración del ambiente de trabajo. A continuación, se describen los pasos que se siguieron para configurar el ambiente de trabajo:

1. Se importan las librerías y módulos necesarios para el procesamiento de imágenes y detección de objetos Numpy, Keras, Pandas, Tensorflow, Matplotlib, Cv2 y módulos de keras retinanet para detección de objetos.
2. Se configura la sesión de Tensorflow definiendo la función `get_session()` para permitir el crecimiento de la memoria en las unidades gráficas de procesamiento GPUs.
3. Se define el modelo de Retinanet pre-entrenado asignado a la variable `model_trained` desde una ubicación específica usando el modelo `resnet152`, se carga usando la función `load_model()`.
4. Definimos un diccionario que mapea los índices para etiquetar las zonas por el modelo.
5. Se leen el conjunto de datos prueba `testset` desde un archivo `.CSV` (delimitado por comas), el cual contiene la lista de imágenes a procesar, este describe el ancho, altura y el nombre de las imágenes, para ello se utiliza la función `pd.read_csv()` de pandas.
6. Se realiza un ciclo `for` para la detección de cada imagen, el número de imágenes se establece en `'tail(n)'`.
7. Se carga la imagen usando la función `read_image_bgr()` desde una ruta específica y se añade el formato de la imagen `'format(img)'`, además de preparar el ambiente para dibujar el delimitador.
8. Se procesa la imagen con la función `'process_image()'` y se almacena una copia en `'draw'` para convertir la imagen en color a formato `'BGR'` a `'RGB'` con la función `'cv2.cvtColor()'`.

9. Se procesan las imágenes por la red neuronal de tensor flow “tf” pasando previamente por la función ‘preprocess_image()’, después se redimensiona la imagen con la función ‘resize_image()’ para tener un tamaño de 600x600 requerido por el modelo.
10. Se realiza la predicción de objetos en la imagen redimensionada y se lleva a cabo la corrección de cambios de la escala.
11. En caso de no encontrar nada, se imprime ‘NO_DETECTION’.
12. Iteramos cada predicción de objetos delimitando una caja, un puntaje de confianza y una etiqueta. Si el puntaje es mayor o igual a 0.5 se dibuja la caja delimitadora.
13. Se crea una visualización de cada imagen con las detecciones y la información de los objetos con ayuda de la biblioteca matplotlib; con la variable ‘draw’ mostramos las cajas delimitadoras, finalmente mostramos en pantalla el resultado de cada imagen procesada con ‘plt.show()’.

Métricas. Las métricas que se utilizaron para evaluar el desempeño del sistema están basadas en la matriz de confusión propuesta en Raschka y Mirjalili (2019) [15]: El accuracy (exactitud) es una métrica para evaluar modelos de clasificación. Informalmente es la fracción de predicciones que el modelo realizó correctamente. De manera formal, el accuracy tiene la siguiente definición [15]:

$$\text{Accuracy} = \frac{\text{No. de predicciones correctas}}{\text{No. total de predicciones}}. \quad (1)$$

Para clasificación binaria, el accuracy también se puede calcular en términos de positivos y negativos de la siguiente manera:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (2)$$

donde TP = Verdaderos positivos, TN = Verdaderos negativos, FP = Falsos positivos y FN = Falsos negativos. También se utilizarán las métricas de Precisión (Precision) y Recall (Sensibilidad). La precisión responde a la pregunta ¿qué proporción de identificaciones positivas fue correcta [15].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (3)$$

El recall responde a la pregunta ¿Qué proporción de positivos reales se identificó en forma correcta? [15]:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (4)$$

Desarrollo y prueba de la App. El sistema entrenado con el modelo de red convolucional se integra en una app en Android Studio mediante el desarrollo por prototipos. El teléfono inteligente se colocará de manera horizontal en el tablero o sobre el volante, la app activará la cámara del dispositivo, y entrará en modo búsqueda de baches, y conforme vaya avanzando, el vehículo identificará los baches

Tabla 3. Métricas de desempeño.

Tipo	Número
Total de anomalías detectadas	66
Verdaderos positivos (TP)	41
Falsos positivos (FP)	18
Detecciones negativas (FN)	7
Accuracy	0,62
Precision	0,70
Recall	0,85

que vayan apareciendo en el horizonte en color verde, marcando su posición GPS (latitud y longitud) y señalando el porcentaje de precisión de la detección correspondiente. Si aparecen varias averías las mostrará también con sus respectivas métricas y geoposición.

4. Resultados y discusión

Las 75 imágenes de prueba fueron tomadas en diferentes ángulos y alturas. Tomando en cuenta esto, con las imágenes que se obtuvieron con cámaras deportivas tipo GoPro de resolución 720p de calidad baja, el modelo no pudo detectar anomalías, salvo en un caso en el cual se apuntó a un poste -véase la imagen 00233-. A partir de la imagen 00256 las fotografías se tomaron en un ángulo de 90°, es decir, la cámara del teléfono apuntando directamente al suelo, considerando circunstancias como la sombra de cables, postes de luz y teléfono.

Un pequeño conjunto de imágenes (siete) fueron tomadas estando de pie y apuntando la cámara del teléfono en un ángulo de 180°, es decir viendo al horizonte, para poder visualizar banquetas, automóviles, sombras y estructuras de casas, esto con el fin de poder diferenciar entre anomalías y sombras.

A continuación en la Figura 4, se muestran los resultados de procesar diferentes imágenes; en la Figura 4a. se muestra una predicción acertada en la sección D40 con una precisión de 0.935 y en la sección D20 con una precisión de 0.601. En la Figura 4b. se muestra un falso positivo, en el que se confundió una sombra de un cable con una avería. En la figura 4c. se muestra un verdadero negativo, es decir existe una avería pero no se detectó; y en la figura 4d. se muestran dos verdaderos positivos en la sección D20, con precisiones de 0.984 y 0.940 respectivamente.

Los resultados completos de la detección en las imágenes tomadas y procesadas con la metodología propuesta pueden ser consultados en Salazar-Segundo (2024). En la Tabla 1 se muestra un subconjunto de los resultados obtenidos. La columna uno indica el número de imagen; las columnas 2 a la 10, el tipo de anomalía detectada de acuerdo a Manzanares-González (2019) [13]; y la última columna indica el número de detecciones. En la tabla 2 se muestra el resumen de la detección de averías, de las 75 imágenes de prueba en 42 sí detectaron averías, y en 33 no se detectaron averías.

De acuerdo a la tabla 3, se alcanzó un Accuracy (exactitud) de 0.62, una Precision (precisión) de 0.70, y un Recall (sensibilidad) de 0.85, lo cual indica que la predicción es buena, pero se requiere de una cantidad mayor de imágenes para mejorar el desempeño de la propuesta. Estos resultados están por debajo de los obtenidos por Manzanares-González (2019) [13] quien obtuvo un porcentaje de acierto de 0.8945, y un porcentaje de fallo de 0.1055. También están por debajo de los obtenidos por Escobar-Arenas et al. (2023) [9] en donde se obtuvo un accuracy promedio de 0.78.

5. Conclusiones y trabajos futuros

Las agencias de transporte y obras públicas pueden mejorar la condición y operación de sus redes de calles y carreteras implementando un sistema de gestión de mantenimiento de pavimentos, que utilice recopilación de datos (imágenes, latitud y longitud) basada en teléfonos inteligentes y predicción mediante visión por computadora, que permita el apoyo a la toma de decisiones. Dado los resultados obtenidos, se concluye que la hipótesis Ho. Es posible la implementación de un sistema de detección de averías en el pavimento mediante el análisis de imágenes por visión por computadora y las tecnologías móviles actuales, se cumple.

Si bien la calidad de la cámara juega un papel muy importante a la hora de grabar vídeo, existen variables que pueden llegar a generar falsos positivos o que simplemente no se muestren datos al momento de procesar las imágenes, es el caso de imágenes tomadas con reflejos del parabrisas provocados por el sol o la suciedad en el mismo, lo que puede confundirse con una anomalía, situaciones que pueden impedir una lectura correcta de las condiciones del pavimento. Además de que las cámaras de los celulares inteligentes no cuentan con ángulos de visión pronunciados (ojo de pez), y que al momento de procesar las imágenes podrían no ser los adecuados.

Para futuras investigaciones, consideramos sujetar la cámara al cofre o a la defensa del vehículo para tener una mejor visión de la vialidad, sobre todo en horarios donde el sol no proyecta sombras en las averías (medio día), al igual que en zonas de mucha sombra ya sea provocada por postes de luz, árboles o cables. También se consideraran calles o carreteras de concreto o concreto hidráulico, ya que los resultados pueden variar debido al estriado que se les realiza para el desvío de la lluvia, provocando que se obtengan resultados erróneos.

Se quedan también como trabajo futuro dos aspectos importantes, el primero consiste en generar una base de datos propia que contenga tres clases balanceadas de averías: piel de cocodrilo, bache y grieta, que permitan mejorar las métricas de desempeño de la metodología propuesta. El segundo consiste en el desarrollo de la aplicación móvil (última etapa de la metodología propuesta).

Referencias

1. Arriola-Oregui, I.: Detección de objetos basada en deep learning y aplicada a vehículos autónomos. Master's Thesis, Universidad del País Vasco (2018)
2. Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Sekimoto, Y.: RDD2020: An annotated image dataset for automatic road damage detection using deep learning. Data in Brief, vol. 36, pp. 107133 (2021) doi: 10.1016/j.dib.2021.107133

3. Arya, D., Maeda, H., Sekimoto, Y.: From global challenges to local solutions: A review of cross-country collaborations and winning strategies in road damage detection. *Advanced Engineering Informatics*, vol. 60, pp. 102388 (2024) doi: 10.1016/j.aei.2024.102388
4. Ashwini, K., Bhagwat, G., Sharma, T., Pagala, P. S.: Trigger-based pothole detection using smartphone and OBD-II. In: *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies*, vol. 6, pp. 1–6 (2020) doi: 10.1109/conecct50063.2020.9198602
5. Azar, M. A., García, J. L., Bernal, S., Aleman, L., Tolaba, M.: Inteligencia artificial aplicada a IoT. In: *XXII Workshop de Investigadores en Ciencias de la Computación*, pp. 54–59 (2020)
6. Azar, M. A., Tapia, M., García, J. L., Pérez, A. J. M.: Inteligencia artificial de las cosas. In: *XXI Workshop de Investigadores en Ciencias de la Computación* (2019)
7. Azar, M. A., Tapia, M. A.: Detección de averías viales mediante IoMT aplicada a smart cities. In: *XXIV Congreso Argentino de Ciencias de la Computación*, pp. 1133–1141 (2018)
8. Dhiman, A., Klette, R.: Pothole detection using computer vision and learning. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3536–3550 (2019)
9. Flórez-Pareja, L. D., Escobar-Arenas, J. P., Fernandez-Mc-Cann, D. S.: Estimación de irregularidades en pavimentos mediante técnicas de procesamiento digital de imágenes. *Revista Politécnica*, vol. 19, no. 37, pp. 20–28 (2023) doi: 10.33571/rpolitec.v19n37a2
10. Guo, G., Zhang, Z.: Road damage detection algorithm for improved YOLOv5. *Scientific Reports*, vol. 12, no. 1 (2022) doi: 10.1038/s41598-022-19674-8
11. Instituto Nacional de Estadística y Geografía: Encuesta nacional de seguridad pública urbana (2024) www.inegi.org.mx/programas/ensu/
12. Khokhar, N.: Pothole detection using python and deep learning (2023) github.com/noorkhokhar99/Pothole-Detection-Pothole-Detection-using-python-and-deep-learning
13. Manzanares-González, A.: Detector de baches con deep learning. Master's Thesis, Universitat Pompeu Fabra Barcelona (2019)
14. Ramírez Venegas, C. A.: Sistema computacional para detectar anomalías en vías urbanas con base en vibraciones mecánicas. Master's Thesis, Centro de Investigación en Computación, Instituto Politécnico Nacional (2015)
15. Raschka, S., Mirjalili, V.: *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2*. Packt Publishing (2017)