

UNIVERSIDAD NACIONAL DE CAAGUAZÚ
FACULTAD DE CIENCIAS Y TECNOLOGÍAS
CARRERA DE INGENIERÍA CIVIL



PROYECTO FINAL DE GRADO

**Segmentación de grietas superficiales en
pavimento asfáltico utilizando técnicas de visión
artificial**

Luz Rashell Soto Olguin

**TUTOR: Ing. M.Sc. Fredy Gabriel Ramírez
Villanueva**

**CORONEL OVIEDO, MES DE DICIEMBRE DE
2024**



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

DERECHO DE AUTOR

Quien suscribe, Luz Rashell Soto Olguin, autora del trabajo de investigación titulado **“Segmentación de grietas superficiales en pavimento asfáltico utilizando técnicas de visión artificial”**, declara que voluntariamente cede a título gratuito en forma pura y simple ilimitada e irrevocablemente a favor de la Facultad de Ciencias y Tecnologías – UNCA, el derecho de autor de contenido patrimonial, que le corresponde sobre el trabajo de referencia. Conforme a lo anteriormente expresado, esta sesión le otorga a la FCyT la Facultad de comunicar la obra divulgarla, publicarla y reproducirla en soportes analógicos o digitales en la oportunidad que así lo estime conveniente. La FCyT deberá indicar qué autoría o creación del trabajo corresponde a mi persona y hará referencia al autor y a las personas que hayan colaborado en la realización del presente trabajo de investigación.

En la ciudad de Coronel Oviedo a los , del mes de diciembre del 2024

.....

Firma



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

PÁGINA DE APROBACIÓN

Trabajo de fin de grado para la obtención del Título de Ingeniero Civil, aprobado en representación de la Facultad Ciencias y Tecnología de la Universidad Nacional de Caaguazú, por el Tribunal Examinador constituido por los siguientes profesores y con la siguiente nota final:

CALIFICACIÓN FINAL: _____

ACTA N°: _____

FECHA : _____

Prof. Ing.

Prof. Ing.

Prof. Ing.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

DEDICATORIA

Dedico este trabajo a mi familia, cuyo amor y apoyo incondicional me han brindado la fuerza y la motivación necesarias para alcanzar mis metas. A mis amigos, quienes han estado a mi lado en cada paso del camino, y a todos aquellos que me han inspirado a seguir adelante en mi búsqueda de conocimiento y excelencia.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

AGRADECIMIENTOS

Quisiera expresar mi más sincero agradecimiento a mi tutor y a todos los profesores que me han guiado y apoyado durante este proceso. Su invaluable orientación y conocimientos han sido fundamentales en el desarrollo de este proyecto. Agradezco también a mis compañeros de clase por compartir experiencias y conocimientos, lo que ha enriquecido mi aprendizaje. Por último, quiero agradecer a todas las personas que, de alguna manera, contribuyeron a la realización de este trabajo, ya sea mediante su apoyo moral, técnico o motivacional.



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

RESUMEN

Este estudio presenta el desarrollo de un algoritmo basado en redes neuronales convolucionales para la segmentación de grietas en pavimentos asfálticos, utilizando una base de datos propia y evaluando la precisión de los resultados del mismo después de implementar las mejoras identificadas durante la evaluación del desempeño del algoritmo para el incremento de la precisión y la fiabilidad en la detección de grietas en el pavimento.

Se capturaron 837 imágenes en la ciudad de Coronel Oviedo, Paraguay, de las cuales 505 contenían grietas. Las imágenes fueron preprocesadas y divididas en secciones de 512x512 píxeles, generándose máscaras manuales para la detección de grietas. La arquitectura U-Net fue seleccionada para el entrenamiento de la red neuronal, con imágenes de 256x256 píxeles, utilizando 500 épocas y un tamaño de lote de 8.

El modelo fue evaluado mediante métricas de segmentación, obteniendo un F1 Score de 0.9956 y un valor de intersección sobre la unión (IoU) de 0.9913, lo que indica una alta precisión en la detección de grietas. Además, se implementó un procedimiento para cuantificar el ancho de las grietas, clasificándolas en dos categorías: menores o iguales a 3 mm que no presentan mayores riesgos para el pavimento, y mayores a 3 mm los cuales deben ser tratados mediante sellantes indicados por el Manual de carreteras del Paraguay. A pesar de la presencia de algunos falsos positivos y negativos, el modelo mostró un rendimiento superior en comparación con estudios previos.

Palabras claves:

Ingeniería estructural, mecánica y materiales

Vulnerabilidad y riesgos naturales

Representaciones gráficas

Tecnologías de la Información y la Comunicación (TIC)

Software

Ciencia de los Sistemas Complejos

Ciencia y Tecnología para el Desarrollo Sostenible

ODS 9: Industria, Innovación e Infraestructura

ODS 11: Ciudades y Comunidades Sostenibles

ODS 12: Producción y Consumo Responsables

ODS 13: Acción por el Clima



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

ABSTRACT

This study presents the development of an algorithm based on convolutional neural networks for the segmentation of cracks in asphalt pavements, utilizing a proprietary database and evaluating the accuracy of the results after implementing improvements identified during the algorithm's performance assessment. A total of 837 images were captured in the city of Coronel Oviedo, Paraguay, of which 495 contained cracks. The images were preprocessed and divided into sections of 512x512 pixels, with manual masks generated for crack detection. The U-Net architecture was selected for training the neural network, using images of 256x256 pixels, 500 epochs, and a batch size of 8.

The model was evaluated using segmentation metrics, achieving an F1 Score of 0.9956 and an Intersection over Union (IoU) value of 0.9913, indicating high precision in crack detection. Additionally, a procedure was implemented to quantify the width of cracks, classifying them into two categories: those less than or equal to 3 mm, which pose no significant risk to the pavement, and those greater than 3 mm, which should be treated with sealants as indicated by the Paraguay Road Manual. Despite the presence of some false positives and negatives, the model demonstrated superior performance compared to previous studies.

Keywords:

Structural, Mechanical, and Material Engineering

Vulnerability and Natural Risks

Graphical Representations

Information and Communication Technologies (ICT)

Software

Complex Systems Science

Science and Technology for Sustainable Development

SDG 9: Industry, Innovation, and Infrastructure

SDG 11: Sustainable Cities and Communities

SDG 12: Responsible Production and Consumption

SDG 13: Climate Action



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.

VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

ÍNDICE

DERECHO DE AUTOR.....	1
INTRODUCCIÓN.....	1
Antecedentes.....	2
1. Introducción a las Redes Neuronales Convolucionales (CNNs)	5
1.1. Fundamentos de las Redes Neuronales Artificiales.....	5
1.3. Concepto de Redes Neuronales Convolucionales (CNN).....	7
1.5. Funciones de pérdida.....	10
1.5.1. Back propagation	10
2. Procesamiento de imágenes	11
3. Métodos Tradicionales de Detección de Grietas utilizados en Paraguay	12
4. Definición de términos básicos	12
Objetivos.....	15
Objetivo General	15
Objetivos Específicos	15
METODOLOGÍA.....	16
1. Recopilación de Datos en la Zona de Coronel Oviedo, Paraguay.....	16
2. Entrenamiento de la Red Neuronal Convolutiva	18
RESULTADOS Y ANÁLISIS	21
CONCLUSIONES Y RECOMENDACIONES	24
Recomendaciones.....	24
ANEXOS.....	1
Anexo A: Código del Algoritmo de Segmentación	1
Anexo B: Código para el Procesamiento de Imágenes y Predicciones.....	7
Anexo C: Código para medir las grietas	10



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

ÍNDICE DE FIGURAS

Figura 1: Curva de precisión de validación y de entrenamiento. U-Net-Based CNN Architecture for Road Crack Segmentation en 2023 3

Figura 2: Arquitectura de un perceptrón simple. Fuente: Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021) 5

Figura 3: Estructura de un Perceptrón multicapa. Fuente: Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021) 7

Figura 4: Operación de convolución con un filtro 3x3 aplicado a una imagen. Fuente: Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021) 8

Figura 5: Aplicación de UpSampling con filtro de dimensión 2x2 y stride igual a 2. Fuente: Carcass image segmentation using CNN-based methods (2020) 9

Figura 6: Ejemplo de usar la configuración de la probabilidad $p=0,5$. Fuente: Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021) 9

Figura 7: U-Net. Fuente: Arquitectura CNN basada en U-Net para segmentación de grietas en carreteras (2015) 11

Figura 8: Gráfico de precisión vs épocas. Fuente: Elaboración propia 21

Figura 9: Resultados obtenidos en las imágenes de prueba. Fuente: Elaboración propia 22

Figura 10: Grietas detectadas por el programa. Rojo: grietas mayores a 3mm. Verde: Grietas menores o iguales a 3mm. Tamaño mínimo de grieta detectada: 1,85 mm. 23



MISIÓN: Formar profesionales excelentes con conocimientos científicos y tecnológicos, competentes, con sentidos crítico, ético y responsabilidad Social.
VISIÓN: Ser una Facultad líder, con excelencia en la formación de profesionales que contribuya al desarrollo del País.

ÍNDICE DE TABLAS

Tabla 1: Comparación de métricas entre el modelo propuesto y el modelo de Di Benedetto et al. (2023).	21
--	----

INTRODUCCIÓN

Según el plan de desarrollo 2030 lanzado por el Banco Interamericano de Desarrollo (BID) la red vial de Paraguay alcanza los 60.000 km. [1], de los cuales; de acuerdo a los datos más recientes proporcionados por el Ministerio de Obras Públicas y Comunicaciones (MOPC) 10.613,4 km corresponden a pavimentos lo que equivale a un 17,7% [2], lo que significa que el 82,3% de los caminos del país puede tener dificultades para el tránsito debido al deterioro, lluvias, etc. En comparación con nuestros vecinos más cercanos como Argentina; que cuenta con 24,9% [3] de sus caminos pavimentados, o Brasil que tiene 12% [4] de sus carreteras pavimentadas, se puede ver que Paraguay está por buen camino en el sentido de inversión pública a la red vial.

Pero; sobre todo, es de común conocimiento para los habitantes del país que muchos de estos pavimentos; principalmente en zonas más pobladas, estén en mal estado con un continuo desgaste que, dejando por un lado la incomodidad que provocan, estos terminan siendo un peligro a la vida de los usuarios.

En términos de ingeniería civil es bien sabido que las fallas detectadas a temprana edad no solo son más sencillas de resolver a nivel técnico, sino conllevan menos costo y tiempo. De ahí surge la necesidad de detectar fallas del pavimento asfáltico desde los primeros síntomas las cuales se presentan en forma de grietas. Por esta razón, llevar a cabo evaluaciones de condición basada en el monitoreo de salud estructural (conocido como SHM por sus siglas en inglés, Structural Health Monitoring) [5, 6] resulta esencial para seguir de cerca el estado operativo, evaluar la condición y detectar daños en diversas estructuras. Sin embargo, en la realidad actual del país, las fallas estructurales de los pavimentos, conocidos comúnmente como baches, no se solucionan ni pasados meses, a veces años, debido a su costo técnico y económico.

Conforme a la regulación de la Asociación Estadounidense de Funcionarios de Transporte y Carreteras Estatales (AASHTO), se consideran como criterios de examen para elementos estructurales tanto el ancho como la longitud de las grietas, junto con la extensión de la región de desprendimiento. Existen pruebas no destructivas y muy complejas para evaluar la estos criterios (por ej., imágenes infrarrojas y térmicas, pruebas ultrasónicas, pruebas con láser y pruebas radiográficas) [7, 8, 9] pero la inspección visual manual sigue siendo ampliamente utilizada en la práctica para hacer dichas evaluaciones, lo que a menudo implica el envío de inspectores calificados al sitio para identificar grietas con marcas físicas, medir el ancho y largo de la grieta

con calibres manuales y fotografiar los defectos [10, 11, 12] que a su vez produce resultados subjetivos y/o ineficientes, sin mencionar los desafíos de tiempo y costo asociados.

En este sentido, la inteligencia artificial (IA) ha cobrado importancia en los últimos años, refiriéndose al desarrollo de sistemas informáticos capaces de percibir y procesar información. Esta técnica está ganando interés en varios campos, incluida la toma de decisiones, el reconocimiento de voz y los procedimientos visuales para identificar defectos superficiales estructurales como grietas [13] ya que solucionaría todos los problemas asociados a la inspección visual o mediante el uso de herramientas de alto costo. Específicamente en esta investigación se propone la utilización de U-Net que es un tipo de red neuronal perteneciente a las redes totalmente convolucionales (FCN, por sus siglas en inglés) utilizadas comúnmente en tareas de segmentación de imágenes y procesamiento de imágenes médicas [14, 15].

Este estudio pretende hacer tres contribuciones para abordar los problemas de detección de daños en las carreteras. Primero, crear una base de datos de entrenamiento que comprende 505 imágenes de daños en carreteras capturadas con un teléfono inteligente instalado en un atril calibrado para sacar las fotos en una determinada altura a modo de estandarizar el proceso. Luego, utilizar la red neuronal convolucional para entrenar el modelo de detección de daños con una configuración propia y comparar la precisión y la velocidad de ejecución con otros estudios similares.

Antecedentes

Di Benedetto et al. [16] en su investigación, utilizaron un modelo U-Net modificado con un codificador ResNet50 preentrenado para segmentar grietas en pavimentos, comparando su rendimiento con el algoritmo U-Net tradicional. El conjunto de datos Crack500 se amplió mediante técnicas de aumento de datos, alcanzando 12,000 imágenes para entrenamiento, 2,400 para validación y 9,600 para pruebas. Los resultados mostraron una alta precisión en la segmentación de las grietas, tanto en forma como en ancho, con una brecha mínima de 0,006 entre las curvas de entrenamiento y validación como se muestra en la Figura 1.

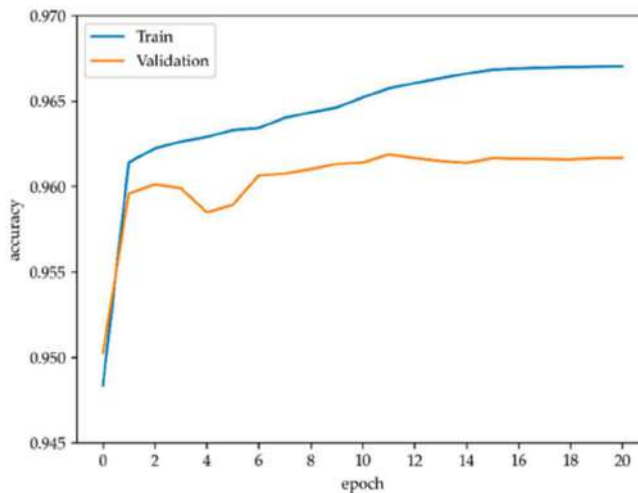


Figura 1: Curva de precisión de validación y de entrenamiento. U-Net-Based CNN Architecture for Road Crack Segmentation en 2023

Sin embargo, en dicha investigación se identificaron varios desafíos, como la necesidad de un extenso etiquetado manual de las imágenes de entrenamiento, lo que puede ser un proceso largo y subjetivo. Además, el rendimiento del modelo disminuye en imágenes con ruido, sombras o áreas sobreexpuestas, aunque en pruebas en condiciones reales los resultados fueron prometedores. Para mejorar el rendimiento futuro, los autores recomendaron optimizar el modelo y probarlo en otros conjuntos de datos, incluyendo grietas en estructuras como edificios y puentes, para aumentar su aplicabilidad y precisión en diversas condiciones.

En 2021 Tello-Cifuentes et al. [17] evaluaron el deterioro de las vías en Colombia por medio de técnicas de fotogrametría terrestre y redes neuronales dividiendo la investigación en seis etapas:

- i. Captura de las imágenes
- ii. Preprocesamiento de las imágenes
- iii. Segmentación mediante técnicas de detección de bordes
- iv. Extracción de las características
- v. Clasificación utilizando redes neuronales
- vi. Evaluación del área de afectación del deterioro.

Utilizando imágenes reales de pavimento con tres tipos de deterioro: grieta longitudinal, piel de cocodrilo y bache. Como clasificador se utilizó una red neuronal multicapa con configuración (12 12 3), entrenada utilizando el algoritmo Levenberg Marquardt de retropropagación; obteniéndose así una exactitud del 96 % en el clasificador, una sensibilidad de 93.33 % y una índice kappa de 0.936.

En México, en 2023 A. Roman et al. [18] presentaron un artículo que se enfoca en la detección y segmentación automática de agrietamientos y baches en pavimentos en situaciones no controladas por medio de un tipo de red neuronal conocida como redes transformadoras llamada Segformer, que se encargó de extraer características de las imágenes. Además, se creó un conjunto de datos compuesto por 245 imágenes, en el cual se amplió a 2052 imágenes mediante técnicas de procesamiento digital de imágenes y aumento de datos para mejorar la precisión del modelo. En el conjunto de datos se contemplaron distintos ambientes obtenidos en puntos situados en la ciudad de Culiacán, México, considerando cambios de iluminación y sombras, lo que permitió evaluar la robustez del modelo obtenido en condiciones similares a las de la vida real donde el ruido ambiental está presente. El modelo obtuvo métricas de Precisión de 82.35 %, F1 Score de 88.89 % y Recall de 96.55% en la detección de agrietamientos y baches en diferentes condiciones ambientales.

En 2023 Uribe S. Et al [19] presentaron un modelo destinado a la segmentación de cálculos renales en imágenes de Tomografía computarizada. Empleando la arquitectura U-Net; generalmente utilizada en la segmentación semántica de imágenes médicas, alcanzaron un índice de similitud del 64% una precisión del 99.87% y una sensibilidad del 88%. Los resultados obtenidos en dicho experimento demostraron que los enfoques basados en U-Net producen un mayor índice de similitud a partir del uso de aumento de datos lo que los hace una herramienta relevante en el análisis médico.

En 2021 Casilla, A. y Abad, W. [20] mostraron que es posible la detección automática del daño en estructuras de concreto reforzado usando redes neuronales. El sistema obtuvo un 98% de precisión para Unet y 74.85% de precisión para Yolov3. Además, el sistema de implementación unió efectivamente las funcionalidades de cada modelo, logrando que tenga un uso intuitivo para todos los posibles usuarios del sistema. También lograron identificar distintos tipos de patologías, que pueden ser clasificadas según distintos criterios. Sin embargo, dicha investigación encontró las siguientes limitaciones:

Algunas patologías conllevan mucha incertidumbre en su predicción y en estos escenarios, los modelos propuestos no podrían ser efectivos. Por ello, recomendaron evitar estas condiciones o estudiar una mejora para que las predicciones mejoren incluso en condiciones no óptimas.

Evidenciaron también que los requerimientos de hardware son muy exigentes. Por ello, recomendaron tener un gabinete que pueda llevar el entrenamiento en un tiempo relativamente

corto y de no ser el caso, usar el entorno COLAB que brinda acceso remoto a algunos servidores de Google con propósitos académicos.

1. Introducción a las Redes Neuronales Convolucionales (CNNs)

1.1. Fundamentos de las Redes Neuronales Artificiales

Las redes neuronales artificiales, también conocidas como Artificial Neural Networks (RNA), toman inspiración de las células nerviosas presentes en organismos vivos. En esencia, una neurona artificial cumple la función de recibir datos provenientes de otras neuronas o receptores, realizar un procesamiento de esta información para generar un código de activación interno, y posteriormente transmitir nuevos impulsos a las neuronas subsecuentes o a otras células. [21]

Debido a que el aprendizaje profundo es una combinación de técnicas de las RNA, su comprensión requiere el análisis previo de la arquitectura y sus aplicaciones. [22]

1.1. El perceptrón

Se podría decir que el perceptrón es el nombre que se le da a lo que, en un organismo, llamaríamos neurona, y en redes neuronales artificiales, neurona artificial. El principio de su funcionamiento consiste en sumar las entradas en base a unos pesos de acuerdo a los datos de entrada, el resultado de este proceso ingresa a la denominada función de activación el cual produce un valor final. [22]

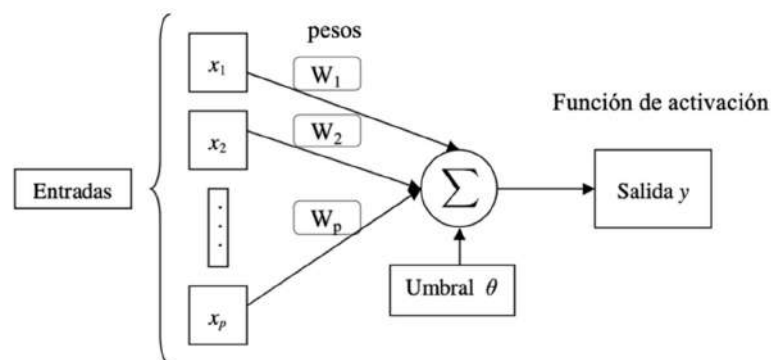


Figura 2: Arquitectura de un perceptrón simple. Fuente: *Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021)*

La arquitectura de red se basa en una estructura monocapa, en la que las “células” de entradas son tantas como fuese necesario, a su vez las “células” de salida son tantas como fuesen necesarias. Todas las “células” de entrada tienen conexiones con todas las “células” de salida, estas conexiones

son las que determinan las superficies de discriminación de todo el sistema. [23]

Para determinar la función de salida, primero se calcula la activación de la célula de salida sumando los pesos ponderados de todas las entradas [22]:

$$\hat{y} = \sum_{i=1}^n w_i x_i - \theta_i$$

Donde:

w_i : Peso

x_i : entrada

θ_i : valor denominado umbral

n : número de entradas

\hat{y} : función de activación

Para producir la salida final, se aplica una función de salida al nivel de activación de la célula. En el caso de los perceptrones, la función de salida es una función escalón que depende de un umbral específico [24]:

$$y = F\left(\sum_{i=1}^n w_i x_i - \theta_i\right)$$

La función de activación devuelve un valor de salida generado por las capas que le anteponen y la cantidad de estas depende de la tarea que va a realizar la neurona. Esta función puede ser continua (pequeños cambios en la entrada producen pequeños cambios en la salida) o diferencial (su derivada existe en cada punto del dominio) [22]

1.2. Capas de una red neuronal

Para solucionar ciertas limitaciones de un perceptrón simple se introduce el concepto de perceptrón multicapas o *Multi Layer Perceptron* (MLP), en inglés, que consiste en la creación de capas ocultas. El mismo es entrenado por medio de retropropagación de errores BP (*Back propagation*). [22]

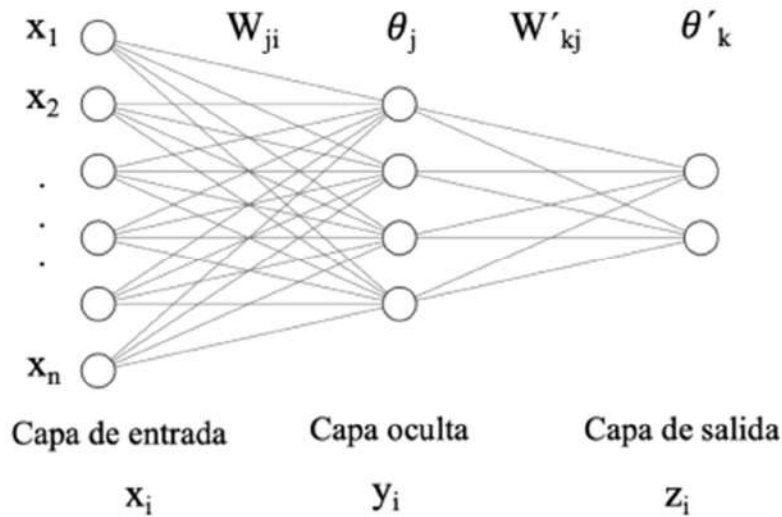


Figura 3: Estructura de un Perceptrón multicapa. Fuente: *Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021)*

Como describe L. Barba [22] una red neuronal artificial se compone de la siguiente topología:

- Capa de entrada: es el conjunto de datos de entrenamiento de la red neuronal.
- Capa oculta: son las neuronas entre la capa de entrada y de salida. Dependiendo de la topología el entrenamiento puede tardar más o menos tiempo.
- Capa de salida: Presenta la salida a través de un conjunto de vectores o un valor.

1.3. Concepto de Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales son un medio; basado en aprendizaje profundo, para la detección y segmentación de objetos, identificación de patrones o tendencia de datos en imágenes. El aprendizaje profundo es un tipo de aprendizaje automático en el que un modelo aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. El término “profundo” se refiere a la cantidad de capas de la red: cuantas más capas, más profunda es la red. Una red neuronal profunda combina múltiples capas de procesamiento no lineal, utilizando elementos simples que operan en paralelo e inspirados en los sistemas nerviosos biológicos. Consta de una capa de entrada, varias capas ocultas y una capa de salida. Las capas están interconectadas a través de nodos o neuronas, y cada capa oculta utiliza la salida de la capa anterior como entrada [25].

1.4. Convolución y agrupación

La operación de convolución consiste en aplicar un filtro o *Kernel* sobre una imagen. Este filtro es una matriz de coeficientes que asigna valores a los píxeles vecinos del píxel base y que va recorriendo la imagen desde la esquina superior izquierda hasta la esquina inferior derecha. Una vez realizada la operación de convolución, el valor de salida se almacena en una matriz llamada mapa de características (*feature map*) y la distancia que se mueve el filtro en la imagen se conoce como *stride*. [22]

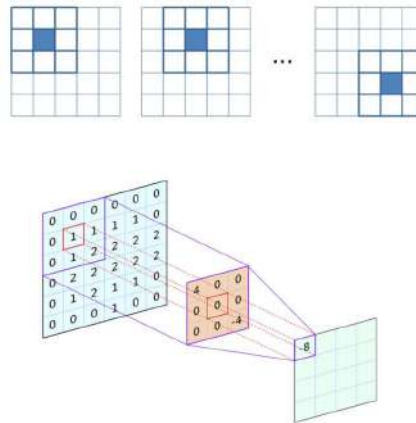


Figura 4: Operación de convolución con un filtro 3x3 aplicado a una imagen. Fuente: *Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021)*

Se debe tener en cuenta que si la imagen original es de $n \times n$ y a la misma se le aplica un filtro de tamaño $f \times f$, el resultado de la operación es de tamaño $(n - f + 1) \times (n - f + 1)$, lo que implica que en cada operación de convolución se pierde información debido a que en el borde de la imagen no se utiliza la operación; para evitar eso se utiliza una técnica llamada *padding*, que deja la dimensión de entrada y de salida como $(n + 2p) \times (n + 2p)$ y $(n + 2p - f + 1) \times (n + 2p - f + 1)$ respectivamente. [22]

La capa *pooling* o de reducción, disminuye el tamaño de representación de la capa anterior guardando solo las características más importantes a través de operaciones como el valor máximo (*Max pooling*) o el promedio (*Average pooling*) o el muestreo ascendente (*UpSampling*), que aumenta el tamaño del mapa de características en un factor de 2 en cada dimensión.

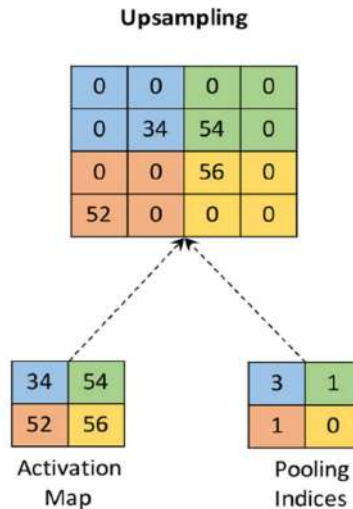


Figura 5: Aplicación de UpSampling con filtro de dimensión 2x2 y stride igual a 2. Fuente: *Carcass image segmentation using CNN-based methods (2020)*

La capa *Fully Connected -FC* o completamente conectada, en español; es una capa de convolución con un filtro de 1x1 que está completamente conectadas a las capas anteriores y se ubican; por lo general; al final de la arquitectura de CNN. Su operación consiste en una multiplicación como una matriz simple seguida de la suma de un vector umbral o *bias*.

Dropout es una técnica de regularización de modelos de redes neuronales que consiste en que por cada lote que ingresa a la fase de entrenamiento, se desactiva de manera aleatoria un porcentaje de neuronas de la siguiente capa, basado en la probabilidad p previamente definida. [22]

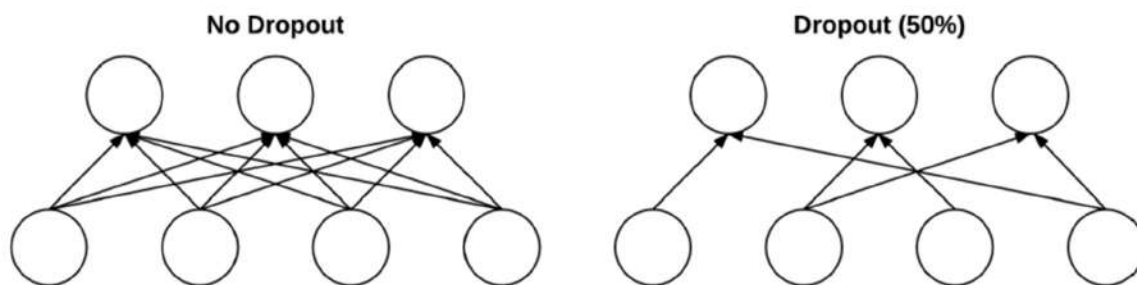


Figura 6: Ejemplo de usar la configuración de la probabilidad $p=0,5$. Fuente: *Uso de técnicas de deep learning para reconocimiento de objetos en áreas Rurales (2021)*

1.5. Funciones de pérdida

La función de pérdida ajusta los pesos del modelo para los objetivos de optimización minimizando los desajustes de regiones entre las segmentaciones predichas y las de verdad fundamental. [26]

Binary cross-entropy es una de las funciones comunes de pérdida utilizada generalmente en problema de clasificación de imágenes que permite asignar probabilidades decimales a cada clase, donde existen múltiples clases. Su forma es como sigue [22]:

$$J(\hat{p}_n) = \frac{-1}{N} \sum_{n=1}^N l_n \log \hat{p}_n + (1 - l_n) \log(1 - \hat{p}_n)$$

Donde:

N=número de datos de entrenamiento.

l_n = indica las etiquetas correctamente clasificadas de las clases.

\hat{p}_n = es la probabilidad de la clase.

1.5.1. Back propagation

Este es uno de los métodos más importante de aprendizaje en una red neuronal. Consiste en calcular el gradiente respecto a todos los pesos de la red y usar el descenso de este para actualizar los pesos y parámetros con la finalidad de minimizar el error de salida. [22] Este consta de dos partes:

- *Forward pass* (fase de propagación): las entradas ingresan a través de la red y se obtiene una predicción de salida
- *Backward pass* (Fase de actualización de pesos): calcula el valor del gradiente de la función de pérdida de la capa final. Con ese valor de gradiente se recorre hacia atrás para actualizar los pesos de la red.

1.6. Segmentación semántica de objetos

Segmentación semántica consiste en encontrar objetos previamente definidos en una imagen digital y se aplica en tecnologías como imágenes médicas, interacción humano-maquina, realidad aumentada, segmentación y clasificación de grietas, entre otros. [27]

U-Net fue presentada por Ronneberger en la conferencia MICCAI en 2015 [28]. La U-Net es una red neuronal convolucional en forma de U que se utilizó originalmente en el campo de la segmentación de imágenes médicas. Tiene dos ramas simétricas y se considera una estructura de red de codificador-decodificador. [16] La arquitectura de la U-Net se explica en la *Figura 6*.

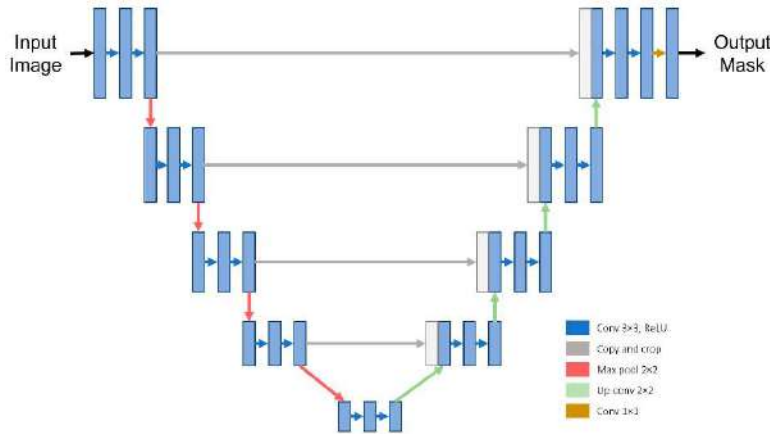


Figura 7. U-Net. Fuente: *Arquitectura CNN basada en U-Net para segmentación de grietas en carreteras (2015)*

En la rama codificadora se extraen las características más importantes de la entrada y son enviadas a la rama decodificadora para iniciar un proceso de convolución para ampliar los mapas de características de forma adecuada para posteriormente ser concatenados y ampliados nuevamente mediante conexiones de salto, de esa forma el decodificador produce los mapas de características necesarios. Luego se añaden las capas de convolución y de normalización. La salida del bloque decodificador se envía a una capa de convolución que seguidamente se introduce a una función de activación sigmoidea para finalmente producir la máscara binaria. [16]

Poniendo como ejemplo el problema principal de investigación. Dentro de cada foto tomada del pavimento podrán o no haber grietas de diferentes formas, tamaños y características; para poder clasificar estas grietas primero se debe poder detectar la existencia o no de la misma en la imagen y también señalar su ubicación dentro de la misma, esto pues, es en lo que consiste la segmentación pixel a pixel. En palabras simples, es el primer paso para todo análisis de imágenes mediante CNN.

2. Procesamiento de imágenes

Como explican F. González y A. Zeballos el procesamiento de imágenes es una técnica para transformar una imagen digital para mejorar su calidad, remover ruido, realzar detalles o para realizar alguna tarea específica. Esto se logra mediante el uso de algoritmos para procesar la

imagen digital para lograr los objetivos deseados. El procesamiento de imágenes también se utiliza para detectar objetos, realizar seguimiento de los mismos, clasificación de objetos, análisis de características, etc. [24]

F. González y A. Zeballos se refieren a que una imagen digital puede ser expresada por una matriz ordenada de números enteros, conocido también como pixel o pixeles, el tamaño de la imagen se determina por la cantidad de columnas M (ancho) y la cantidad de filas N (alto). Para ubicar un elemento de la imagen (pixel), se utiliza un sistema de coordenadas donde el origen es la esquina superior izquierda y va de arriba hacia abajo y de izquierda hacia la derecha. [24]

Así también mencionan que los valores del pixel son palabras binarias de longitud k, denominado profundidad de bits; de modo que un pixel puede representar cualquiera de 2^k valores diferentes. Estos valores a su vez dependen del tipo de imagen (Binario, Escala de grises o RGB). [24]

Para esta investigación, servirá para analizar y mejorar la visibilidad de las características relevantes en las imágenes capturadas del pavimento, lo que permitirá identificar y medir las grietas de manera automatizada y precisa.

3. Métodos Tradicionales de Detección de Grietas utilizados en Paraguay

3.1. Inspección visual humana

Actualmente, los únicos métodos aplicados para la detección de grietas utilizadas en Paraguay, según el manual de carreteras del Paraguay [29] es la inspección visual. De acuerdo a la tolerancia descrita en el manual ninguna grieta mayor a 3mm no podrá permanecer sin sello por más de 24hs.

4. Definición de términos básicos

4.1. Grietas

El manual de carreteras del Paraguay [29] indica que toda abertura superior a 3mm es considerada una grieta que debe ser sellada a lo largo de su línea mediante sellado en frío o caliente con un mástic asfáltico pertinente para asegurar la impermeabilización de la estructura en el sitio tratado.

Las grietas se producen a raíz de la eventual contracción de las capas inferiores del pavimento y se reflejan en la superficie de rodadura de la carretera que pueden ser producidas a su vez por envejecimiento del asfalto, baja densidad de la base, falta de soporte lateral, asentamiento de la subrasante, mala unión entre pavimento y banquina, unión débil entre pistas contiguas, reflexión

de grietas desde capas inferiores (para el caso de grietas longitudinales) o debido a el termino de la vida de diseño, volumen de tránsito excesivo, espesor de capas insuficiente deflexión excesiva por inestabilidad de la subrasante en capas granulares, debido a la falta de compactación y/o problemas de drenaje (para casos de grietas piel de cocodrilo) [29].

Sin embargo, el sellado de grietas no se debe ejecutar en áreas en que las grietas formen bloques interconectados de carácter poliédrico, semejante al agrietamiento piel de cocodrilo, cuya formación se debe en mayor parte a la fatiga del pavimento, debido a repeticiones de carga por exceso de peso en los ejes, por lo tanto, implica otro tipo de tratamiento. [29]

Para este trabajo se considerará grieta a toda aquella en la que se observe una separación de 3mm o superior, ya sea aquellas que formen bloques interconectados de carácter poliédrico, o las que no.

4.2.Base de datos normalizadas por lotes

Una base de datos de imágenes, brinda un apoyo al momento de entrenar y validar algoritmos de visión artificial [22].

Según Szegedy et al. [30] la ganancia (disminución sustancial del error entre evaluaciones con una menor cantidad de sumas múltiples, menor cantidad de parámetros y menor costo computacional) se atribuye al espacio mejorado de variaciones que la red puede aprender especialmente cuando se normalizan los lotes.

Por ejemplo, aplicar la normalización al objetivo de este proyecto consistiría en utilizar un conjunto de datos que contiene imágenes de pavimento asfáltico, algunas de las cuales tienen grietas y otras que no. Cada imagen se etiqueta como "con grietas" o "sin grietas".

Luego de diseñar una arquitectura de red neuronal convolucional; que consta de varias capas convolucionales y capas completamente conectadas, para aprender las características relevantes de las imágenes y hacer la clasificación de "con grietas" o "sin grietas", se aplica la normalización por lotes. Esto significa que, para cada lote de imágenes de entrenamiento con el que alimentamos a la red durante el entrenamiento, se calculan las estadísticas de normalización, como la media y la desviación estándar, para cada canal de activación en esa capa. Luego, se normalizan las activaciones utilizando esas estadísticas.

Durante el entrenamiento, la red aprenderá gradualmente a reconocer las características asociadas a las grietas en el pavimento. La normalización por lotes ayudará a que el proceso de aprendizaje

sea más eficiente y a evitar problemas como el desvanecimiento del gradiente y el saturamiento de la memoria.

4.3.Redes neuronales

Las redes neuronales artificiales o *Artificial Neural Networks* son una rama del área denominada aprendizaje profundo que están inspiradas en las neuronas o células nerviosas de los seres vivos. Básicamente una neurona tiene como misión recibir información de otras neuronas o de receptores, procesa esta información en un código de activación interno y luego transmite nuevos impulsos a las neuronas siguientes u otras células. [21, 22]

En otras palabras, consiste en un complejo método de análisis de datos computacionales que toma como referencia el comportamiento de las neuronas humanas sistematizando ciertos procesos algorítmicos para llegar a un resultado basado en datos de aprendizaje previamente introducidos.

4.4.Deep learning

El aprendizaje profundo o *deep learning* es una colección de técnicas de la Red neuronal Artificial con un número significativo de capas interconectadas. [22]

Por lo tanto, el aprendizaje profundo es un método de análisis automatizado basado en redes neuronales artificiales que está compuesto de un gran número de capas de distintos niveles de profundidad y capas ocultas.

4.5.Visión por computadora

Es una rama de la Inteligencia artificial usada para la detección de objetos. [22]

Contrario a la simpleza de su concepto; las aplicaciones que se le da a la visión por computadora le dan sentido a la existencia de muchas de las definiciones anteriormente explicadas. En este sentido, la visión por computadora permitiría llevar a cabo el monitoreo de forma automática de las degradaciones que se forman en el asfalto. Un constante monitoreo y sistema de alarma adecuado permitirá realizar el mantenimiento preventivo en aquellas calles con señales tempranas de fallas y ahorrar tiempo, mano de obra y dinero en la búsqueda de estas.

Objetivos

Objetivo General

Desarrollar un algoritmo con redes neuronales artificiales convolucionales para segmentar grietas utilizando imágenes tomadas del pavimento asfáltico.

Objetivos Específicos

- Diseñar una base de datos de imágenes del pavimento para la optimización del proceso de entrenamiento de la red neuronal convolucional y el aumento de la precisión en la detección de grietas en el pavimento.
- Implementar las mejoras identificadas durante la evaluación del desempeño del algoritmo para el incremento de la precisión y la fiabilidad en la detección de grietas en el pavimento.
- Evaluar la eficacia del algoritmo desarrollado en términos de precisión en la segmentación de grietas y eficiencia computacional.

METODOLOGÍA

Este estudio sigue una investigación experimental aplicada. El objetivo fue implementar y evaluar un modelo de redes neuronales convolucionales (U-Net) para la segmentación y cuantificación de grietas en pavimentos asfálticos, mediante procesamiento de imágenes. Se generó un sistema automatizado capaz de detectar grietas en pavimentos utilizando aprendizaje profundo.

1. Recopilación de Datos en la Zona de Coronel Oviedo, Paraguay

1.1. Elección de ubicaciones representativas

En total se tomaron 342 imágenes sin grietas y 505 imágenes con grietas de 5 ubicaciones distintas las cuales son:

- HH54+4V4, Carlos Antonio López, Cnel. Oviedo 050107
- GGMH+5C Cnel. Oviedo
- GGMP+G9X Cnel. Oviedo.
- -25.473234, -56.449340. Dr. Blas Garay, Cnel. Oviedo 050132
- -25.428268, -56.450806. Cerrito Rugua, Cnel. Oviedo 050101

1.2. Captura de imágenes con teléfonos inteligentes

Se capturaron imágenes de las ubicaciones seleccionadas utilizando un teléfono inteligente soportados mediante un atril posicionado a una altura de 110 centímetros (1,10 metros). Estas imágenes son de alta resolución y se tomaron en diversas condiciones de luz para obtener una muestra diversa.

Las imágenes fueron tomadas desde un teléfono inteligente con las siguientes características:

- Marca: Motorola
- Modelo: moto g52
- Sistema operativo: Android version 12
- Procesador: Snapdragon 680 4G
- Tarjeta gráfica (GPU): Adreno 610
- Características de la cámara:
 - Resolución de 50 MP
 - Apertura de f/2.2
 - Cuenta con cámara dual
- Tamaño de la imagen: 2296x4080, 72ppp, 24bits

Cabe destacar que inicialmente las imágenes fueron tomadas desde la aplicación Arduino Science Journal sin embargo, por motivos de espacio de almacenamiento se tomaron 143 fotos mediante otra dispositivo inteligente el cual presenta las siguientes características:

- Marca: Tecno Mobile
- Modelo: Tecno Spark 10 pro
- Sistema operativo: Android version 13
- Procesador: Mediatec Helio G88
- Tarjeta gráfica (GPU): Mali-G52 MC2
- Características de la cámara:
 - Resolución de 50 MP
 - Triple cámara trasera IA
 - Doble flash
- Tamaño de la imagen: 700x1600, 96pp, 24bits

Las características del equipo utilizado son las siguientes:

- Procesador: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- RAM Instalada: 16.0 GB (15.8 GB utilizable)
- Unidad de procesamiento de gráficos (GPU): NVIDIA GeForce RTX 2060-6GBVRAM

1.3. Protocolo de etiquetado de grietas y segmentación de imágenes

El protocolo para etiquetar claramente las áreas con grietas en las imágenes fue el siguiente: de forma manual se resaltaron las grietas con un color altamente saturado, en este caso el negro (0,0,0) dejando de fondo el color blanco, de esta forma se crearon las máscaras para el proceso de entrenamiento. Se tomaron 395 imágenes al azar con grietas para ser enmascaradas manualmente, 110 imágenes con grietas fueron utilizadas para realizar las pruebas. Adicionalmente se tomaron 342 fotos de pavimento sin grietas y 142 de las mismas fueron utilizadas para las pruebas. En total se tomaron 847 fotos, 505 de ellas con grietas y 342 sin grietas.

Debido a que la arquitectura U-Net utiliza imágenes cuadradas, se dividieron las imágenes en dos (2) y así también sus respectivas máscaras, también se redujo el tamaño de las imágenes recortadas a 512x512 pixeles cada una.

2. Entrenamiento de la Red Neuronal Convolutiva

La red utilizada en el presente estudio sigue la arquitectura *U-Net* [28], la cual está diseñada específicamente para tareas de segmentación de imágenes. Esta red está compuesta por dos partes principales: un codificador y un decodificador, que trabajan en conjunto para identificar regiones específicas en las imágenes de entrada.

2.1. Codificador (Encoder)

El codificador de la red tiene cuatro capas principales que permiten extraer características de alto nivel a partir de las imágenes de entrada. Cada una de estas capas se compone de operaciones de convolución, normalización por lotes y la función de activación ReLU [28].

Cada capa convolutiva aplica filtros de tamaño 3x3 a las imágenes de entrada, donde los filtros son matrices entrenables que permiten a la red detectar características específicas en diferentes áreas de la imagen. Este proceso se repite dos veces por cada capa para capturar más detalles.

Después de cada operación de convolución, se realiza una normalización por lotes para así ajustar los valores de salida de la convolución y que de esa forma tengan una distribución más uniforme, lo cual estabiliza el proceso de entrenamiento y acelera la convergencia.

Seguidamente, se aplica la función de activación ReLU (Rectified Linear Unit [28]), que introduce no linealidad en la red. Esta función transforma los valores negativos en cero, ayudando a la red a manejar patrones complejos en los datos de entrada.

Al final de cada capa del codificador, se aplica una operación de agrupación máxima (MaxPooling) [22] con un tamaño de 2x2 y un paso de 2, lo que reduce el tamaño de la imagen en un factor de 2. Este proceso permite que la red se enfoque en las características más importantes a medida que reduce la resolución de las imágenes.

El número de filtros o canales de características comienza en 32 en la primera capa y se duplica a medida que avanza por las etapas del codificador, llegando a 256 en la parte final del codificador, conocida como el "cuello de botella". Este patrón ayuda a la red a capturar tanto los detalles finos como las estructuras más amplias.

2.2. Cuello de botella (Bottleneck)

Entre el codificador y el decodificador se encuentra una capa especial llamada el cuello de botella, donde la resolución de la imagen es más baja y el número de características es máximo. En esta

etapa, la red ha condensado toda la información de la imagen en un espacio más reducido y ha aprendido las características clave necesarias para la segmentación [28].

2.3. Decodificador (Decoder)

El decodificador [28] utiliza una operación de muestreo ascendente (*UpSampling*), para aumentar el tamaño del mapa de características en un factor de 2 en cada dimensión, lo que permite restaurar la imagen a su tamaño original, de esa forma en cada capa del decodificador, los mapas de características del codificador correspondientes se añaden directamente mediante una operación de concatenación. Esto significa que la información detallada extraída en las primeras etapas del codificador se combina con la información reconstruida del decodificador, lo que permite que la red recupere mejor las características perdidas durante la reducción de resolución.

Después de cada operación de muestreo ascendente y concatenación, se aplican dos capas convolucionales adicionales para refinar las predicciones de la red. El número de canales de características en el decodificador comienza en 256 y se reduce a medida que la imagen se va reconstruyendo, terminando con 64 en la última capa.

2.4. Capa de salida

La capa final de la red es una capa de convolución con un solo filtro de tamaño 1x1 que reduce las características a una sola salida por píxel [22]. Esta capa utiliza una activación sigmoidea, lo que permite que la red realice predicciones binarias para cada píxel (grieta o no grieta). El valor de salida es un número entre 0 y 1, y se utiliza un umbral de 0.5 para determinar si un píxel pertenece a la clase de interés (grieta).

2.5. Función de pérdida y métrica

La función de pérdida utilizada es Binary Crossentropy [22], que mide la diferencia entre las predicciones de la red y los valores reales de la máscara de segmentación. La métrica seleccionada para evaluar el rendimiento del modelo es la exactitud binaria (Binary Accuracy), que calcula cuántos píxeles fueron clasificados correctamente.

2.6. Optimizador Adam

El modelo se entrena utilizando el optimizador Adam [31], un algoritmo de optimización que ajusta los pesos de la red de manera eficiente durante el entrenamiento. Adam es conocido por su

capacidad de converger rápidamente y manejar grandes cantidades de datos, lo que lo hace ideal para este tipo de tareas. La tasa de aprendizaje inicial seleccionada es de $1e-4$.

2.7. Entrenamiento y validación

El modelo se entrena en imágenes de 256×256 píxeles, utilizando un tamaño de lote de 8 y un total de 500 épocas. El conjunto de datos se divide en un 80% para entrenamiento y 20% para validación, asegurando que la red aprenda de forma efectiva sin sobreajustar.

2.8. Métricas de evaluación

Las métricas más comunes de evaluación para la segmentación son el F1 Score y la intersección sobre la unión (IoU por sus siglas en inglés).

F1 Score es la combinación de precisión y recuperación (*Recall*) calculada como la media armónica de las dos cantidades. [32] Su ecuación es como sigue:

$$F_1 = \frac{2PR}{P + R}$$

Donde:

$$P = \frac{TP}{TP+FP} \text{ y } R = \frac{TP}{TP+FN}$$

TP = verdaderos positivos (*true positive*)

FP = falsos positivos (*false positive*)

FN= falsos negativos (*false negative*)

La intersección sobre la unión (*IoU*) es un tipo de medida geométrica que describe la proximidad de los resultados previstos a los cuadros delimitadores de los resultados verdaderos y se expresa como [16]:

$$IoU = \frac{Area(B_P \cap B_g)}{Area(B_P \cup B_g)}$$

RESULTADOS Y ANÁLISIS

En la Figura 7 se muestran las curvas de precisión tanto del conjunto de entrenamiento como del de validación. Es evidente que ambas curvas presentan una tendencia creciente hasta estabilizarse en valores altos y similares, con una diferencia de apenas 0.00016. Este comportamiento sugiere que el modelo ha alcanzado un aprendizaje adecuado sin indicios de sobreajuste o subajuste.

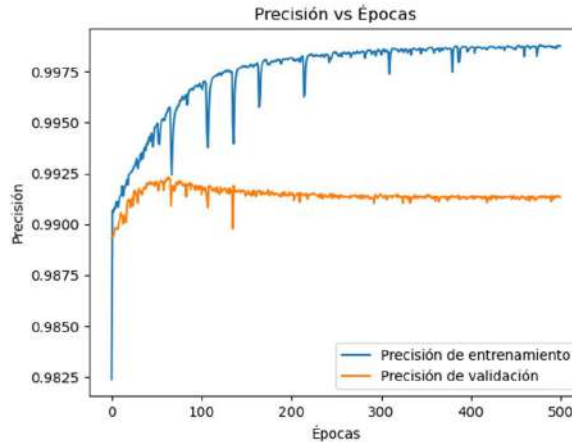


Figura 8. Gráfico de precisión vs épocas. Fuente: Elaboración propia.

A continuación, se presentan los resultados obtenidos en la evaluación del modelo propuesto, comparándolos con los resultados reportados por Di Benedetto et al. (2023) para un modelo basado en U-Net. Las métricas incluyen *F1 Score* e *IoU* (Intersección sobre la Unión), entre otras [16].

Tabla 1: Comparación de métricas entre el modelo propuesto y el modelo de Di Benedetto et al. (2023).

Método	Precisión	Recall	F1 Score	IoU
Di Benedetto et Al.	0.9692	0.6902	0.7301	0.5391
Propuesta	0.9912	0.9971	0.9956	0.9913

Aunque la base de datos utilizada en el estudio de Di Benedetto et al. difiere de la empleada en esta investigación, la arquitectura de red neuronal es comparable. El aumento observado en la precisión del modelo propuesto sugiere una mayor confiabilidad, es decir, una reducción en la tasa de falsos positivos. La elevada recuperación del modelo implica una alta sensibilidad en la detección de grietas, aunque esto también contribuye a la aparición de algunos falsos positivos (por ejemplo, sombras y patas del atril), así como falsos negativos en grietas muy pequeñas. Sin

SEGMENTACIÓN DE GRIETAS EN PAVIMENTO ASFÁLTICO UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL LUZ RASHELL SOTO OLGUIN – 2024

embargo, el aumento en el $F1$ Score y el IoU es notable, lo cual es atribuible al incremento en el número de épocas de entrenamiento (500 en total) y el tamaño de lote de 8.

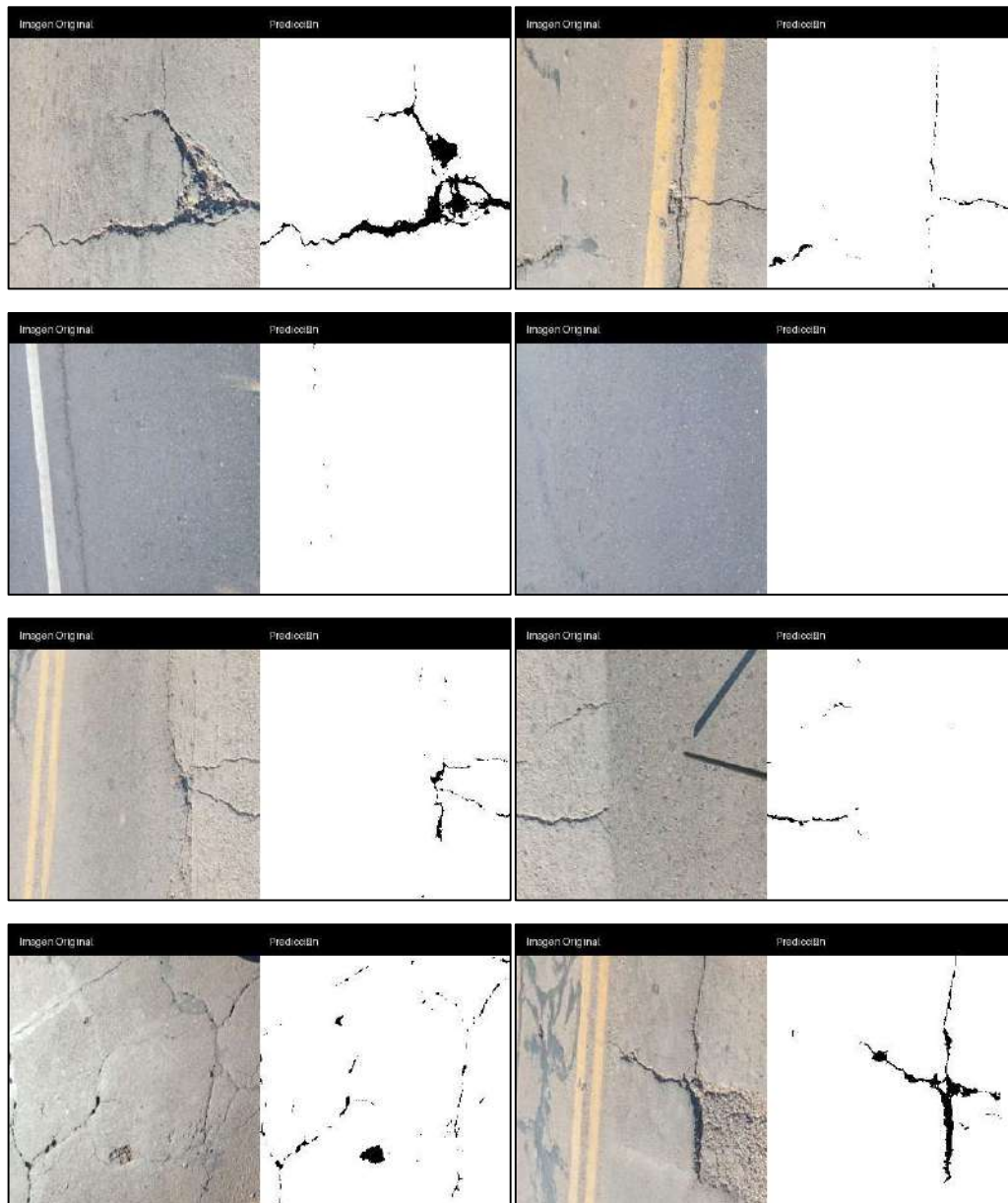


Figura 9. Resultados obtenidos en las imágenes de prueba. Fuente: Elaboración propia

Además, para garantizar la precisión del modelo en la detección de grietas de 3 mm o mayores, se desarrolló un código en Python que cuantifica el ancho de las grietas en las imágenes segmentadas. Este código utiliza la función `skeletonize` de `skimage` [33] para extraer el esqueleto de las grietas y la función `distance_transform_edt` de `scipy.ndimage` [34] para calcular la distancia desde el esqueleto a los bordes. Posteriormente, se multiplica dicha distancia por 2 para obtener el ancho

de las grietas en píxeles, convirtiéndose luego a milímetros usando un factor de conversión de 0.924 mm/píxel, obtenido de un objeto de referencia.

Para facilitar la visualización, se generó una imagen categorizada donde las grietas con anchos mayores a 3 mm se destacan en rojo, mientras que las grietas de anchos iguales o menores a 3 mm se muestran en verde, alineándose con los criterios del MOPC [29]. El tamaño mínimo de grieta detectable con este método es de 1.85 mm.

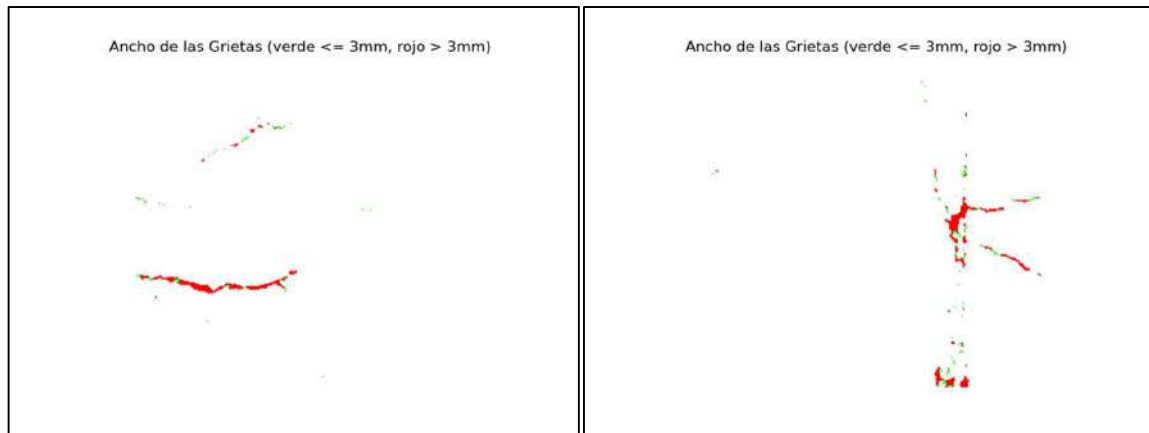


Figura 10. Grietas detectadas por el programa. Rojo: grietas mayores a 3mm. Verde: Grietas menores o iguales a 3mm. Tamaño mínimo de grieta detectada: 1,85 mm.

CONCLUSIONES Y RECOMENDACIONES

El desarrollo del algoritmo basado en redes neuronales convolucionales ha demostrado ser efectivo para segmentar y cuantificar grietas en el pavimento asfáltico, alcanzando resultados destacables en términos de precisión y confiabilidad. Las curvas de precisión del modelo mostraron un aprendizaje adecuado, estabilizándose en niveles altos y con una brecha mínima entre los conjuntos de entrenamiento y validación, lo que indica un buen desempeño general del modelo.

La creación de una base de datos optimizada de imágenes del pavimento fue fundamental para mejorar el proceso de entrenamiento y aumentar la precisión del modelo en la detección de grietas. Esta base de datos permitió al modelo aprender a segmentar grietas de diferentes características, incluyendo aquellas en condiciones complejas de iluminación o texturas del pavimento, fortaleciendo así su capacidad de generalización y confiabilidad. Tanto la base de datos como el modelo entrenado se encuentran disponible en <https://github.com/LuzSoto1999/Segmentacion-de-grietas-superficiales-en-pavimento-asf-ltico>.

Las evaluaciones del desempeño del modelo durante su desarrollo permitieron implementar mejoras que aumentaron su precisión y fiabilidad. Las métricas F1 Score (0.9956) e IoU (0.9913) evidencian que el modelo es sensible a la detección de grietas, aunque se registraron algunos falsos positivos y negativos, particularmente en situaciones de sombras o cuando las grietas eran muy pequeñas. A pesar de esto, la arquitectura utilizada mostró ser más confiable en comparación con estudios previos, lo que sugiere que el algoritmo es adecuado para su aplicación práctica.

Finalmente, el código implementado para cuantificar el ancho de las grietas permitió identificar grietas de 3 mm o más, estableciendo un umbral claro entre las grietas que representan un riesgo alto y aquellas que no. Con un tamaño mínimo de grieta detectada de 1,85 mm, el algoritmo se posiciona como una herramienta valiosa para el monitoreo y mantenimiento de pavimentos asfálticos, contribuyendo así a la mejora de la infraestructura vial.

Recomendaciones

1. Se sugiere recopilar un conjunto de imágenes más diverso y extenso de diferentes tipos de pavimentos y condiciones ambientales para mejorar la robustez del modelo. Esto podría ayudar a minimizar los falsos positivos y negativos, especialmente en situaciones desafiantes como sombras o variaciones de iluminación.

2. Investigar métodos alternativos para la cuantificación del ancho de las grietas, como el uso de técnicas de aprendizaje profundo que no solo se enfoquen en la segmentación, sino también en la medición directa del ancho, lo que podría aumentar la precisión en la identificación de grietas críticas.
3. Desarrollar un sistema que permita la detección y cuantificación de grietas en tiempo real mediante el uso de cámaras montadas en vehículos, facilitando el monitoreo continuo de la infraestructura vial.
4. Considerar la incorporación de datos complementarios, como información de estrés y deformaciones del pavimento, para proporcionar un análisis más holístico del estado del pavimento y predecir el deterioro.
5. Al Ministerio de Obras Públicas y Comunicaciones (MOPC), incorporar un sistema automatizado de inspección vial basado en redes neuronales para priorizar y planificar el mantenimiento de carreteras. Esto puede reducir significativamente los tiempos de diagnóstico y optimizar la asignación de recursos.
6. A las Municipalidades, implementar un sistema de monitoreo de grietas que permita detectar deterioros en las calles urbanas más transitadas, mejorando la planificación de reparaciones y la seguridad de los usuarios.
7. Al Ministerio de Economía y Finanzas, financiar la adquisición e implementación de tecnologías basadas en inteligencia artificial para la evaluación vial, demostrando su viabilidad económica mediante estudios de costo-beneficio en regiones prioritarias.

BIBLIOGRAFÍA

- [1] Secretaría Técnica de Planificación y de desarrollo Económico y Social de Paraguay, «Plan Nacional de Desarrollo Paraguay 2030,» Diciembre 2014. [En línea]. Available: <https://www.stp.gov.py/pnd/wp-content/uploads/2014/12/pnd2030.pdf>. [Último acceso: 18 Agosto 2023].
- [2] Ministerio de Obras Públicas y Comunicaciones, «Contador de nuevos asfaltados habilitados,» 31 Julio 2023. [En línea]. Available: <https://www.mopc.gov.py/index.php/contador-de-nuevos-asfaltados-habilitados>. [Último acceso: 18 Agosto 2023].

- [3] E-asfaltos, «Red Vial Argentina,» [En línea]. Available: <https://www.e-asfalto.com/redvialarg/redvial.htm>. [Último acceso: 27 Noviembre 2023].
- [4] A. Buhler, «Caminos rurales, una puerta al desarrollo y la conectividad territorial / Brasi,» Corporación Andina de Fomento, 2023. <https://scioteca.caf.com/handle/123456789/2160>.
- [5] B. Budiansky y R. J. O'connell, «Elastic moduli of a cracked solid,» *International Journal of Solids and Structures.*, vol. 12, n° [https://doi.org/10.1016/0020-7683\(76\)90044-5](https://doi.org/10.1016/0020-7683(76)90044-5)., pp. 81-97, 1976.
- [6] J. Aboudi, «Reducción la rigidez de los sólidos fisurados,» *Eng. Fract. Mech*, vol. 26, n° [https://doi.org/10.1016/0013-7944\(87\)90129-9](https://doi.org/10.1016/0013-7944(87)90129-9), p. 637–650, 1987.
- [7] Y. Fujita y Y. Hamamoto, «A robust automatic crack detection method from noisy concrete surfaces, Mach,» *Vis. Appl.*, vol. 22, n° <https://doi.org/10.1007/s00138-009-0244-5>, pp. 245-254, 2016.
- [8] D. Dhital y J. Lee, Un sistema de imágenes de propagación ultrasónica sin contacto para la evaluacion de grietas en superficies cerradas, <https://link.springer.com/article/10.1007/s11340-011-9567-z>, 2012.
- [9] M. Vagnoli, R. Remenye-Prescott y J. Andrews, «Monitoreo de la salud estructural de puentes ferroviarios y detección de fallas: métodos de vanguardia y desafíos futuro,» *Struct. Sanar.*, vol. 17, n° <https://doi.org/10.1177/1475921717721137>, pp. 971-1007, 2018.
- [10] Y. Sen Yang, C. M. Yang y C. W. Huang, «Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis,» *Adv. Eng. Softw*, vol. 83, n° <https://doi.org/10.1016/j.advengsoft.2015.02.005>., pp. 99-108, 2015.
- [11] D. Dias-da-Costa, J. Valença, E. Júlio y H. Araújo, «Crack propagation monitoring using an image deformation approach,» *Struct. Control Heal. Monit*, vol. 24, n° <https://doi.org/10.1002/stc.1973>., pp. 1-14, 2017.
- [12] G. Li, S. He, Y. Ju y K. Du, «Long-distance precision inspection method for bridge,» *Autom. Constr*, vol. 41, n° <https://doi.org/10.1016/j.autcon.2013.10.021>., pp. 83-95, 2014.

- [13] S. Nishiyama, N. Minakata, T. Kikuchi y T. Yano, «Improved digital photogrammetry technique for crack monitoring,» *Adv. Eng. Informatics*, vol. 29, n° <https://doi.org/10.1016/j.aei.2015.05.005>, pp. 851-858, 2015.
- [14] J. Long, E. Shelhamer y T. Darrell, «Fully convolutional networks for semantic segmentation,» de *In Proceedings of the IEEE conference on computer vision and pattern recognition*, <https://ieeexplore.ieee.org/document/7298965>, (pp. 3431-3440), 2015.
- [15] S. Y. Huang, W. L. Hsu, R. J. Hsu y D. W. Liu, «Fully convolutional network for the semantic segmentation of medical images: A survey,» <https://doi.org/10.3390/diagnostics12112765>, 2022.
- [16] A. Di Benedetto, M. Fiani y L. M. Gujski, «Arquitectura CNN basada en U-Net para segmentación de grietas en carreteras,» *Infraestructuras. Técnicas de Teledetección para la Inspección y Monitoreo de Infraestructuras*, vol. 8, n° 5, <https://doi.org/10.3390/infraestructuras8050090>, pp. 1-13, 2023.
- [17] L. Tello-Cifuentes, M. Aguirre-Sánchez, J. P. Díaz-Paz y F. Hernández, «Evaluación de daños en pavimento flexible usando fotogrametría terrestre y redes neuronales,» *TecnoLógicas*, vol. 24, n° 50, <https://www.redalyc.org/journal/3442/344264813013/344264813013.pdf>, 2021.
- [18] M. A. Roman-Garay, L. A. Morales-Rosales, H. Rodríguez-Rangel y S. . I. Fernandez-Gregorio, «Deteccion de deterioros superficiales en pavimentos flexibles basada en segmentacion semantica y redes transformadoras,» *Research in Computing Science*, vol. 152(6), https://rcs.cic.ipn.mx/2023_152_6/, p. 273–286, 2023.
- [19] S. Calderón Uribe, U. Calderón Uribe, L. A. Morales Hernández y I. A. Cruz Albarrán, «Segmentación Automática deCálculos RenalesmedianteU-Net a través de Imágenes de Tomografía Computarizada,» *Jovenes en la Ciencia*, <https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/4222>, pp. 1-8, 2023.
- [20] A. B. Casilla Gallegosa y W. E. Abad, «Detección automática del daño en estructuras de concreto reforzado usando redes neuronales,» *Journal of Research and Innovation in Civil*

Engineering, vol. 1, nº 2, <https://revistas.unam.edu.pe/index.php/jrice/article/view/10>, pp. 65-70, 2021.

- [21] F. G. Ramirez Villanueva, técnicas de aprendizaje automático para análisis a flexión en vigas rectangulares de hormigón armado, San Lorenzo, <https://drive.google.com/drive/search?q=%C3%89CNICAS%20DE%20APRENDIZAJE%20AUTOM%C3%81TICO%20PARA%20AN%C3%81LISIS%20A%20FLEXI%C3%93N%20EN%20VIGAS%20RECTANGULARES%20DE%20HORMIGON%20ARMADO>, 2019.
- [22] L. R. Barba Guamán, Uso de técnicas de deep learning para reconocimiento de objetos en áreas rurales, Madrid: Universidad Politécnica de Madrid, <https://oa.upm.es/67969/>, 2021.
- [23] P. I. Viñuela y I. M. Galván León, REDES DE NEURONAS ARTIFICIALES. UN ENFOQUE PRÁCTICO., Madrid: Pearson educación S.A., <https://sedici.unlp.edu.ar/handle/10915/9491>, 2004.
- [24] F. E. González Barreto y A. A. Zevallos Chaparro, ANÁLISIS DE LA RESISTENCIA DEL HORMIGÓN A COMPRESIÓN MEDIANTE PROCESAMIENTO DE IMÁGENES, Coronel Oviedo, <https://drive.google.com/drive/search?q=AN%C3%81LISIS%20DE%20LA%20RESISTENCIA%20DEL%20HORMIG%C3%93N%20A%20COMPRESI%C3%93N%20MEDIANTE%20PROCESAMIENTO%20DE%20IM%C3%81GENES>, 2022.
- [25] R. Usha, T. Prasannavenkatesan , J. Jacob y V. Yendapalli, «Binary cross entropy with deep learning technique for Image Classification,» International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, nº 4, <https://www.warse.org/IJATCSE/static/pdf/file/ijatcse175942020.pdf>, p. 5393 – 5397, 2020.
- [26] M. Montazerolghaem, S. Yu, G. Sasso y A. Haworth, «Arquitectura U-Net para la segmentación de próstata: el impacto de la función de pérdida en el rendimiento del sistema,» Bioingeniería, vol. 10, nº 4, <https://www.mdpi.com/2306-5354/10/4/412>, 2023.

- [27] O. A. Soto Orozco, A. D. Corral Sánenz, C. E. Rojo González y J. A. Ramire Quintana, «Análisis de desempeño de redes neuronales profundas para segmentación semantida en hardware limitado,» ReCIBE, vol. 8, n° 2, <https://recibe.cucei.udg.mx/index.php/ReCIBE/article/view/142>, pp. 1-21, 2019.
- [28] O. Ronneberger, P. Fischer y T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation,» de En Actas de la Conferencia internacional sobre computación de imágenes médicas e intervención asistida por computadora, Minich, Alemania, <https://arxiv.org/pdf/1505.04597>, 2015.
- [29] Asociación Paraguaya de Carreteras, «UNIDAD 7: Mantenimiento y gestión de carreteras,» Manual de carreteras del Paraguay, vol. 7, n° 1, <https://es.scribd.com/document/605918539/9-Unidad-7-Mantenimiento-y-Gestion-de-Carreteras-Paraguay>, p. 164, 2019.
- [30] C. Szegedy, V. Vanhoucke, S. Ioffe y J. Shlens, «Rethinking the Inception Architecture for Computer Vision,» Proceedings of the IEEE conference on computer vision and pattern recognition, https://openaccess.thecvf.com/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html, pp. 2818-2826, 2016.
- [31] D. P. Kingma y J. Ba, «Adam: A Method for Stochastic Optimization,» <https://arxiv.org/abs/1412.6980>, vol. 9, 2014.
- [32] «scikit-image,» [En línea]. Available: https://scikit-image.org/docs/stable/auto_examples/edges/plot_skeleton.html. [Último acceso: 28 noviembre 2024].
- [33] «scipy,» [En línea]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.distance_transform_e [.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.distance_transform_e.html). [Último acceso: 2024 noviembre 28].

ANEXOS

Anexo A: Código del Algoritmo de Segmentación

El siguiente código implementa un algoritmo basado en redes neuronales convolucionales (CNN) para la segmentación de grietas en pavimentos asfálticos. Se utilizó la biblioteca TensorFlow y Keras para la construcción y entrenamiento del modelo U-Net. A continuación se presentan las líneas de código relevantes:

```
import tensorflow as tf
from PIL import Image
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, UpSampling2D,
concatenate, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import BinaryCrossentropy
from tensorflow.keras.metrics import BinaryAccuracy
from tensorflow.keras.metrics import Recall
import tensorflow.keras.backend as K
import os
import numpy as np

# Configurar TensorFlow para usar solo la GPU 0
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    try:
        tf.config.set_visible_devices(gpus[0], 'GPU')
        tf.config.experimental.set_virtual_device_configuration(
            gpus[0],
            [tf.config.experimental.VirtualDeviceConfiguration(memory_limit=4096)]) # Establecer
el límite de memoria a 4 GB
    except RuntimeError as e:
        print(e)

# Verificar la configuración
print("GPUs visibles:", tf.config.get_visible_devices('GPU'))

# Verificar si se detecta alguna GPU
gpu_devices = tf.config.list_physical_devices('GPU')
if len(gpu_devices) == 0:
```

**SEGMENTACION DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFALTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
print("No se detectó ninguna GPU. TensorFlow está utilizando la CPU.")
else:
    print("Se detectó la siguiente GPU:")
    for device in gpu_devices:
        print(device)

def unet(input_size=(256, 256, 3)):
    inputs = Input(input_size)

    # Encoder
    c1 = Conv2D(32, (3, 3), activation='relu', padding='same')(inputs)
    c1 = Conv2D(32, (3, 3), activation='relu', padding='same')(c1)
    c1 = Dropout(0.3)(c1)
    p1 = MaxPooling2D((2, 2))(c1)

    c2 = Conv2D(64, (3, 3), activation='relu', padding='same')(p1)
    c2 = Conv2D(64, (3, 3), activation='relu', padding='same')(c2)
    c2 = Dropout(0.3)(c2)
    p2 = MaxPooling2D((2, 2))(c2)

    c3 = Conv2D(128, (3, 3), activation='relu', padding='same')(p2)
    c3 = Conv2D(128, (3, 3), activation='relu', padding='same')(c3)
    c3 = Dropout(0.4)(c3)
    p3 = MaxPooling2D((2, 2))(c3)

    c4 = Conv2D(256, (3, 3), activation='relu', padding='same')(p3)
    c4 = Conv2D(256, (3, 3), activation='relu', padding='same')(c4)
    c4 = Dropout(0.4)(c4)
    p4 = MaxPooling2D((2, 2))(c4)

    # Bottleneck
    c5 = Conv2D(512, (3, 3), activation='relu', padding='same')(p4)
    c5 = Conv2D(512, (3, 3), activation='relu', padding='same')(c5)
    c5 = Dropout(0.5)(c5)

    # Decoder
    u6 = UpSampling2D((2, 2))(c5)
    u6 = concatenate([u6, c4], axis=-1)
    c6 = Conv2D(256, (3, 3), activation='relu', padding='same')(u6)
    c6 = Conv2D(256, (3, 3), activation='relu', padding='same')(c6)
    c6 = Dropout(0.4)(c6)

    u7 = UpSampling2D((2, 2))(c6)
    u7 = concatenate([u7, c3], axis=-1)
    c7 = Conv2D(128, (3, 3), activation='relu', padding='same')(u7)
```

**SEGMENTACIÓN DE GRUETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
c7 = Conv2D(128, (3, 3), activation='relu', padding='same')(c7)
```

```
c7 = Dropout(0.4)(c7)
```

```
u8 = UpSampling2D((2, 2))(c7)
```

```
u8 = concatenate([u8, c2], axis=-1)
```

```
c8 = Conv2D(64, (3, 3), activation='relu', padding='same')(u8)
```

```
c8 = Conv2D(64, (3, 3), activation='relu', padding='same')(c8)
```

```
c8 = Dropout(0.3)(c8)
```

```
u9 = UpSampling2D((2, 2))(c8)
```

```
u9 = concatenate([u9, c1], axis=-1)
```

```
c9 = Conv2D(32, (3, 3), activation='relu', padding='same')(u9)
```

```
c9 = Conv2D(32, (3, 3), activation='relu', padding='same')(c9)
```

```
c9 = Dropout(0.3)(c9)
```

```
outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)
```

```
model = Model(inputs=[inputs], outputs=[outputs])
```

```
return model
```

```
def cargar_imagenes_y_mascaras(ruta_imagenes, ruta_mascaras, tamaño=(256, 256)):
```

```
    imagenes = []
```

```
    mascaras = []
```

```
    # Procesar las imágenes y máscaras
```

```
    archivos_imagenes = [archivo for archivo in os.listdir(ruta_imagenes) if
```

```
archivo.lower().endswith(".jpg")]
```

```
    archivos_mascaras = {archivo.lower(): archivo for archivo in os.listdir(ruta_mascaras) if
```

```
archivo.lower().endswith(".jpg")}
```

```
    for archivo in archivos_imagenes:
```

```
        nombre_mascara = archivos_mascaras.get(archivo.lower())
```

```
        if not nombre_mascara:
```

```
            print(f'No se encontró la máscara correspondiente para {archivo}. Se omite este  
archivo.")
```

```
            continue
```

```
        # Cargar y redimensionar imagen
```

```
        imagen = Image.open(os.path.join(ruta_imagenes, archivo)).convert("RGB")
```

```
        imagen = imagen.resize(tamaño)
```

```
        imagen_array = np.array(imagen, dtype=np.float32) / 255.0 # Convertir a float32
```

```
        imagenes.append(imagen_array)
```

```
        # Cargar y redimensionar máscara
```

**SEGMENTACIÓN DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
mascara = Image.open(os.path.join(ruta_mascaras, nombre_mascara)).convert("L")
mascara = mascara.resize(tamaño)
mascara_array = np.array(mascara, dtype=np.float32) / 255.0
mascara_array = (mascara_array > 0.5).astype(np.float32)
mascaras.append(mascara_array)

imagenes = np.array(imagenes, dtype=np.float32)
mascaras = np.expand_dims(np.array(mascaras, dtype=np.float32), axis=-1)

print(f"Imágenes cargadas: {len(imagenes)}, Máscaras cargadas: {len(mascaras)}")
return imagenes, mascaras

# Probar la función
ruta_imagenes_entrenamiento = r"datos\Entrenamiento\imagenes_procesadas_3"
ruta_mascaras_entrenamiento = r"datos\Entrenamiento\mascaras_procesadas_3"

# Cargar imágenes y máscaras de entrenamiento
X_train_full, y_train_full = cargar_imagenes_y_mascaras(ruta_imagenes_entrenamiento,
ruta_mascaras_entrenamiento)

# Dividir los datos en conjuntos de entrenamiento y validación
X_train, X_val, y_train, y_val = train_test_split(X_train_full, y_train_full, test_size=0.2,
random_state=42)

def iou(y_true, y_pred):
    y_pred = K.cast(y_pred > 0.5, dtype='float32') # Binarización de predicciones
    intersection = K.sum(K.abs(y_true * y_pred), axis=[1, 2, 3])
    union = K.sum(y_true, axis=[1, 2, 3]) + K.sum(y_pred, axis=[1, 2, 3]) - intersection
    iou = K.mean((intersection + K.epsilon()) / (union + K.epsilon()), axis=0)
    return iou

def fl_m(y_true, y_pred):
    y_pred = K.cast(y_pred > 0.5, dtype='float32')

    # Calcular precisión
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())

    # Calcular recall
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())

    # Calcular F1
    fl = 2 * (precision * recall) / (precision + recall + K.epsilon())
```

**SEGMENTACION DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
return f1

# Crear y compilar el modelo UNet
model = unet()

# Compilar el modelo incluyendo IoU y F1-Score como métricas
model.compile(optimizer=Adam(),
              loss=BinaryCrossentropy(),
              metrics=[BinaryAccuracy(), Recall(), iou, f1_m])

# Entrenar el modelo con early stopping
history = model.fit(
    X_train, y_train,
    epochs=500,
    batch_size=8,
    validation_data=(X_val, y_val)
)

# Guardar el modelo entrenado
model.save('./modelos/model500_256_8.h5')

# Crear la carpeta de gráficos si no existe
ruta_graficos = r'./gráficos'
os.makedirs(ruta_graficos, exist_ok=True)

# Evaluar el modelo
loss, accuracy, recall_metric, iou_metric, f1_metric = model.evaluate(X_val, y_val)

print(f'Pérdida: {loss}')
print(f'Precisión: {accuracy}')
print(f'Recall: {recall_metric}')
print(f'IoU: {iou_metric}')
print(f'F1-Score: {f1_metric}')

def guardar_graficos(history):
    # Gráfico de pérdida (loss)
    plt.figure()
    plt.plot(history.history['loss'], label='Pérdida de entrenamiento')
    plt.plot(history.history['val_loss'], label='Pérdida de validación')
    plt.title('Pérdida vs Épocas')
    plt.xlabel('Épocas')
    plt.ylabel('Pérdida')
    plt.legend()
    plt.savefig(os.path.join(ruta_graficos, 'perdida_vs_epocas.png'))
    plt.close()
```

**SEGMENTACIÓN DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
# Gráfico de precisión (binary_accuracy)
if 'binary_accuracy' in history.history and 'val_binary_accuracy' in history.history:
    plt.figure()
    plt.plot(history.history['binary_accuracy'], label='Precisión de entrenamiento')
    plt.plot(history.history['val_binary_accuracy'], label='Precisión de validación')
    plt.title('Precisión vs Épocas')
    plt.xlabel('Épocas')
    plt.ylabel('Precisión')
    plt.legend()
    plt.savefig(os.path.join(ruta_graficos, 'precision_vs_epocas.png'))
    plt.close()

# Gráfico de IoU
if 'iou' in history.history and 'val_iou' in history.history:
    plt.figure()
    plt.plot(history.history['iou'], label='IoU de entrenamiento')
    plt.plot(history.history['val_iou'], label='IoU de validación')
    plt.title('IoU vs Épocas')
    plt.xlabel('Épocas')
    plt.ylabel('IoU')
    plt.legend()
    plt.savefig(os.path.join(ruta_graficos, 'iou_vs_epocas.png'))
    plt.close()

# Gráfico de F1-Score
if 'f1_m' in history.history and 'val_f1_m' in history.history:
    plt.figure()
    plt.plot(history.history['f1_m'], label='F1-Score de entrenamiento')
    plt.plot(history.history['val_f1_m'], label='F1-Score de validación')
    plt.title('F1-Score vs Épocas')
    plt.xlabel('Épocas')
    plt.ylabel('F1-Score')
    plt.legend()
    plt.savefig(os.path.join(ruta_graficos, 'f1_score_vs_epocas.png'))
    plt.close()

guardar_graficos(history)
```

Anexo B: Código para el Procesamiento de Imágenes y Predicciones

A continuación, se presenta el código utilizado para procesar imágenes de prueba y generar predicciones mediante un modelo de red neuronal convolucional. Este código carga las imágenes, realiza las predicciones, y guarda tanto las comparaciones entre las imágenes originales y las predicciones como los resultados de las predicciones individuales.

```
import os
import numpy as np
from PIL import Image, ImageDraw, ImageFont
from tensorflow.keras.models import load_model

# Crear carpetas de salida si no existen
ruta_comparaciones = r"comparaciones"
ruta_resultados = r"predicciones"
os.makedirs(ruta_comparaciones, exist_ok=True)
os.makedirs(ruta_resultados, exist_ok=True)

# Función para cargar todas las imágenes de una vez
def cargar_imagenes(ruta_imagenes, size=(256, 256)):
    archivos_imagenes = [f for f in os.listdir(ruta_imagenes) if f.endswith(".jpg")]
    imagenes = [
        np.array(Image.open(os.path.join(ruta_imagenes, archivo)).convert("RGB").resize(size)) /
        255.0
        for archivo in archivos_imagenes
    ]
    return np.array(imagenes), archivos_imagenes

# Cargar el modelo entrenado
modelo = load_model(r'modelos\model500_256_8.h5', compile=False)

# Definir la ruta de las imágenes de prueba
ruta_imagenes_prueba = r".\datos\Prueba\imagenes"

# Función para combinar imagen original y predicción en una sola imagen
def combinar_imagenes(imagen_original, prediccion):
    # Convertir la imagen original a formato uint8 y escalarla
    imagen_original_uint8 = (imagen_original * 255).astype(np.uint8)

    # Convertir la predicción a formato uint8 (binaria)
    prediccion_uint8 = (prediccion.squeeze() * 255).astype(np.uint8)

    # Convertir ambas imágenes a objetos PIL
    imagen_pil = Image.fromarray(imagen_original_uint8)
    prediccion_pil = Image.fromarray(prediccion_uint8)
```

**SEGMENTACION DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFALTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
# Asegurarse de que ambas imágenes tengan el mismo modo (RGB o L)
if imagen_pil.mode != 'RGB':
    imagen_pil = imagen_pil.convert('RGB')
if prediccion_pil.mode != 'L': # Predicción en escala de grises
    prediccion_pil = prediccion_pil.convert('L')

# Crear una nueva imagen que combine ambas
ancho_total = imagen_pil.width + prediccion_pil.width
alto_maximo = max(imagen_pil.height, prediccion_pil.height) + 30 # Espacio extra para los
títulos

# Crear una imagen vacía donde se combinarán ambas
imagen_combined = Image.new('RGB', (ancho_total, alto_maximo), color=(0, 0, 0))

# Pegar la imagen original y la predicción en la imagen combinada
imagen_combined.paste(imagen_pil, (0, 30)) # Añadir 30px de espacio para el título
imagen_combined.paste(prediccion_pil.convert('RGB'), (imagen_pil.width, 30))

# Añadir títulos a las imágenes
draw = ImageDraw.Draw(imagen_combined)
font = ImageFont.load_default() # Usar fuente por defecto

# Añadir títulos en la parte superior
draw.text((10, 10), "Imagen Original", fill="white", font=font)
draw.text((imagen_pil.width + 10, 10), "Prediccion", fill="white", font=font)

return imagen_combined

# Función para guardar la imagen combinada
def guardar_comparaciones_combinadas(imagen_original, prediccion, nombre_archivo):
    imagen_combined = combinar_imagenes(imagen_original, prediccion)
    imagen_combined_path = os.path.join(ruta_comparaciones,
f'{nombre_archivo}_comparacion.png')
    imagen_combined.save(imagen_combined_path)

# Función para guardar solo la predicción
def guardar_resultado(prediccion, nombre_archivo):
    prediccion_pil = Image.fromarray((prediccion.squeeze() * 255).astype(np.uint8))
    resultado_path = os.path.join(ruta_resultados, f'{nombre_archivo}_resultado.png')
    prediccion_pil.save(resultado_path)

# Procesar todas las imágenes y guardar comparaciones combinadas y resultados
def procesar_imagenes():
    X_test, nombres_imagenes = cargar_imagenes(ruta_imagenes_prueba)
```

**SEGMENTACIÓN DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
predicciones = model.predict(X_test)
predicciones_binarizadas = (predicciones > 0.5).astype(np.uint8)

for i in range(len(X_test)):
    nombre_archivo = os.path.splitext(nombres_imagenes[i])[0]
    guardar_comparaciones_combinadas(X_test[i], predicciones_binarizadas[i],
nombre_archivo)
    guardar_resultado(predicciones_binarizadas[i], nombre_archivo) # Guardar solo la
predicción

print(f'Todas las imágenes han sido procesadas y guardadas en la carpeta de comparaciones y
resultados.')
```

Procesar y guardar las predicciones
procesar_imagenes()

Anexo C: Código para medir las grietas

Este código permite medir el ancho de las grietas en imágenes de pavimentos asfálticos utilizando técnicas de procesamiento de imágenes.

```
import matplotlib.pyplot as plt
import numpy as np
from skimage import io, morphology
from scipy.ndimage import distance_transform_edt

# Cargar la imagen (asegúrate de que esté en escala de grises)
BW = io.imread(r'C:\Users\Luz
Rashell\Desktop\Segmentacion\predicciones\147_1_resultado.png', as_gray=True)

# Asegurarse de que la imagen sea binaria
BW = BW > 0.5

# Obtener el esqueleto
BW_skel = morphology.skeletonize(BW)

# Calcular la distancia del esqueleto a los bordes de las grietas
distancia = distance_transform_edt(~BW_skel)

# Ancho de la grieta en píxeles
ancho_grieta = 2 * distancia

# Convertir de píxeles a milímetros (según tu factor de conversión)
ancho_grieta_mm = ancho_grieta * 0.924

# Crear una imagen de colores para las categorías
resultado = np.zeros((BW.shape[0], BW.shape[1], 3)) # Imagen RGB vacía

# Definir categorías:
# Verde para grietas <= 3 mm, Rojo para grietas > 3 mm
resultado[ancho_grieta_mm <= 3] = [0, 1, 0] # Verde
resultado[ancho_grieta_mm > 3] = [1, 0, 0] # Rojo
```

**SEGMENTACIÓN DE GRIETAS SUPERFICIALES EN PAVIMENTO ASFÁLTICO
UTILIZANDO TÉCNICAS DE VISIÓN ARTIFICIAL - ANEXO
LUZ RASHELL SOTO OLGUIN – 2024**

```
# Mantener el color blanco donde no hay grietas (en la máscara original)
resultado[BW == 1] = [1, 1, 1] # Blanco

# Mostrar el resultado
plt.imshow(resultado)
plt.title('Ancho de las Grietas (verde <= 3mm, rojo > 3mm)')
plt.axis('off') # Ocultar ejes
plt.show()

# Calcular y mostrar el tamaño mínimo de grieta detectada en mm
min_grieta_mm = np.min(ancho_grieta_mm[ancho_grieta_mm > 0]) # Excluyendo ceros
print(f'Tamaño mínimo de grieta detectada: {min_grieta_mm:.2f} mm')
```