INFORME DE TRABAJO 1 (TP)

1ACC0235 – PROCESAMIENTO DE IMÁGENES - VIRTUAL
Carrera de Ciencias de la Computación

Sección: 307
Docente: Carlos Fernando Montoya Cubas

Integrantes:

Código	Apellidos	Nombres
u202316342	Vásquez Trujillano	Edson Fabrizio

2025

ÍNDICE

Descripción del caso de uso

Las vías en mal estado representan un peligro constante para la integridad y seguridad de la población peruana. Según el Observatorio Nacional de Seguridad Vial (2024), en el año 2023, se registraron 1217 siniestros causados por vías en malas condiciones; mientras que el boletín estadístico más reciente del ONSV (2025) registró 1134 siniestros debido a la misma causa en 2024.

Actualmente, la red vial nacional se divide en carreteras concesionadas y no concesionadas, cuyo mantenimiento es realizado por el Ministerio de Transportes y Comunicaciones por medio de dos entidades. Por un lado, el mantenimiento de las carreteras concesionadas está bajo la responsabilidad de la Dirección General de Proyectos de Transportes y es financiado por el pago de peajes. Por otro lado, el mantenimiento de las carreteras no concesionadas depende de empresas contratistas.

Según Saenz (2024), el modelo de contratación es ineficiente puesto que cada contrato con duración de cinco años debe estar respaldado por un perfil técnico previo, el cual puede tardar hasta dos años en realizarse y aprobarse. En consecuencia, no es posible mantener una programación efectiva de cinco años y retrasa el mantenimiento de las carreteras no concesionadas, mientras que el deterioro de estas sigue en aumento.

En ese sentido, es fundamental desarrollar planes, técnicas y métodos que permitan detectar vías en mal estado de forma rápida y eficiente, con el fin de que las vías sean debidamente reparadas en una menor cantidad de tiempo y, por ende, reducir el riesgo de siniestros vehiculares.

Una empresa con resultados positivos ante esta problemática es ASIMOB, la cual ha implementado dispositivos implementados con una inteligencia artificial entrenada para detectar cambios o deterioros en señales de tránsito o vías vehiculares. Según Gil (2023), la circulación diaria de estos dispositivos en las carreteras de Madrid ha generado resultados más eficientes y menos costosos en comparación a los obtenidos por medio de la monitorización humana.

Teniendo en cuenta la experiencia de ASIMOB, proponemos dos soluciones ante las carreteras en malas condiciones. En primer lugar, una solución orientada al uso de técnicas de procesamiento de imágenes con el objetivo de identificar vías que presentan deterioro visible. En segundo lugar, una solución enfocada al uso de un modelo basado en redes profundas para clasificar imágenes de vías en mal estado. En ambos casos, se plantea el uso de cámaras instaladas en vehículos que recorran rutas específicas a lo largo de la red vial nacional. Estas cámaras se encargarán de capturar imágenes de las vías recorridas de manera continua, las cuales serán analizadas posteriormente para identificar baches, huecos profundos, deformaciones, fisuras, entre otros tipos de daños que puedan desembocar en un accidente vehicular.

Descripción del conjunto de datos (dataset)

Descripción del Conjunto de Datos: "Cracks and Potholes in Road Images"

El conjunto de datos **"Cracks and Potholes in Road Images"** fue creado con el propósito de facilitar estudios sobre la detección automática de defectos en pavimentos asfaltados, específicamente grietas (*cracks*) y baches (*potholes*), mediante técnicas de procesamiento de imágenes y algoritmos de aprendizaje automático. Link de acceso: https://biankatpas.github.io/Cracks-and-Potholes-in-Road-Images-Dataset/

Características de los datos

- **Total de imágenes:** 2,235 imágenes seleccionadas manualmente.
- **Formato:** El dataset cuenta con una carpeta para cada foto capturada, dentro de la cual se observan **tres imágenes en formato PNG** que señalan:
    - Imagen original de la carretera.
    - Máscara binaria que registra grietas (cracks) presentes.
    - Máscara binaria que registra baches (potholes) presentes.
- **Resolución:** Las imágenes poseen una resolución de 640 x 1024 píxeles.
- **Condiciones de selección:**
    - No deben contener vehículos ni personas.
    - No deben presentar defectos visuales como colores alterados o partes faltantes.

Origen de los datos

- **Fuente:** Las imágenes fueron proporcionadas por el **Departamento Nacional de Infraestructura de Transportes de Brasil (DNIT)**, mediante la Ley de Acceso a la Información (Protocolo 50650.003556/2017-28).
- **Regiones:** Las imágenes provienen de carreteras ubicadas en los estados de:
    - **Espírito Santo:** BR 101, 259, 262, 393, 447, 482 y 484.
    - **Rio Grande do Sul:** BR 101, 290 y 386.
    - **Distrito Federal:** BR 010, 020, 060, 070, 080 y 251.
- **Período de captura:** Entre los años **2014 y 2017**.

- **Medio de captura:** Las imágenes fueron extraídas de videos capturados por un **Vehículo de Diagnóstico de Carreteras (HDV)** equipado con:
    - Cámara de alta resolución (imagen cada 5 metros).
    - Cámaras de video frontal y trasera (30 FPS, resolución mínima de 1280x729).
    - Sensores láser, odómetro de precisión y sistema de geolocalización satelital.

Analítica del conjunto de datos

Con el objetivo de maximizar el entrenamiento de modelos, se procedió a analizar el conjunto de datos, a fin de encontrar cualquier irregularidad. Según los resultados, se realizó dos observaciones:

- **Se encontró desbalance de clases en el conjunto de datos**
  Ciertamente el conjunto de imágenes cuenta con 2235 fotografías de carreteras, al igual que sus máscaras binarias; sin embargo, al recorrer el conjunto de máscaras binarias, se observan los siguientes resultados:}
    - **Total del conjunto de imágenes**: 2235
    - **Máscaras con potholes**: 564 (incluye las que posean cracks)
    - **Máscaras con cracks**: 1921 (incluye las que posean potholes)
    - **Máscaras con ambas fallas**: 252

  Esto indica que existen más imágenes con presencia de grietas que imágenes con baches. Por ello, se procede a realizar una preselección de imágenes, a fin de balancear ambas clases. Se obtuvieron los siguientes resultados:
    - **Total seleccionado**: 1005 imágenes
    - **Máscaras con solo potholes**: 312
    - **Máscaras con solo cracks**: 441
    - **Máscaras con ambas fallas**: 252

- **Forma y estructura de las grietas**
  Se observa en base a las máscaras binarias, que en muchos casos, las grietas presentan regiones de píxeles más finas y dispersas, lo cual puede generar pérdida en información durante la transformación de las imágenes previamente al entrenamiento del modelo. Por ejemplo:

Figura1: Ejemplo de máscara binaria con etiqueta "crack"

La imagen mostrada cuenta con píxeles regiones pequeñas de píxeles dispersos que pueden llegar a perderse durante la transformación de las imágenes para aumentar la variabilidad. En ese sentido, se tendrá en cuenta dos factores en la modelización basada en redes neuronales:

- Reducir la pérdida de información si se aplica transformaciones
- Dar mayor peso en el entrenamiento a la clasificación de "crack"

Propuesta de modelización

Para abordar la problemática de las vías en mal estado, que incrementan de forma directa el riesgo de siniestros de tránsito, propongo dos estrategias de procesamiento de imágenes basadas en dos enfoques distintos: un enfoque tradicional (siguiendo el contenido inicial del curso) y un enfoque moderno (basado en aprendizaje automático e inteligencia artificial).

1. Enfoque tradicional

    - Conversión a escala de grises: permite centrar el análisis en variaciones de intensidad, de forma que sea posible distinguir baches (potholes) y grietas (cracks) en las imágenes.

    - Umbralización: basándose en las variaciones de intensidad, separa las zonas de interés (grietas y baches) del fondo de la vía o carretera.

    - Detección de contornos: extrae el contorno de las regiones umbrales, facilitando la detección de las fallas.

- Filtrado por área, forma y textura: permite descartar “falsos positivos”, por ejemplo, líneas de cebra, y conservar únicamente aquellas predicciones que coinciden con las características de una falla estructural.

- Comparación con máscaras pothole y crack: considerando que el dataset cuenta con imágenes para comprobar la eficiencia de modelos, estas son de utilidad para corroborar la precisión o eficiencia del algoritmo. Por ejemplo, se puede aplicar puntuación IoU (Intersection over Union) para comparar la predicción realizada por el algoritmo y una máscara resultante que una las máscaras reales de “potholes” y “cracks”.

2. Enfoque moderno (IA / Deep Learning) por medio de Segmentación Semántica

    - Entrenar un modelo de redes neuronales enfocado a la visión computacional. En ese sentido, propongo el entrenamiento de un modelo U-NET para aplicar segmentación semántica al conjunto de datos.

    - El modelo U-NET permite etiquetar pixel por pixel de una imagen en una o más clases. De esta forma, el modelo será entrenado por medio de las máscaras binarias de “pothole” y “cracks” para maximizar sus predicciones.

    - El modelo entrenado tendrá la capacidad de predecir y mostrar dos máscaras binarias. La primera, es una máscara que contenga la predicción de píxeles etiquetados como “potholes”. En cambio, la segunda, contendrá la predicción de los píxeles etiquetados como “cracks”.

    - Finalmente, el resultado será visualizado a fin de comparar la eficiencia del modelo.

Se tendrá en cuenta una o ambas propuestas de modelización, a fin de cumplir con el objetivo de detectar baches y grietas en las carreteras o vías que puedan generar siniestros vehiculares.

Modelización

La propuesta seleccionada fue el uso de U-NET para aplicar Segmentación Semántica al conjunto de imágenes. Se describen los siguientes pasos abordados durante la modelización:

1. Preparación previa al entrenamiento

    - Se realizó el análisis del conjunto de datos para determinar observaciones que puedan influir en el entrenamiento del modelo.
    - Se creó la clase “RoadAnomalyDataset”, la cual recibe la ruta del conjunto de datos, la función de transformación de imágenes y el número límite de imágenes con “cracks” que se procesarán (balance entre clases). Esta clase se encarga de cargar las imágenes desde la ruta entregada, seleccionar las imágenes para balancear las clases y

aplicar la función de transformación a cada imagen para generar un dataset procesado.
    - Se establece la función de transformación “transform”, la cual aplica escalado, inversión horizontal, ajuste de brillo, desenfoque, deformación y normalización. Asimismo, evita la interpolación de píxeles para evitar la pérdida de información. Esta función retorna la imagen en formato tensor de Pytorch para que sea ingresado al modelo.
    - Se obtiene el dataset procesado en base a la clase “RoadAnomalyDataset” y la función “transform”.
    - Por último, se establece el uso de GPU para aumentar la eficiencia del entrenamiento.

2. Separar datos de entrenamiento y de prueba

    - A partir del dataset, se separa 80% de los datos para entrenamiento y el resto para prueba.
    - Se establecen los cargadores de datos para entrenamiento y prueba, en base a los recursos de mi dispositivo (GPU de 4 GB).
    - Se guardarán los índices de las imágenes para entrenamiento y prueba en archivos JSON, a fin de poder ser utilizados para reproducir el entorno.
    - Se genera una carpeta con las imágenes utilizadas para el entrenamiento y las pruebas. Funciona como dataset final y permite reproducir el entorno.
    - Por último, se establece un código para cargar directamente las imágenes seleccionadas desde la carpeta generada en el paso anterior.

3. Evaluación del entrenamiento

    - Se genera una clase “BCEDiceLoss” dedicada a calcular la pérdida BCE y la pérdida DICE, combinando ambas para generar una métrica de pérdida robusta que permita al modelo entrenar y aprender.
        - BCE (Binary Cross-Entropy)
          Es una puntuación que mide las diferencias entre el contenido de las máscaras reales y las máscaras predichas. Solo permite calcular una clase a la vez y es más eficiente con clases balanceadas. En este caso, se aplica a cada clase para determinar la pérdida por clase.
        - Dice Loss (Dice Coefficient)
          Es una puntuación similar a IoU en el sentido de que busca calcular la similitud entre máscaras reales y predichas en base a la intersección y unión de ambas. En el proyecto, se calcula por cada clase, a fin de combinarse con la pérdida BCE, la cual tiene como principal desventaja su menor eficiencia cuando el fondo (clase fondo) es mucho mayor a la clase analizada (como baches o grietas). En ese sentido, Dice se propone como un medio para solventar esa desventaja.
    - Se genera una función para aplicar “BCEDiceLoss” a todo el lote, calculando la pérdida promedio.
    - Evaluación por IoU, es decir, se establece la función “iou_score” la cual calcula píxel por píxel la intersección y la unión entre la máscara predicha y la máscara original para obtener una métrica de eficiencia de entrenamiento para segmentación semántica. Retorna la puntuación IoU promedio del lote.

- Por último, se ajusta el modelo U-NET mediante codificador “resnet18” y pesos pre entrenados de “imagenet”. Asimismo, se selecciona el optimizador Adam.

4. Entrenamiento

    - Se establece el número de épocas a 100.
    - Se establece una lista “history” que guarda los resultados de las métricas por época.
    - Se establece “best_test_loss” y “best_mean_iou” para guardar la menor pérdida actual y la mayor puntuación IoU actual, respectivamente.
    - Se realiza el entrenamiento con una barra de carga visual, considerando lo siguiente:
        - Las imágenes y máscaras se envían a GPU para mejorar el rendimiento.
        - Se realizan las predicciones e inmediatamente se calcula la pérdida.
        - La pérdida de BCEDiceLoss sirve para backpropagation.
        - Se muestra el cálculo final de la pérdida por lote de entrenamiento.
        - Se evalúa el modelo con el dataset de prueba (Test) y se calcula la pérdida.
        - Se muestra la pérdida en base a “Test”.
        - Se calcula la puntuación IoU para cada clase, se imprime y se calcula el promedio entre el resultado de ambas clases.
        - Se guardan los resultados en “history”.
        - Se guarda el mejor modelo basado en la menor pérdida.
        - Se guarda el mejor modelo basado en la puntuación IoU.

5. Resultados del modelo

    - Se establece la función “visualizar_prediccion” para mostrar la imagen real, las máscaras reales de “pothole” y “crack”, y las máscaras predichas de ambas clases.

6. Gráficas de pérdida y puntuación IoU por época

    - Se establece la función “mostrar_Graficas”, la cual recibe la lista “history” que contiene todas las métricas por época obtenidas durante el entrenamiento. Esta función muestra dos gráficas que incluye la pérdida calcula por época y la puntuación IoU calculada por época en base al modelo seleccionado.
    - Se procede a aplicar “mostrar_Graficas” a los dos modelos guardados: “best_model_iou.pth” y “best_model_loss.pth”.
    - Finalmente, se guarda la lista “history” en un archivo JSON para reutilización en futuras ejecuciones.

Publicación de los resultados

1. Mejores modelos

    - El mejor modelo basándose en la puntuación IoU se alcanzó en la época 76.

      🌱 Epoch 76/100 - Train Loss: 1.3837 | Test Loss: 1.5448
      IoU Pothole: 0.6639 | IoU Crack: 0.4211
      ✅ Mejor modelo IoU actualizado (IoU Promedio: 0.5425)

- El mejor modelo basándose en la pérdida total se alcanzó en la época 65.

  🌱 Epoch 65/100 - Train Loss: 1.3293 | Test Loss: 1.4026
  IoU Pothole: 0.6383 | IoU Crack: 0.4287
  ✅ Mejor modelo Loss actualizado (Test Loss: 1.4026)

2. Gráfica de pérdida y puntuación IoU por época para ambos modelos

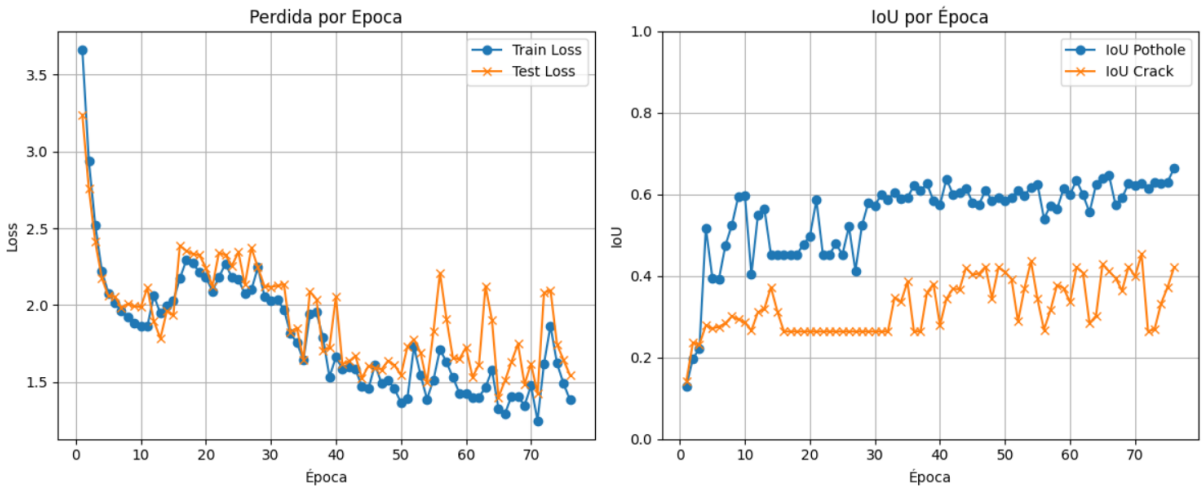


Figura 2: Gráfica de pérdida por época (modelo con mayor puntuación IoU)

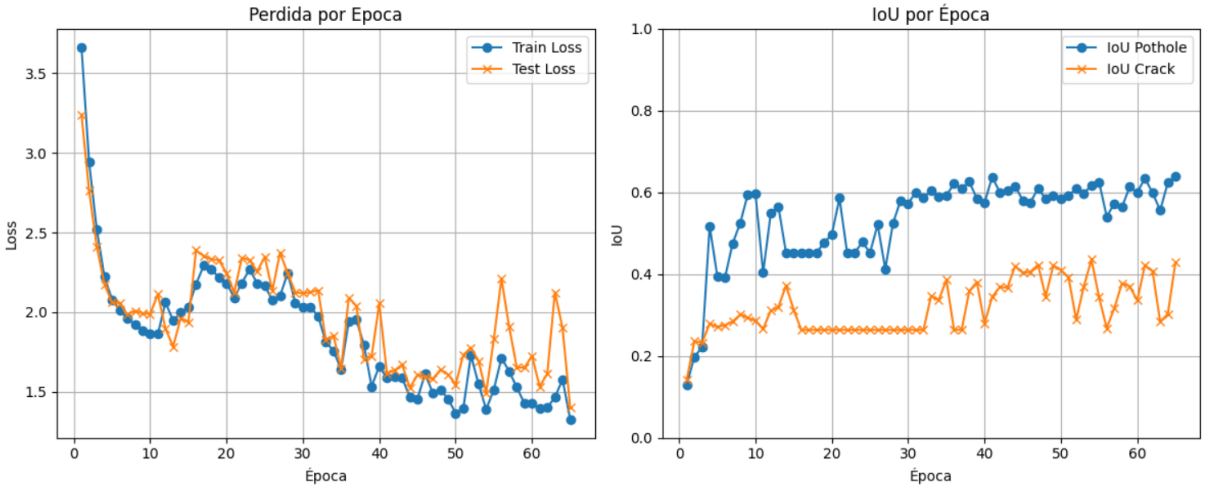


Figura 3: Gráfica de pérdida por época (modelo con menor pérdida)

Ambas figuras muestran gráficas con el mismo historial de épocas, diferenciándose en la última época, la cual indica el momento en que se obtuvo su mejor métrica. En base a ambas figuras, se puede observar lo siguiente:

- Las curvas de pérdida de “train” y “test” tienden a disminuir con el tiempo. Esto indica que el modelo está aprendiendo.

- La curva de pérdida de “train” y “test” muestran picos de subida y bajada a partir de la época 50. Esto podría indicar que el modelo está comenzando a sobre ajustarse, principalmente si los picos de la pérdida de “Test” siguen en aumento, tal como se puede apreciar a partir de la época 70. En ese sentido, el modelo basado en la menor pérdida se obtiene antes de los indicios de un posible sobre ajuste o de un ruido.

- Se observa que a partir de la época 40, la curva de pérdida tanto de “train” como de “test” comienza a estabilizarse. Esto puede ser un indicio de que el modelo se acerca a su punto de convergencia, es decir, que está pronto a dejar de mejorar significativamente. Esta idea se refuerza con los picos generados a partir de la época 50.

- La curva de puntuación IoU para “Potholes” tiende a incrementar con el tiempo. Lo cual indica que el modelo aprende a detectar baches (potholes).

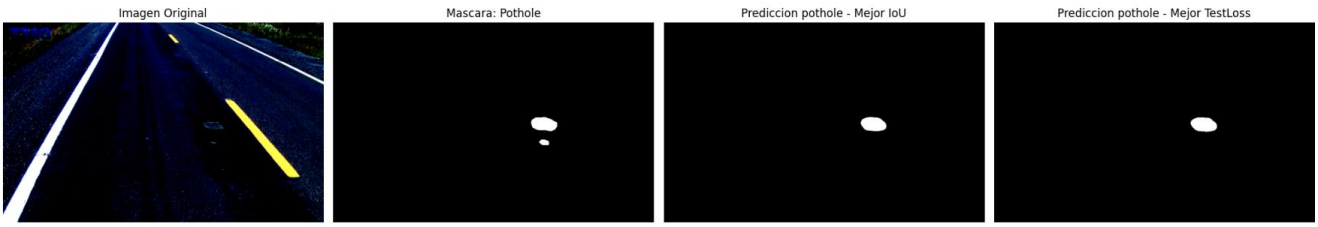
- Esta misma curva se estabiliza alrededor del valor 0.6 (aceptable) a partir de la época 30, lo cual indica que muy probablemente está por alcanzar su punto de convergencia.

- La curva de puntuación IoU para “Cracks” incrementa al inicio del entrenamiento. Sin embargo, alrededor de la época 15, la curva tiende a volverse inestable. Esto indica problemas en el aprendizaje de las máscaras “Crack”.

- Durante las 15 a 30, la curva de puntuación IoU para “Cracks” muestra una meseta, indicando un largo tramo en el cual el modelo no aprendió o aprendió muy poco sobre esta clase.

- La curva de puntuación IoU para “Cracks” en general es más baja que la de “Potholes”, lo cual indica que el modelo aprende mejor a detectar baches (potholes) que grietas (cracks).

3. Ejemplos de predicciones



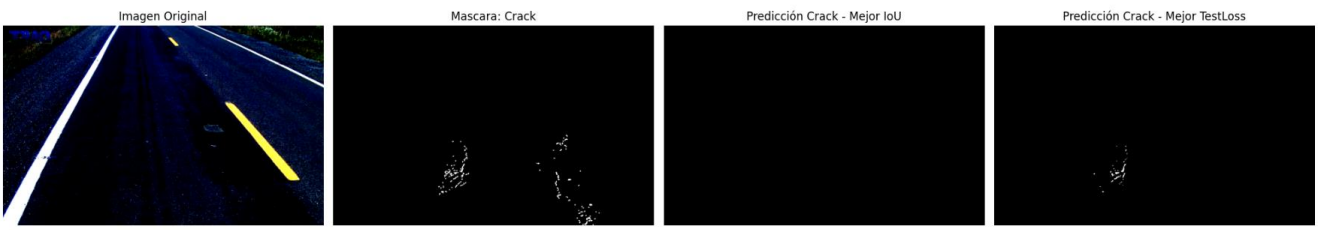


Figura 4: Predicción de una imagen del dataset - 1

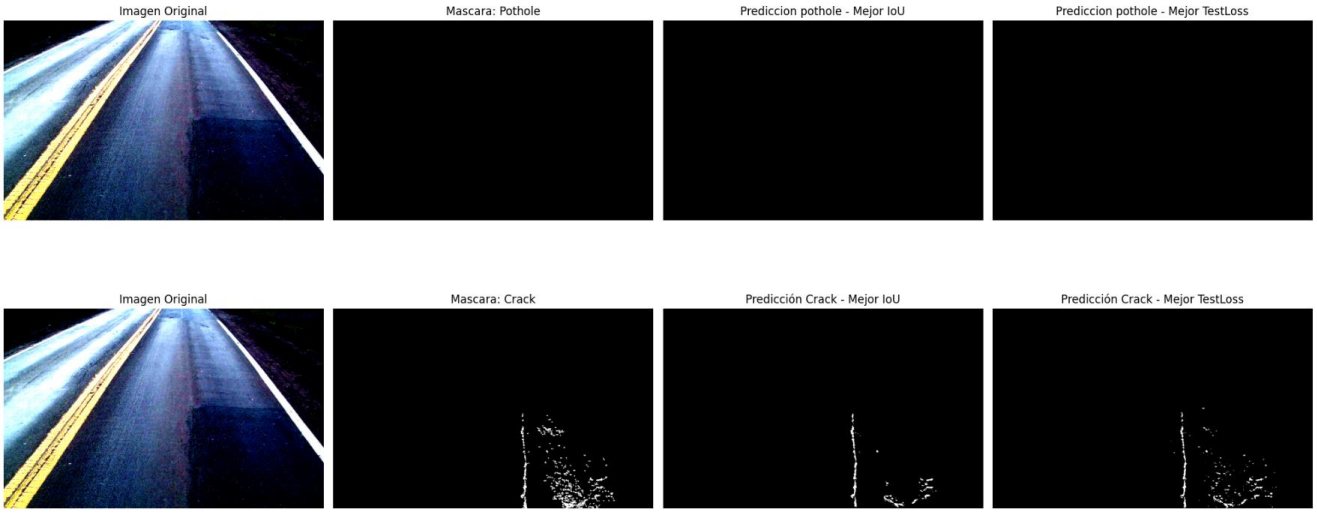


Figura 5: Predicción de una imagen del dataset - 2

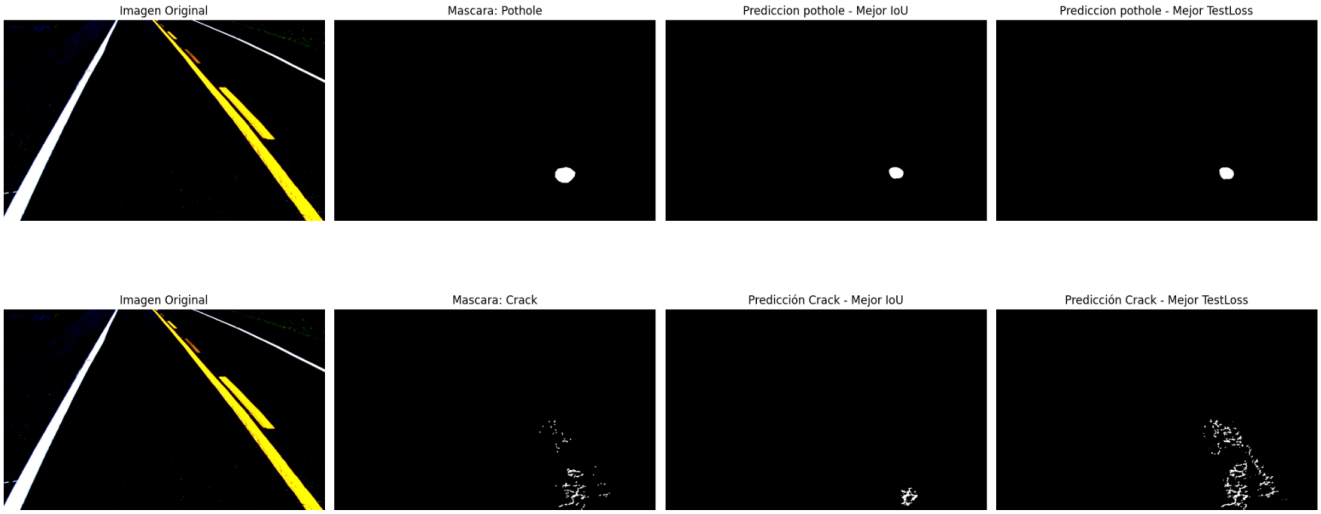


Figura 6: Predicción de una imagen del dataset - 3

A lo largo de las figuras 4, 5 y 6, se observa que el modelo tiene una capacidad aceptable para distinguir baches (potholes); sin embargo las predicciones de grietas (cracks) son inestables. En la figura 1, el modelo basado en la menor pérdida fue capaz de predecir una pequeña cantidad de las grietas; en cambio, el modelo con mayor puntuación IoU no predijo ninguna. En la figura 2, ambos modelos fueron capaces de predecir en gran medida las grietas, siendo el modelo con menor pérdida el cual dio mejor resultado a vista humana.
En la figura 3, ambos modelos fueron capaces de predecir grietas, pero de forma distinta. El modelo con menor pérdida predijo grietas de más, es decir, ha reconocido parte de la carretera en buen estado como grieta. Por otro lado, el modelo con mayor puntuación IoU solo pudo predecir una parte de las grietas.

Conclusiones

El modelo U-NET estándar, con un codificador ResNet18 pre entrenado en ImageNet, alcanzó un desempeño aceptable en la segmentación semántica de baches (potholes); sin embargo, presentó predicciones inestables para la detección de grietas (cracks). Mostró una tendencia a reducir las pérdidas durante el entrenamiento, lo cual evidencia un aprendizaje efectivo a partir de los datos del entrenamiento. Considero que la mejor versión del modelo se obtuvo en la época con menor pérdida de validación (TestLoss) en comparación a aquella con la mayor puntuación IoU, ya que la primera logró un mejor valor IoU para la clase "Crack" (0.4287) frente a la segunda (0.4211). Además, esta versión presentó predicciones más precisas en la visualización de las máscaras generadas.

Como recomendación para futuros trabajos, sugiero aplicar ajustes destinados a reducir la inestabilidad en la predicción de grietas, tales como: asignar mayor peso en el cálculo de pérdidas de la clase "Crack", explorar otras arquitecturas más especializadas o versiones más avanzadas de U-NET, e incrementar la variabilidad en las imágenes que contengan grietas.

Referencias bibliográficas

Darth Espressius. (29 de enero de 2022). 3 Common Loss Functions for Image Segmentation. [Entrada en blog]. https://dev.to/_aadidev/3-common-loss-functions-for-image-segmentation-545o

DataScientest (26 de abril de 2025). U-NET: todo lo que tienes que saber sobre la red neuronal de Computer Vision. [Entrada en blog]. https://datascientest.com/es/u-net-lo-que-tienes-que-saber

Gil, A. (22 de diciembre de 2023). Carreteras vigiladas por inteligencia artificial para tenerlo todo bajo control. EL MOTOR. https://motor.elpais.com/tecnologia/asimob-inteligencia-artificial-carretera-bajo-control/

Observatorio Nacional de Seguridad Vial (2024). Boletín estadístico de siniestralidad vial, 2023. https://www.onsv.gob.pe/post/boletin-estadistico-de-siniestralidad-vial-2023/

Observatorio Nacional de Seguridad Vial (2025). Boletín estadístico de siniestralidad vial, 2024.https://www.onsv.gob.pe/post/boletin-estadistico-de-siniestralidad-vial-2024/

Saenz, M. (16 de julio de 2024). Carreteras olvidadas: El precio de la negligencia en la infraestructura vial peruana [INFORME]. *RPP*. https://rpp.pe/economia/economia/cual-es-el-estado-de-las-pistas-y-carreteras-en-el-peru-informe-noticia-1557854?ref=rpp#google_vignette