



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

DETECCIÓN Y CLASIFICACIÓN DE GRIETAS EN ASFALTO MEDIANTE  
REDES CONVOLUCIONALES Y PROCESAMIENTO DE IMÁGENES

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELÉCTRICO

CLAUDIO ISAÍAS URBINA LARA

PROFESOR GUÍA:  
JOSÉ SAAVEDRA RONDO

MIEMBROS DE LA COMISIÓN:  
GONZALO SANDOVAL PALMA  
ANDRÉS CABA RUTTE

Este trabajo ha sido parcialmente financiado por APSA Ltda.

SANTIAGO DE CHILE  
2020

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO  
POR: CLAUDIO ISAÍAS URBINA LARA  
FECHA: 2020  
PROF. GUÍA: JOSÉ SAAVEDRA RONDO

## DETECCIÓN Y CLASIFICACIÓN DE GRIETAS EN ASFALTO MEDIANTE REDES CONVOLUCIONALES Y PROCESAMIENTO DE IMÁGENES

Los avances tecnológicos, sociales y demográficos demandan garantizar seguridad y calidad en infraestructuras como carreteras y calles. La detección y análisis de grietas es un problema que necesita ser resuelto cada vez con mayor velocidad y precisión debido a que es necesario determinar cuándo y cómo abordar las grietas presentes en miles de kilómetros. El problema a abordar en la memoria contempla las etapas de detección, clasificación y análisis de severidades de grietas presentes en imágenes de carreteras de asfalto capturadas por el equipo Laser Crack Measurement System (LCMS) en Chile. Para la primera etapa se presentan dos propuestas; una basada en técnicas de procesamiento de imágenes de bajo nivel y otra con redes convolucionales. Para la segunda, se presenta una propuesta basada en análisis topológico de componentes conectadas y otro de redes convolucionales. Finalmente, para la tercera, se propone un algoritmo para medir el ancho y severidad de las grietas.

Para la etapa de detección, la primera propuesta se basó en la aplicación en serie de distintas técnicas de procesamiento de imágenes de bajo nivel, tales como; *TV Chambolle*, *CLAHE*, Histéresis, entre otros. La segunda propuesta se basó en la utilización de la red *pix2pix*, que fue entrenada con la base de datos CRACK256, creada para objetivos de la memoria. De estas propuestas, la segunda es la que presenta mejores resultados, obteniendo un coeficiente de *dice* de 0,81 y un tiempo promedio de ejecución de 1,58 [s].

Para la etapa de clasificación, la primera propuesta se basó en el análisis topológico de las grietas detectadas, mientras que, la segunda es un modelo convolucional de clasificación; entrenada con un conjunto de imágenes etiquetadas para la memoria. La evaluación se realiza utilizando una matriz de confusión para determinar la precisión de ambas propuestas. De las propuestas presentadas, la primera es la que tiene mejor desempeño y rendimiento, obteniendo una precisión promedio de 80 [%] y tiempo promedio de procesamiento de 0,63 [s].

La última etapa, análisis de severidades, es resuelta con un algoritmo que analiza el grosor de las grietas detectadas, el cual fue evaluado por un experto en el área de severidades de grietas, quien determinó que la efectividad del algoritmo es de al menos un 92 [%] y un tiempo promedio de procesamiento 0,22 [s].

Finalmente, se concluye que los problemas abordados en la memoria fueron resueltos satisfactoriamente. Esto, debido a que las métricas de evaluación obtuvieron un gran desempeño tanto en precisión como en tiempo.



*A mi abuelita Gilda.*



# Agradecimientos

Quisiera agradecer en primer lugar a mi familia; mis padres Debora y Roberto, que son quienes me han dado apoyo y amor durante toda la vida, gracias a ellos he podido lograr cada uno de los objetivos que me he propuesto en la vida. También quisiera agradecer a mi nina, que siempre me regalona y apoya como si fuera su hijo. A mi güela Gilda, que siempre apoyaba mis mañas y mis antojos, quien me dio su amor hasta el último día. A mi tata, que me acompañaba y apoyaba en los partidos de futbol cuando era niño, y siempre quiere compartir sus chistes de Condorito conmigo. A mi hermano Cristian, que a pesar de que cuando chicos peleábamos hartos, hoy somos súper unidos y compartimos todos los memes, videos y juegos que encontramos. Gracias a todos por su preocupación y amor constante.

También quisiera agradecer a mi compañera de aventuras, mi amor, Geraldine. Gracias por los años que me hemos disfrutado, las experiencias que me has entregado, las locuras, el apoyo incondicional, y por sobre todo, el amor que me das cada día. Eres fundamental en cada una de las decisiones que tomo a diario; me has ayudado a crecer, mejorar, tener nuevas perspectivas y ser cada vez una persona más completa. Espero seguir creciendo, aprendiendo y compartiendo a tu lado por muchos años más.

A mis tíos, tías, primos y primas, que siempre han confiado en mí y me han apoyado. Gracias por las risas y el amor cada vez que nos vemos.

Quisiera agradecer también a mis mejores amigas Fran y Mely, por todos los años de amistad, risas y apoyo desde que nos conocemos. También a Oscar y Juan Pablo, quienes han sido mis amigos desde kinder, gracias por los momentos que hemos compartido y las tonteras por las que nos reímos siempre. Espero podamos conservar nuestra amistad por muchos años más.

A mis amigos de Electrobogitos, Jean Franco, Malú, Diego, Iván, Feña, Gabo y Vale, por ser el mejor grupo de eléctrica. Gracias por la simpatía, el apañe, las juntas para estudiar y las comidas que compartimos. Agradecimiento especial a JF por todos los ramos que tomamos juntos, las tareas en grupo, las conversaciones, los memes y sobretodo, la amistad.

Finalmente, agradecer a todos quienes me entregaron buenos momentos, conocimiento y experiencias durante mi vida, que me ayudaron a llegar donde estoy.



# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación y Problema . . . . .	1
1.2. Hipótesis . . . . .	3
1.3. Objetivos . . . . .	3
1.3.1. Objetivo General . . . . .	3
1.3.2. Objetivos Específicos . . . . .	3
1.4. Estructura de la Memoria . . . . .	4
<b>2. Marco Teórico y Estado del Arte</b>	<b>5</b>
2.1. Procesamiento de Imágenes . . . . .	5
2.1.1. Procesamiento de Imágenes de bajo nivel . . . . .	5
2.1.2. Redes Convolucionales . . . . .	14
2.1.3. Métricas de Evaluación . . . . .	18
2.1.4. Detección de Grietas mediante procesamiento de imágenes .	20
2.2. Aspectos generales de caminos, deterioros y mediciones . . . . .	23
2.2.1. Componentes de una carretera . . . . .	23
2.2.2. Clasificación y Severidad de Grietas . . . . .	24
2.2.3. Laser Crack Measurement System . . . . .	25
<b>3. Desarrollo</b>	<b>28</b>
3.1. Formalización del problema . . . . .	28



3.2.	Propuestas . . . . .	28
3.3.	Bases de datos utilizadas . . . . .	29
3.3.1.	Imágenes <i>Range</i> e <i>Intensity</i> . . . . .	29
3.3.2.	Base de Datos para Etapa de Detección . . . . .	30
3.3.3.	Base de Datos para Etapa de Clasificación . . . . .	32
3.4.	Pre-Procesamiento . . . . .	33
3.5.	Etapa de Detección . . . . .	34
3.5.1.	Procesamiento de Imágenes . . . . .	35
3.5.2.	Red <i>pix2pix</i> . . . . .	36
3.6.	Etapa de Clasificación . . . . .	37
3.6.1.	Análisis de Componentes Conectadas . . . . .	37
3.6.2.	CNN . . . . .	40
3.7.	Etapa de Análisis de Severidades . . . . .	41
3.8.	Post-Procesamiento . . . . .	41
3.8.1.	Visualización . . . . .	41
<b>4.</b>	<b>Resultados y Discusión</b>	<b>43</b>
4.1.	Resultados . . . . .	43
4.1.1.	Etapa de Detección . . . . .	43
4.1.2.	Etapa de Clasificación . . . . .	46
4.1.3.	Etapa de Análisis de Severidades . . . . .	49
4.2.	Discusión . . . . .	52
4.2.1.	Etapa de Detección . . . . .	52
4.2.2.	Etapa de Clasificación . . . . .	56
4.2.3.	Análisis de Severidades . . . . .	59
<b>5.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>60</b>

5.1. Conclusiones . . . . .	60
5.2. Trabajo Futuro . . . . .	62

<b>Bibliografía</b>	<b>64</b>
---------------------	-----------

# Índice de Tablas

2.1. Arquitectura del generador <i>pix2pix</i> . . . . .	18
2.2. Arquitectura del discriminador <i>pix2pix</i> . . . . .	18
2.3. Ejemplo de matriz de confusión. . . . .	20
3.1. Resumen para conjuntos CRACK10M y CRACK256. . . . .	31
3.2. Resumen de imágenes utilizadas para clasificación en CNN. . . . .	32
3.3. Detalle de estructura de CNN para clasificación. . . . .	40
3.4. Definición de severidades en milímetros y píxeles. . . . .	41
3.5. Colores para cada clase y severidad. . . . .	42
4.1. <i>Dice Coefficient</i> en etapa de detección. . . . .	44
4.2. Tiempo en etapa de detección. . . . .	44
4.3. Matriz de Confusión para clasificación obtenida con Análisis de CCO. . . . .	46
4.4. Matriz de Confusión para clasificación obtenida con CNN. . . . .	46
4.5. Tiempo en etapa de clasificación. . . . .	47
4.6. Tiempo en etapa de análisis de severidades. . . . .	50
4.7. Comparación en tiempos de procesamiento en imágenes con poca y muchas bifurcaciones. . . . .	57

# Índice de Ilustraciones

2.1.	Ejemplo de aplicación de <i>Total Variation Denoising</i> . . . . .	6
2.2.	Ejemplo de histograma de una imagen. . . . .	7
2.3.	Histograma e histograma acumulado pre-ecualización para la imagen Fig 2.2a. . . . .	7
2.4.	Imagen y sus respectivos histograma e histograma acumulado post-ecualización para la imagen Fig 2.2a . . . . .	8
2.5.	Redistribución de valores en el histogramas posterior al <i>clipping</i> . . . . .	9
2.6.	Imagen y sus respectivos histograma e histograma acumulado post-ecualización de histogramas adaptativo con limitación de contraste para la imagen Fig 2.2a. . . . .	9
2.7.	Visualización de binarización mediante threshold en el histograma. . . . .	12
2.8.	Resultados al aplicar Thresholding o Binarización de imagen. . . . .	12
2.9.	Arquitectura de red AlexNet [14]. . . . .	15
2.10.	Funcionamiento CNN. . . . .	15
2.11.	Ejemplo de Segmentación Semántica [16]. . . . .	16
2.12.	Arquitectura U-Net [17]. . . . .	17
2.13.	Arquitectura <i>pix2pix</i> [19]. . . . .	18
2.14.	Imágenes de ejemplo para métrica <i>dice coefficient</i> . . . . .	19
2.15.	<i>Connected components linkage results. Each pair of rows (from left to right and top to bottom) represents: original image; segmentation result; the connected components identified as relevant; the final crack regions detection after the proposed linkage</i> [23]. . . . .	21
2.16.	Resultados Experimentales: (a) <i>Captured image (left column, from top to bottom); (b) CrackForest (the second column); (c) Normal Otsu processed image (the third column); (d) Improved Otsu processed image (right column)</i> [24]. . . . .	21
2.17.	<i>Sample crack segmentation results: initial crack seeds (left) and after crack segmentation refinement (right).</i> [25]. . . . .	22
2.18.	Experimental Results [26]. . . . .	22
2.19.	<i>Comparative results on static images. Row1: original images (the last without crack), row2: CWT Markov method, row3: Threshold, row4: FFA</i> [27]. . . . .	23
2.20.	Ejemplo de grieta de fatiga [28]. . . . .	24
2.21.	Ejemplo de grieta longitudinal [28]. . . . .	24
2.22.	Ejemplo de grieta transversal [28]. . . . .	25
2.23.	Esquema de información obtenida de los láseres del LCMS. . . . .	26

2.24. Posibles detecciones a partir del láser <i>intensity</i> . . . . .	26
2.25. Grieta observada a partir de láser <i>range</i> . . . . .	27
3.1. Ejemplo de imágenes <i>range</i> e <i>intensity</i> de la base de datos. . . . .	30
3.2. Ejemplo de imagen de entrada y salida para base de datos CRACK10M. . . . .	31
3.3. Ejemplo de aplicación de Data Augmentation en sub-imagenes de 256x256 [px]. . . . .	32
3.4. Filtración de delimitaciones vial e información irrelevante. . . . .	34
3.5. Esquema de Etapa de Detección. . . . .	34
3.6. Esquema de técnicas utilizadas en la etapa de Procesamiento de Imágenes. . . . .	35
3.7. Esquema de técnicas utilizadas en la etapa de Detección. . . . .	36
3.8. Esquema de entrenamiento de red <i>pix2pix</i> . . . . .	37
3.9. Ejemplo de funcionamiento de red <i>pix2pix</i> . . . . .	37
3.10. Ejemplo para condición 1. (a) Ejemplo de ventana para pixel $(x_p, y_p)$ que no es bifurcación, (b) Ejemplo de ventana para pixel $(x_p, y_p)$ que sí es una bifurcación. . . . .	38
3.11. Ejemplo de intersecciones posibles al trazar un círculo centrado en un pixel $(x_p, y_p)$ que corresponde a una bifurcación [29]. . . . .	39
3.12. Análisis de severidad dada la distancia de entre pixel de grieta y pavimento. . . . .	41
4.1. Ejemplo 1. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos Procesamiento de Imágenes, (d) Resultados <i>pix2pix</i> . . . . .	44
4.2. Ejemplo 2. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos Procesamiento de Imágenes, (d) Resultados <i>pix2pix</i> . . . . .	45
4.3. Ejemplo 3. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos Procesamiento de Imágenes, (d) Resultados <i>pix2pix</i> . . . . .	45
4.4. Ejemplo 4. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos Procesamiento de Imágenes, (d) Resultados <i>pix2pix</i> . . . . .	46
4.5. Ejemplo 1. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos CNN, (d) Resultados Análisis de CCO. . . . .	47
4.6. Ejemplo 2. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos CNN, (d) Resultados Análisis de CCO. . . . .	48
4.7. Ejemplo 3. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos CNN, (d) Resultados Análisis de CCO. . . . .	48
4.8. Ejemplo 4. (a) Imagen <i>range</i> , (b) Detección Esperada, (c) Resulta- dos CNN, (d) Resultados Análisis de CCO. . . . .	49
4.9. Ejemplo 1. (a) Imagen <i>range</i> , (b) Grietas presentes en (a), (c) Re- sultados Análisis de Severidad . . . . .	50
4.10. Ejemplo 2. (a) Imagen <i>range</i> , (b) Grietas presentes en (a), (c) Re- sultados Análisis de Severidad . . . . .	51
4.11. Ejemplo 3. (a) Imagen <i>range</i> , (b) Grietas presentes en (a), (c) Re- sultados Análisis de Severidad . . . . .	51
4.12. Ejemplo 4. (a) Imagen <i>range</i> , (b) Grietas presentes en (a), (c) Re- sultados Análisis de Severidad . . . . .	52

4.13. Ejemplo de detección. (a) Detección esperada, (b) Detección obtenida con Pdl. . . . .	53
4.14. Ejemplo de detección ruidosa. (a) Imagen <i>Range</i> , (b) Detección esperada, (c) Detección obtenida con Pdl. . . . .	53
4.15. Ejemplo de grietas en carreteras con y sin ruido. (a) Imagen <i>Range</i> con ruido, (b) Imagen <i>Range</i> sin ruido. . . . .	54
4.16. Ejemplo de grieta con poca profundidad. (a) Imagen <i>Range</i> , (b) Detección obtenida por la red <i>pix2pix</i> . . . . .	55
4.17. Ejemplo de grieta con alta profundidad. (a) Imagen <i>Range</i> , (b) Detección obtenida por la red <i>pix2pix</i> . . . . .	55
4.18. Ejemplo de detección. (a) Detección esperada, (b) Detección obtenida con <i>pix2pix</i> . . . . .	55
4.19. Ejemplo de detección ruidosa. (a) Imagen <i>Range</i> , (b) Detección esperada, (c) Detección obtenida con <i>pix2pix</i> . . . . .	56
4.20. Ejemplo de diferencias de bifurcaciones. (a) Grietas con pocas bifurcaciones, (b) Grietas con muchas bifurcaciones. . . . .	58
4.21. Ejemplos de problemas al clasificar por área con CNN. (a) Error al identificar distintas grietas, (b) Error al determinar inicio y término de una clase de grieta. . . . .	59



# Capítulo 1

## Introducción

### 1.1. Motivación y Problema

Desde el comienzo de la civilización, el ser humano ha necesitado crear caminos o carreteras como guías para transportarse de un lugar a otro. Ya entre los años 2600 y 2200 a.C., en Egipto, se situó el primer camino pavimentado registrado en la historia [1]. Hoy en día, son el medio de conexión terrestre entre distintas ciudades y localidades en cualquier parte del mundo. Tan solo en Chile, para el año 2017, existían cerca de 82 mil kilómetros de carretera, de las cuales aproximadamente 20 mil kilómetros se encuentran pavimentados con asfalto, hormigón o una mezcla de ambas, mientras que, el resto se dividen en bischofita, ripio y tierra [2].

El compendio del INE del año 2017 [3], registra que casi 400 millones de vehículos fueron controlados tan solo en plazas de peajes en carreteras y cerca de 1.500 millones transitaron a través de pódicos de autopistas urbanas en la Región Metropolitana. Este gran volumen obliga a que la mantención y revisión de carreteras sea cada vez más certero y eficiente, ya sea, en tiempo o en recursos a utilizar.

Dado los avances tecnológicos, sociales y demográficos, es necesario garantizar la seguridad en carreteras, y así, prevenir accidentes automovilísticos de cualquier tipo. Para ello, se deben realizar estudios y revisiones constantes, que incluyen desde verificar la calidad en la pavimentación, hasta analizar las señalizaciones que poseen las mismas (retroreflectancia, disposición, etc.).

Los planes de conservación de carreteras consideran las decisiones a realizar en los pavimentos, garantizando su comportamiento de manera óptima. Para esto, se realizan mediciones que levantan distintos parámetros, entre los que se encuentran ahuellamiento, agrietamiento, regularidad superficial (IRI), entre otros. Con esta información se determina la condición y en consecuencia, las intervenciones a realizar a la ruta.

Uno de los procesos más lentos en revisión y reparación, es el agrietamiento



formado en cualquier tipo de pavimento. Este, necesita ser reportado con exactitud y de manera constante, lo que permite realizar un análisis de como aumentan en el tiempo y si necesitan ser reparadas a la fecha. La información necesaria a obtener de estas revisiones es la **cuantificación** de grietas que presentes en la carretera, la **clase** (transversales, longitudinales, etc.) de estas y que **severidad** poseen.

Es fácil intuir como esta tarea no puede ser realizada por personas, pues, analizar imágenes de kilómetros y kilómetros de carretera con precisión, puede tardar incluso meses.

Dentro de los desafíos que conllevan estas revisiones, se encuentra el ruido provocado por la textura de la pavimentación, lo que dificulta el proceso de distinguir las grietas presentes en una imagen. A pesar de diversos estudios realizados durante los últimos años, cada caso necesita de una implementación distinta, pues, depende tanto de la calidad de las imágenes, como, de la carretera que se esté revisando. Así también, otro desafío es el comportamiento de las grietas en distintas zonas y climas, ya que, en climas áridos suelen taparse con arena, provocando que la cámara no las capte en su totalidad, mientras que, en climas húmedos, tienden a resaltar. Situaciones como las mencionadas implican el diseño de una solución que pueda adaptarse a distintos contextos, sin aumentar los tiempos de procesamiento.

En la actualidad, para obtener estos parámetros se utilizan equipos de alto rendimiento, dentro de los que se encuentra el *Laser Crack Measurement System* (LCMS). Este equipo permite extraer parámetros como: IRI, textura y agrietamiento. El equipo LCMS esta compuesto por dos láseres 3D, capaces de medir de manera transversal una pista de carretera con una precisión de 1[mm] [4].

Si bien la detección de estos equipos es buena, el tiempo de procesamiento puede aumentar considerablemente en algunas zonas, llegando a tardar incluso semanas. Por otra parte, los equipos no poseen una precisión muy alta a la hora de determinar la clase o severidad de una grieta, por consecuencia, se deben realizar inspecciones visuales de los resultados obtenidos por el equipo, y así, corregir errores, utilizando recursos extras.

Desde el punto de vista académico, la mayoría de las investigaciones relacionadas con el análisis de grietas en carreteras se enfocan solo en la detección y no dan relevancia a la severidad o la clase que poseen, lo que impide realizar un análisis completo del agrietamiento presente en carreteras.

Lo mencionado anteriormente, permite observar que a pesar de que existen equipos capaces de obtener imágenes en alta resolución, el procesamiento y análisis de estas no ha sido suficiente, lo que conlleva a perder recursos en obtener un completo análisis de las grietas presentes.

En la presente memoria, se abordarán propuestas de procesamiento de imágenes y *deep learning* para solucionar los problemas de detección, clasificación y análisis de severidades en grietas presentes en imágenes de carreteras de asfalto, obtenidas por el equipo LCMS.

## 1.2. Hipótesis

Para la formulación de las hipótesis se consideraron los siguientes antecedentes de la memoria:

- El problema de detección presenta una amplia diversidad de imágenes posibles de carreteras, ya sea en: textura, forma de grietas, cantidad de ruido, etc.
- El problema de clasificación presenta la posibilidad de grietas de distinta clase que estén interconectadas y/o convivan en un espacio común.
- El problema de análisis de severidad es una medición exacta y no una interpretación.

Bajo estas condiciones, se formulan las siguientes hipótesis:

1. La propuesta de redes convolucionales tendrá un mejor desempeño en la detección de grietas que la propuesta de procesamiento de imágenes de bajo nivel.
2. La propuesta de análisis de componentes conectadas presentará un mejor desempeño en la clasificación de grietas que la propuesta de redes convolucionales de clasificación.
3. Un algoritmo de bajo nivel será suficiente para la solucionar la problemática abordada en el análisis de severidades, sin la utilización de modelos de *machine learning*.

## 1.3. Objetivos

A continuación, se presenta el objetivo general del trabajo de memoria y sus objetivos específicos.

### 1.3.1. Objetivo General

Desarrollar un algoritmo que detecte, clasifique y analice la severidad de grietas presentes en imágenes de asfalto capturadas por el equipo LCMS en carreteras chilenas.

### 1.3.2. Objetivos Específicos

A continuación se enlistarán los objetivos específicos correspondientes a la memoria.

1. Diseñar e implementar un algoritmo para la detección de grietas basada en procesamiento de imágenes.
2. Diseñar e implementar una propuesta para la detección de grietas basada en la implementación de la red pix2pix.
3. Construir conjunto de imágenes de entrenamiento y evaluación para la red pix2pix.
4. Diseñar e implementar un algoritmo para la clasificación de grietas basada en análisis de componentes conectadas.
5. Diseñar e implementar una propuesta para la clasificación de grietas utilizando una red convolucional de clasificación.
6. Construir conjunto de imágenes de entrenamiento y evaluación para la red convolucional de clasificación.
7. Diseñar e implementar un algoritmo capaz de determinar la severidad de las grietas detectadas a partir de su grosor.

## 1.4. Estructura de la Memoria

La memoria se estructura bajo la presente configuración de capítulos:

1. **Introducción:** Se indican el problema y motivación de la memoria, hipótesis de la memoria, y finalmente, objetivos generales y específicos.
2. **Marco Teórico y Estado del Arte:** En el Marco Teórico se explican y detallan los conceptos, técnicas y antecedentes necesarios para entender y abordar el problema. Posteriormente, en el Estado del Arte, se presentan algunos de las soluciones e investigaciones ya existentes para el problema presente en la memoria.
3. **Metodología:** Se presenta la metodología utilizada para abordar el problema, explicando en detalle las partes y etapas presentes en ella, así como bases de datos, arquitecturas y procesos.
4. **Resultados y Discusión:** Primero, se presentan los resultados obtenidos por cada propuesta y etapa, realizando una comparación cuantitativa de los resultados mediante métricas de evaluación, además, se incluye una comparación visual. Posteriormente, se realiza una discusión en relación a las propuestas para cada etapa, explicando ventajas y desventajas en cada una de estas.
5. **Conclusión y Trabajo Futuro:** Finalmente, se entregan las conclusiones obtenidas en la memoria. Además, se presentan propuestas para mejorar el rendimiento y escalar el trabajo realizado.

# Capítulo 2

## Marco Teórico y Estado del Arte

### 2.1. Procesamiento de Imágenes

#### 2.1.1. Procesamiento de Imágenes de bajo nivel

El procesamiento de imágenes de bajo nivel es la aplicación de distintas técnicas sobre imágenes digitales con el objetivo de mejorar la calidad o naturaleza de estas para ser utilizadas posteriormente en otras aplicaciones. A continuación se introducirán las técnicas de bajo nivel aplicadas dentro del marco de la memoria.

#### **Total Variation Denoising**

*Total Variation Regularization* [5] o *Total Variation Denoising* se basa en la idea de que una señal ruidosa posee una variación total alta, es decir, la integral del gradiente absoluto es alto. Así, reducir la variación total de la señal sujeta a que sea parecida a la original, eliminaría detalles molestos, pero, mantiene detalles importantes como los bordes. Para señales en dos dimensiones, Rudin, Osher y Fatemi proponen un método de optimización capaz de reducir este ruido. Sea  $u_0(x, y)$  la función de intensidad de una imagen ruidosa con  $x, y \in \Omega$ , se puede expresar  $u(x, y)$  como la imagen *limpia* de la siguiente forma:

$$u_0(x, y) = u(x, y) + n(x, y) \tag{2.1}$$

donde  $n$  es el ruido aditivo. La idea es reestructurar  $u$  a partir de  $u_0$ . El problema de optimización queda expresado en la ecuación 2.2a y las restricciones de media

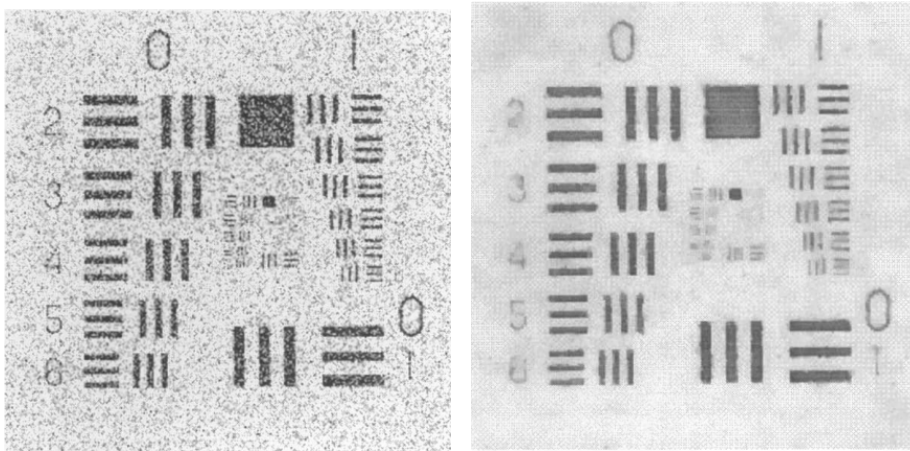
2.2b y desviación estándar 2.2c.

$$\min \int_{\Omega} \sqrt{u_x^2 + u_y^2} \, dx dy \quad (2.2a)$$

$$\text{s.a.} \quad \int_{\Omega} u_0 \, dx dy = \int_{\Omega} u \, dx dy, \quad (2.2b)$$

$$\int_{\Omega} \frac{1}{2} (u - u_0)^2 \, dx dy = \sigma^2, \text{ donde } \sigma > 0 \text{ es dado.} \quad (2.2c)$$

En la práctica, esta técnica es capaz de obtener imágenes menos ruidosas como la presentada en la Fig. 2.1b a partir de una imagen ruidosa como en la Fig. 2.1a.



(a) Imagen Ruidosa.

(b) Imagen obtenida al aplicar *Total Variation Denoising* sobre la imagen 2.1a.

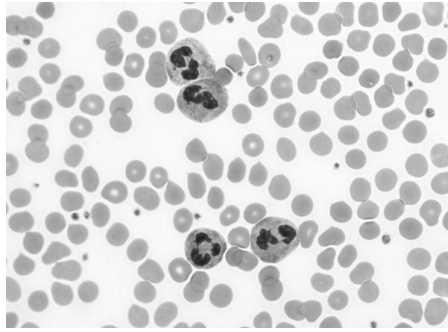
Figura 2.1: Ejemplo de aplicación de *Total Variation Denoising*.

### • Total Variation Chambolle

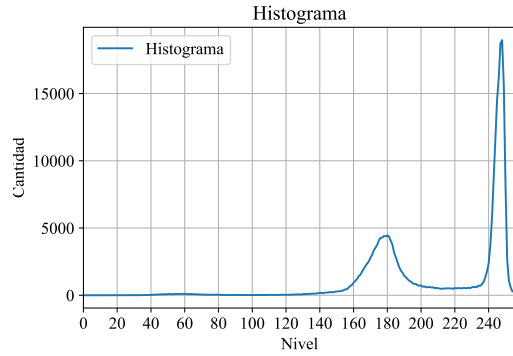
*Total Variation Chambolle* es una modificación de *Total Variation Denoising* realizada por Chambolle [6], en el cual propone un método para resolver el problema de optimización de *Total Variation Denoising* propuesto en 2.2a pero bastante más eficiente en términos de tiempo.

### Histograma de Imágenes

El histograma de una imagen representa el conteo de los niveles en la escala de color o grises que posee cada pixel de una imagen. Este histograma tiene como máximo 256 niveles. En la Fig. 2.2b se puede apreciar el histograma de la imagen en la Fig. 2.2a.



(a) Imagen de ejemplo en escala de grises.



(b) Histograma.

Figura 2.2: Ejemplo de histograma de una imagen.

Sea  $I$  una imagen de  $N \times M$ ,  $h$  el histograma, y  $(x, y) \in N \times M$  un pixel de la imagen. Para cada  $(x, y)$ :

$$h(I(x, y)) = 1 \quad (2.3)$$

### Ecualización de Histogramas

El proceso de ecualización de histograma intenta que cada nivel que pueda tomar un pixel aporte la misma cantidad de información en la imagen. Para realizar la ecualización, es necesario calcular el histograma acumulado de la imagen que se desea ecualizar. El histograma acumulado sin ecualizar de la Fig 2.2a se puede observar en la Figura 2.3.

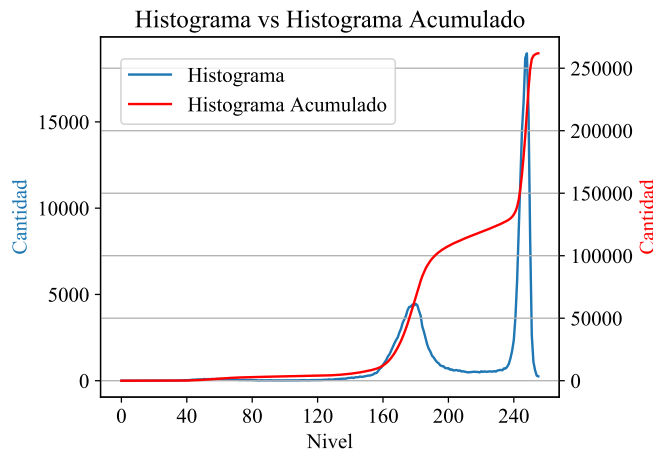
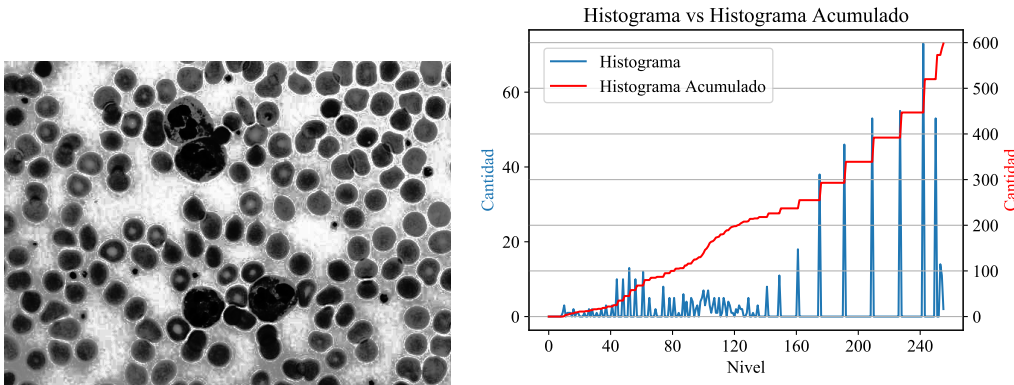


Figura 2.3: Histograma e histograma acumulado pre-ecualización para la imagen Fig 2.2a.

Sea  $M \times N$  las dimensiones de la imagen y  $L = 255$  los niveles que puede tomar un pixel, se define  $g = \frac{M \cdot N}{L}$ . Finalmente, sea  $J$  la imagen ecualizada que se desea obtener, se define la ecualización de histograma como:

$$J(x, y) = \frac{H(I(x, y))}{g} \quad (2.4)$$

Con esta expresión, cada pixel es ponderado para aportar la misma información que otro nivel. Finalmente, el resultado de ecualizar el histograma se puede observar en la Figura 2.4



(a) Imagen después de aplicar ecualización de histogramas.

(b) Histograma e histograma acumulado de la imagen 2.4a

Figura 2.4: Imagen y sus respectivos histograma e histograma acumulado post-ecualización para la imagen Fig 2.2a

### • Ecualización de Histogramas Adaptativo

La ecualización de Histogramas Adaptativo [7] o AHE por su nombre en inglés (*Adaptive Histogram Equalization*) es una técnica que permite mejorar el contraste en imágenes. A diferencia de la ecualización de histogramas normal, se calcula el histograma para distintas secciones de la imagen y se utilizan en conjunto para redistribuir con mayor precisión las intensidades de la imagen. Así, para cada pixel de una imagen, se calcula su nuevo valor en relación al histograma correspondiente al vecindario que lo rodea, de la misma forma que en una ecualización de histogramas común.

### • Ecualización de Histogramas Adaptativo Limitado de Contraste

CLAHE [7] por su nombre en inglés (*Contrast Limited Adaptive Histogram Equalization*) es una modificación al método de AHE. Esta técnica nace a partir de la sobre-amplificación que posee el método de *Adaptive Histogram Equalization* en zonas donde la intensidad es casi constante, pues, el histograma es concentrado

en estas partes de la imagen. Así, CLAHE limita el cambio en los nuevos valores de intensidad de la imagen, evitando que se formen zonas ruidosas donde no las hay. Para esto, se define un límite en el cual se cortará el histograma, pero, la parte sobrante, es redistribuida en el resto del histograma de manera uniforme, tal como se puede apreciar en la Fig. 2.5.

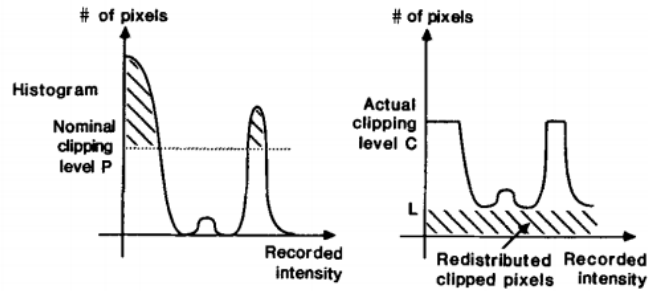
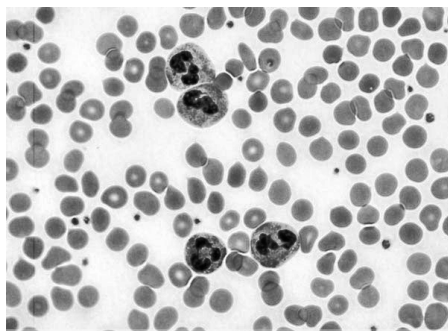
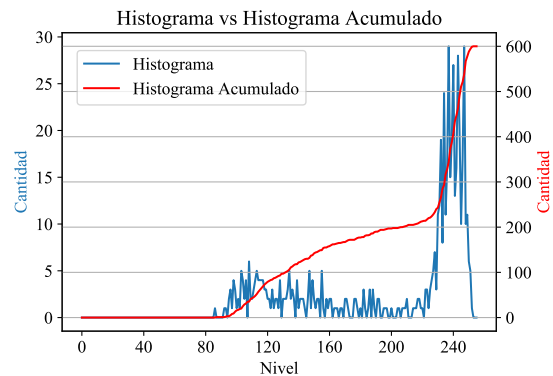


Figura 2.5: Redistribución de valores en el histogramas posterior al *clipping*.

Finalmente, el resultado de ecualizar el histograma de manera adaptativa y limitada se puede observar en la Figura 2.6



(a) Imagen después de aplicar ecualización de histogramas adaptativo con limitación de contraste.



(b) Histograma e histograma acumulado de la imagen 2.6a

Figura 2.6: Imagen y sus respectivos histograma e histograma acumulado post-ecualización de histogramas adaptativo con limitación de contraste para la imagen Fig 2.2a.

## Operaciones Morfológicas

A continuación se describirán las operaciones de morfológicas [8] básicas utilizadas.



• **Erosión**

La erosión es una de las operaciones morfológicas básicas. Está definida por el operador  $\ominus$ . Sea  $E$  un espacio Euclideo,  $A$  una imagen Binaria en  $E$ . La erosión de una imagen binaria  $A$  por una estructura  $B$  se define en la Ecuación 2.5.

$$A \ominus B = \{z \in E | B_z \subseteq A\} \quad (2.5)$$

donde  $B_z$  es la traslación de  $B$  por el vector  $z$ , es decir:

$$B_z = \{b + z | b \in B\}, \forall z \in E \quad (2.6)$$

Básicamente, se traslada el centro o punto de referencia de la estructura  $B$  por cada pixel  $(i, j)$  de  $A$ . Si los puntos de la estructura  $B$  se encuentran completamente contenidos en la vecindad del pixel  $(i, j)$ , el pixel es retenido en la imagen, por el contrario, es eliminado. A continuación se puede apreciar un ejemplo de erosión. Sea

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.7)$$

con el centro de  $B$  como punto de referencia, entonces

$$A \ominus B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.8)$$

• **Dilatación**

La dilatación es la segunda operación morfológica básica. Está definida por el operador  $\oplus$ . Sea  $E$  un espacio Euclideo,  $A$  una imagen Binaria en  $E$ . La erosión de una imagen binaria  $A$  por una estructura  $B$  se define en la Ecuación 2.9.

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \emptyset\} \quad (2.9)$$

donde  $B^s$  es la matriz simétrica de  $B$ , es decir:

$$B^s = \{x \in E \mid -x \in B\} \quad (2.10)$$

Similar a la erosión, se traslada el centro o punto de referencia de la estructura  $B$  por cada pixel  $(i, j)$  de  $A$ . Sí el pixel  $(i, j)$  corresponde a un 1, el resto de los pixeles de la imagen  $A$  que están contenidos en la estructura  $B$  son cambiados a 1. Sea

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.11)$$

con el centro de  $B$  como punto de referencia, entonces

$$A \oplus B = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad (2.12)$$

### • Closing

*Closing* es una operación morfológica que nace de la combinación de las operaciones *Erosión*  $\ominus$  y *Dilatación*  $\oplus$ . Está definida por el operador  $\bullet$  y es la erosión de la dilatación entre la imagen  $A$  y la estructura  $B$  (2.13).

$$A \bullet B = (A \oplus B) \ominus B \quad (2.13)$$

### Image Binarization o Image Thresholding

La *binarización* de una imagen tiene como fin convertir una imagen originalmente en escala de grises, a blanco y negro. Es decir, reduce la escala de 256 tonos distintos, a tan solo 2 (binaria). El objetivo es encontrar el nivel exacto -un umbral- en la escala de niveles que pueda diferenciar dos clases en una imagen. Así, sea  $I$  una imagen cualquiera,  $\hat{I}$  la imagen binarizada y  $T$  el umbral de división escogido:

$$\hat{I}(x, y) = \begin{cases} 0, & \text{Sí } I(x, y) \leq T \\ 255, & \text{Sí } I(x, y) > T \end{cases} \quad (2.14)$$

En la Fig. 2.7 se puede visualizar como un *threshold* corta el histograma en dos partes, esto, para la imagen de la Fig. 2.2b. Además, en la Fig. 2.8 se pueden apreciar los resultados obtenidos en las binarizaciones aplicadas.

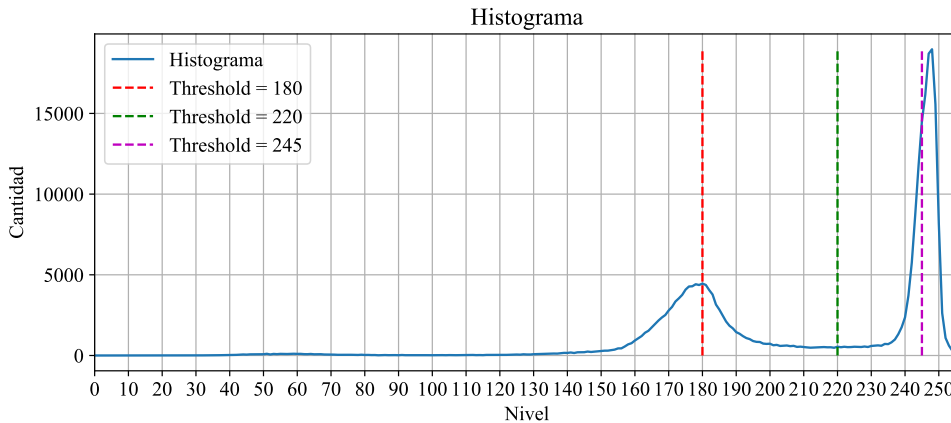


Figura 2.7: Visualización de binarización mediante threshold en el histograma.

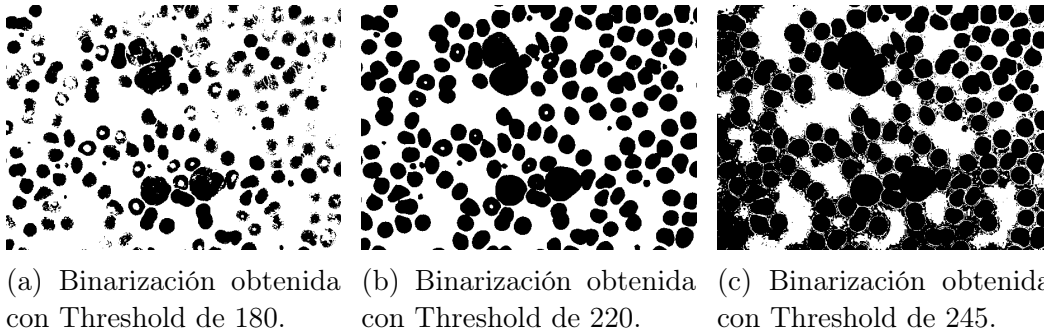


Figura 2.8: Resultados al aplicar Thresholding o Binarización de imagen.

## Detección de Bordos

La Detección de Bordos es el resultado de la convolución de una imagen con un filtro pasa altos. Es decir, un filtro que deja pasar las frecuencias altas (*i.e.* diferencias bruscas en la intensidad de los píxeles) de una imagen, eliminando las bajas.

- Sobel

El filtro Sobel [9] o Sobel-Feldman, nombrado así por sus creadores Irwin Sobel y Gary Feldman, es un operador de dimensiones 3x3 que actúa calculando un

aproximado del gradiente. Así, para cada pixel de una imagen, se calcula el gradiente vertical y horizontal de este, que posteriormente son relacionados. Los filtros verticales y horizontales son convolucionados con la imagen correspondientes, los gradientes obtenidos en cada iteración se pueden apreciar en las Ecuaciones 2.15 y 2.16.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (2.15)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I \quad (2.16)$$

Donde  $G_x$  es la gradiente horizontal obtenida entre la convolución del filtro de Sobel horizontal y la Imagen  $I$  y  $G_y$  es la gradiente vertical obtenida entre la convolución del filtro de Sobel vertical y la Imagen  $I$ . Finalmente, el gradiente se obtiene mediante la relación de la Ecuación 2.17.

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.17)$$

Además, se puede obtener la dirección del gradiente con la Ecuación 2.18.

$$\Theta = \text{atan} \left( \frac{G_y}{G_x} \right) \quad (2.18)$$

### • Histéresis

La histéresis [10] es un método de selección de bordes y es utilizado posterior a la aplicación de filtros pasa altos, como Sobel. Esta técnica define dos umbrales que se utilizarán para determinar que bordes son relevantes de una imagen. Sea  $T_H$  el umbral *alto*,  $T_L$  el umbral *bajo* e  $I_G$  una imagen que ya ha sido convolucionada por un filtro pasa altos. Un pixel pertenece a un borde relevante si:

1. La intensidad del pixel  $I_G(x, y)$  es mayor a  $T_H$ .
2. La intensidad del pixel  $I_G(x, y)$  es mayor a  $T_L$ , pero, el pixel pertenece a un borde que en algún punto es superior a  $T_H$ .

Por el contrario, todo pixel que, en intensidad, sea menor a  $T_L$ , no corresponde a un borde relevante.

## Regiones o Componentes Conectadas

En imágenes binarias, un pixel puede ser negro o blanco, estos valores representan el *fondo* o *background* y el *primer plano* o *foreground*, respectivamente. Las regiones [11] o componentes conectadas (*CCO*), son aquellos pixeles que están conectados (*i.e.* que se tocan) y que pertenecen a un *primer plano*.

### 2.1.2. Redes Convolucionales

Las redes convolucionales o CNN por su nombre en inglés *Convolutional Neural Network*, es una red neuronal utilizada en visión computacional. Es altamente demandada debido a su capacidad de extraer características semánticas a partir de sus capas convolucionales. Esta red está inspirada en como las personas organizan la información que perciben visualmente, y su arquitectura fue implementada por primera vez en 1980 [12], aunque esta aun no consideraba la utilización de *backpropagation*. Este ultimo, fue aplicado por primera vez por LeCun en 1989 [13].

Las redes convolucionales son aplicadas para diversas tareas, dentro de las más utilizadas se encuentran problemas de clasificación, detección, segmentación, generación, entre otros.

En su base, las redes convolucionales están formadas por una capa de entrada, una de salida y múltiples capas ocultas que pueden distribuirse en capas convolucionales y *pooling*.

#### Capa Convolutiva

Son la capa principal de una red convolutiva y la que realiza el mayor trabajo dentro de la red. Una capa convolutiva cumple la función de extraer las características semánticas de las imágenes entregadas a la red. Las capas convolucionales suelen tener tres dimensiones (ancho, largo y alto). Cada celda perteneciente a la capa posee un *peso* que se ajustará durante la etapa de entrenamiento [13].

Al operar con convolución, se extraen las características semánticas de la capa anterior manteniendo una relación espacial entre los *pesos* que lo rodean. Esto es sumamente importante, dado que en una imagen la relación espacial entre un pixel y otro es lo que define lo que hay en esta.

#### Capa de Pooling

Una capa de *pooling* cumple la función de disminuir la dimensionalidad de las capas convolucionales dentro de una red. Esto, para evitar tener demasiados pesos

que ajustar (lo cual implica tener un conjunto de imágenes más grande y mayor tiempo de entrenamiento). Para cumplir este objetivo, una capa de *pooling* opera sobre ventanas de pesos que se encuentran en la capa convolucional antecesora. La operación de una capa de *pooling* puede ser: escoger el máximo (*max pooling*), escoger el mínimo (*min pooling*) o el promedio de los pesos seleccionados (*average pooling*).

## Redes Convolucionales de Clasificación

Las redes convolucionales de clasificación se utilizan para determinar la clase de una imagen. La estructura de una red convolucional de clasificación consiste en una combinación entre capas convolucionales y capas de *pooling*, donde la última puede o no estar presente [13].

Para realizar la clasificación, la última capa corresponde a una capa *fully connected*, que conecta los pesos de la capa convolucional anterior a cada una de las posibles clases. Cada una de las neuronas presentes en la capa *fully connected* posee una función de activación no lineal, siendo las más utilizadas: tangente hiperbólica, sigmoidea y ReLu. Estas funciones determinarán la probabilidad de clase de la imagen procesada, siendo la probabilidad más alta, la escogida.

En la Fig. 2.9 se presenta un ejemplo de arquitectura de una red convolucional de clasificación llamada AlexNet [14]. Además, en la Fig. 2.10 se presenta un ejemplo de su funcionamiento.

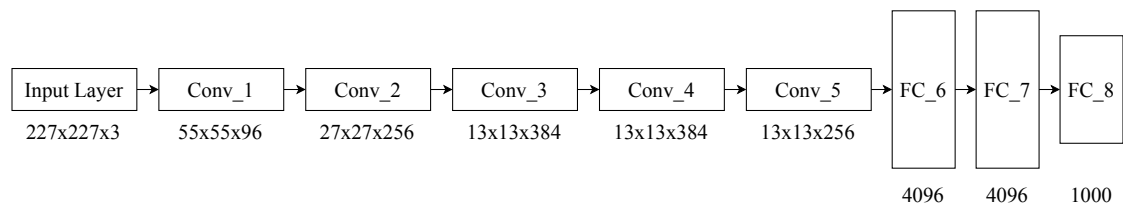


Figura 2.9: Arquitectura de red AlexNet [14].

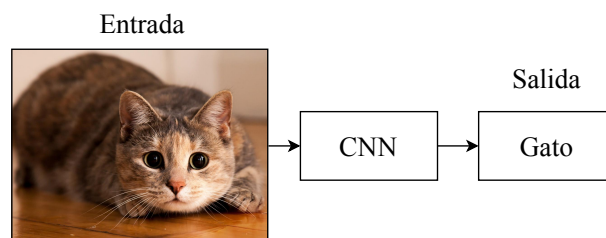


Figura 2.10: Funcionamiento CNN.

## Redes de Segmentación Semántica

La segmentación semántica es un área del procesamiento de imágenes que busca dividir una imagen en áreas o grupos de clases para identificar las distintas sec-

ciones que la componen. Si bien la tarea de clasificar múltiples categorías dentro de una misma imagen se intentó abordar con redes convolucionales como R-CNN [15], no fue hasta 2015, cuando se propuso por primera vez una solución *end-to-end* capaz de resolver el problema de segmentación utilizando solo una red convolucional profunda. Fully Convolutional Networks for Semantic Segmentation [16] fue publicada en marzo de 2015 por Long, Shelhamer y Darrell.

Las redes de segmentación semántica están compuestas principalmente por dos partes. La primera, es el codificador o *encoder*, una arquitectura de red capaz de reducir la información o características semánticas de la imagen analizada una capa reducida que puede ser incluso un vector. La segunda, corresponda al decodificador o *decoder*, esta arquitectura es inversa a la del *encoder* y en ocasiones son simétricas. Esta, busca convertir la información codificada en una imagen de las mismas dimensiones a la original, pero, clasificada o segmentada. En la Fig. 2.11 se puede observar un ejemplo de segmentación semántica realizada por la red [16].



(a) Imagen ejemplo para segmentación semántica.

(b) Segmentación Semántica de 2.11a realizada por [16].

Figura 2.11: Ejemplo de Segmentación Semántica [16].

- U-Net

U-Net [17] es una red convolucional diseñada originalmente para resolver problemas de segmentación semántica en el área de la biomedicina. Ronneberger, Fischer y Brox basaron la arquitectura en la recién publicada Fully Convolutional Networks for Semantic Segmentation [16]. A diferencia de esta, U-Net posee una arquitectura más simétrica y la implementación de *skip connections*, la cual permite realizar una segmentación más fina y precisa.

La arquitectura de U-Net se puede apreciar en la Fig. 2.12.

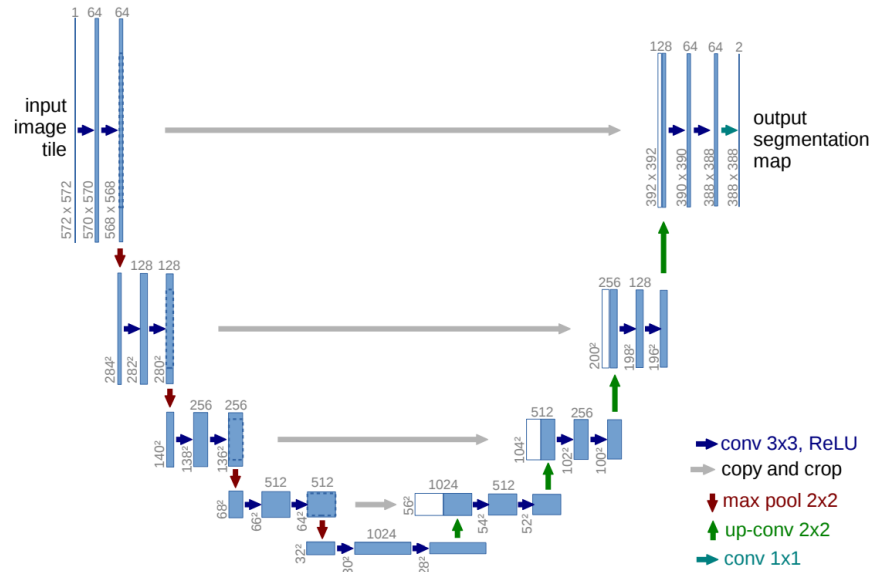


Figura 2.12: Arquitectura U-Net [17].

## Redes Generativas Adversarias

Una Red Generativa Adversaria o *Generative Adversarial Network* (GAN) es una red compuesta por dos sub-redes: un generador y un discriminador. El generador es una red capaz de generar resultados sintéticos a partir de un *input* aleatorio o condicional, mientras que, el discriminador es capaz de determinar si un *input* entregado es real o sintético. Estas redes son adversarias, pues, compiten entre ellas para mejorar la capacidad de generar o discriminar[18].

El generador tiene como objetivo engañar al discriminador (es decir, maximizar su error), para esto, la red es entrenada para mejorar sus resultados sintéticos, siendo estos cada vez más realistas, mientras que, el discriminador es entrenado con el objetivo de mejorar sus criterios de clasificación, evitando ser engañado por los resultados sintéticos del generador[18].

### • Pix2pix

*Pix2pix* es una red *image-to-image* basada en una generativa adversaria condicional. Esta particularidad implica que la imagen resultante por el generador es condicional al *input* que este posee[19].

El generador propuesto para *pix2pix* está basado en la arquitectura *U-Net*[19][17], una red *encoder-decoder* que posee *skip connections* que permiten conservar detalles mientras se codifica y decodifica la información en la red. *U-Net* es una red *image-to-image*, es decir, fabrica una imagen de salida a partir de una de entrada[17].



Por otra parte, el discriminador propuesto es una red convolucional que clasifica si la imagen de entrada es real o generada (sintética). Para mejorar el rendimiento de esta red, se implementó una función de *loss* capaz de determinar que partes de las imagen parecen 'falsas', es decir, no solo determina si la imagen es real o sintética, sino también, especifica cuales son las áreas que concluyen este resultado[19].

En la práctica, la red *pix2pix* ha sido utilizada para problemas de: colorización de imágenes, segmentación, detección de bordes, creación de mapas, entre otros problemas.

Finalmente, la arquitectura de la red *pix2pix* presentada en [19] se puede apreciar en la Fig. 2.13 y en las Tablas 2.1 y 2.2.  $Ck$  hace referencia a una capa convolucional con *BatchNorm* y activación *ReLU* con  $k$  filtros. Por otra parte,  $CDk$  hace referencia a una capa convolucional con *BatchNorm*, *Dropout* de 50 % y activación *ReLU*. Todas las capas convolucionales poseen filtros espaciales de tamaño  $4 \times 4$  y *stride* 2.

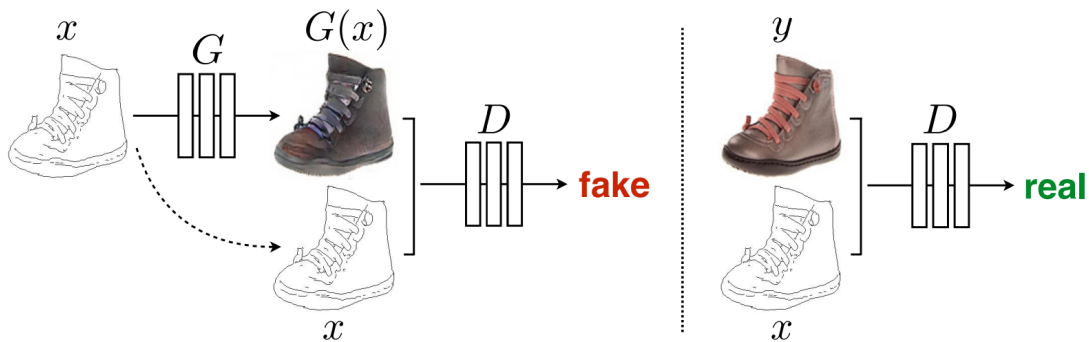


Figura 2.13: Arquitectura *pix2pix* [19].

Tabla 2.1: Arquitectura del generador *pix2pix*

Generador							
<i>Encoder</i>	C64	C128	C256	C512	C512	C512	C512
<i>Decoder</i>	CD512	CD512	CD512	C512	C256	C128	C64

Tabla 2.2: Arquitectura del discriminador *pix2pix*

Discriminador					
C64	C128	C256	C512	C512	C512

### 2.1.3. Métricas de Evaluación

A continuación se presentaran las métricas de evaluación utilizadas en el marco de la memoria.

## Dice Coefficient o $F_1$ Score

Dice Coefficient [20] es una métrica de evaluación utilizada para determinar la similitud entre dos muestras para cualquier estudio. En el ámbito de procesamiento de imágenes también es conocido como  $F_1$  Score, pues, es la adaptación del Coeficiente de Dice para el análisis estadístico de una clasificación binaria [21].

Esta métrica considera la *precisión* y el *recall*, donde:

$$\text{precision} = \frac{VP}{VP + FP} \quad (2.19)$$

$$\text{recall} = \frac{VP}{VP + FN} \quad (2.20)$$

siendo; VP: Verdaderos Positivos, FP: Falsos Positivos, VN: Verdaderos Negativos y FN: Falsos Negativos.

$F_1$  Score es la media armónica entre la precisión y el *recall* [21], así:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.21)$$

A continuación, se presenta un ejemplo de como utilizar esta métrica sobre la Fig. 2.14.

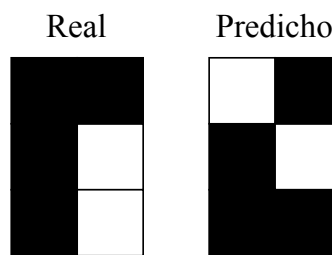


Figura 2.14: Imágenes de ejemplo para métrica *dice coefficient*.

Al realizar un análisis sobre la imagen predicha sobre la imagen real de la Fig. 2.14, se puede observar que hay dos píxeles que fueron erróneamente clasificados. Obteniendo un falso positivo en la esquina superior izquierda y un falso negativo en la esquina inferior derecha. En resumen, la imagen predicha presenta:  $VP = 1$ ,  $FP = 1$ ,  $VN = 3$  y  $FN = 1$ . Al resolver las ecuaciones 2.19 y 2.20, se obtiene  $\text{precision} = \frac{1}{1+1} = 0,5$  y  $\text{recall} = \frac{1}{1+1} = 0,5$ . Finalmente, la media armónica para esta clasificación puede determinarse como  $F_1 = 2 \cdot \frac{0,5 \cdot 0,5}{0,5 + 0,5} = 0,5$ .

## Matriz de Confusión

La matriz de confusión o matriz de error [22] es una métrica que permite obtener una visualización del desempeño de un algoritmo respecto a la tarea de clasificación. Las columnas representan las predicciones de clase realizadas por el algoritmo, mientras que, las filas representan la clase real. En la Tabla 2.3 se presenta un ejemplo de una matriz de error.

Tabla 2.3: Ejemplo de matriz de confusión.

		Clase Predicha		Precisión [%]
		A	B	
Clase Real	A	2	2	50 %
	B	1	3	75 %

De la Tabla 2.3, se puede interpretar que el algoritmo tuvo una precisión del 50[%] en clasificar la clase A, pues, predijo correctamente 2 de 4 ejemplos. Mientras que, para la clase B, obtuvo una precisión del 75[%], pues, predijo correctamente 3 de 4 ejemplos.

### 2.1.4. Detección de Grietas mediante procesamiento de imágenes

A continuación se presentan distintas investigaciones relacionadas a la detección de grietas sobre pavimento utilizando diversas técnicas de procesamiento de imágenes.

## Regiones Enlazadas

Eliminar el ruido del pavimento ha sido una de las variables más complicadas de tratar luego de aplicar un *thresholding*. Oliveira y Lobato, en [23], tratan este problema relacionando las regiones obtenidas posterior a la binarización y determinando cuales tienen una conexión o enlace entre sí. Para esto, proponen dos densidades de probabilidad que determinan si una región  $r_i$  se encuentra enlazada a una región  $r_j$ . Esta probabilidad nace de la relación entre la menor distancia que poseen estas regiones, el menor ángulo entre las orientaciones de las elipses que envuelven las regiones y el menor ángulo entre la orientación de la recta que une los centroides de ambas regiones y la elipse que tiene el mismo segundo momento central normalizado que la región  $r_i$ . Así, la relación de estas componentes determinan si las regiones se encuentran o no enlazadas, de ser así, ambas permanecen en la imagen como grietas, en el caso contrario, son excluidas como ruido. En la Fig. 2.15 se puede apreciar como el método de regiones enlazadas logra eliminar ruido del pavimento y proporcionar continuidad a las grietas.

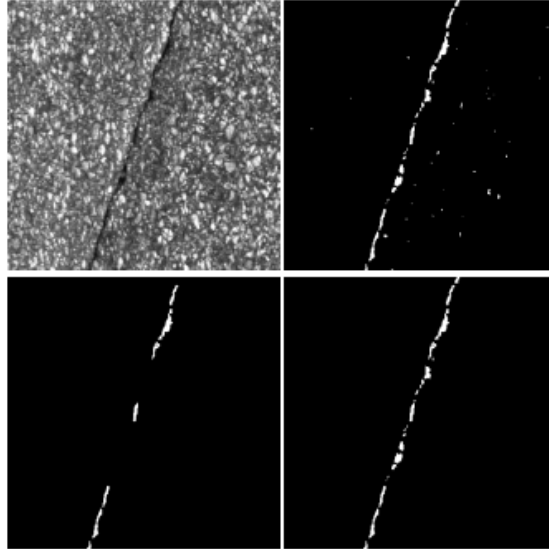


Figura 2.15: *Connected components linkage results. Each pair of rows (from left to right and top to bottom) represents: original image; segmentation result; the connected components identified as relevant; the final crack regions detection after the proposed linkage [23].*

### Umbrales de Intensidad o *Thresholding*

Los métodos más sencillos de detección utilizan técnicas de *thresholding* como base para determinar que es grieta y que es pavimento en una imagen. En [24] se muestra una mejora del método de binarización de *OTSU*. Esta, realiza un cambio en la varianza entre clases (grieta y pavimento) agregando como ponderador la distribución de probabilidad de todos los pixeles del vecindario del umbral. Este cambio proporciona mayor adaptabilidad en la elección del umbral en la detección, eliminando gran parte del ruido proporcionado por el pavimento. En la Fig. 2.16 se puede apreciar los resultados de esta modificación.

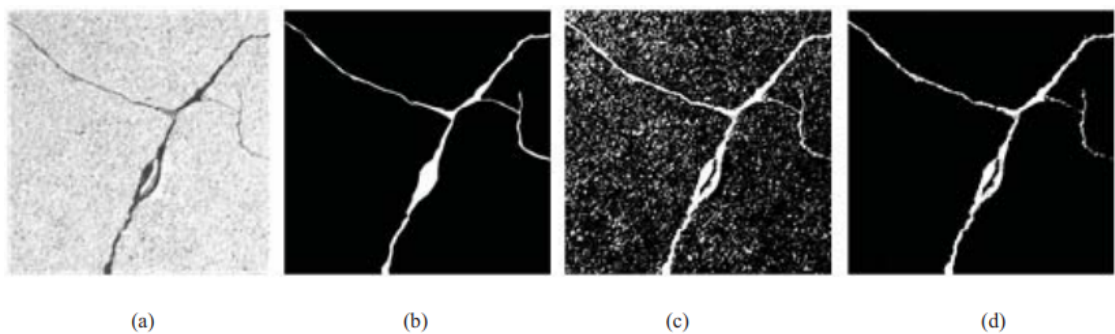


Figura 2.16: Resultados Experimentales: (a) *Captured image (left column, from top to bottom)*; (b) *CrackForest (the second column)*; (c) *Normal Otsu processed image (the third column)*; (d) *Improved Otsu processed image (right column) [24].*

## Z-Score

Otro método utilizado para identificar las grietas presentes en una imagen, es realizar un análisis de los píxeles que se encuentran en una misma fila de la imagen. Así, aquellas intensidades en los píxeles que escapan (son más bajas) de la media y varianza del pavimento se pueden interpretar como grietas. El método de Oliveria y Lobato en [25] utiliza Z-Score para identificar las distribuciones de grietas y pavimentos y posteriormente diferenciarlas iterativamente. En la Fig. 2.17 se pueden apreciar los resultados obtenidos utilizando este método.

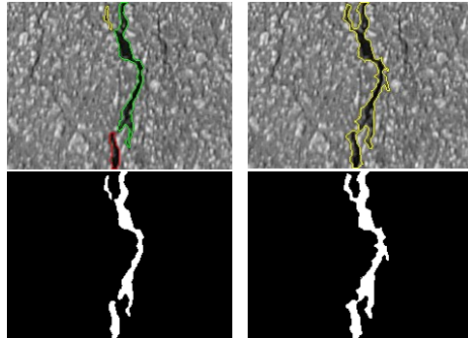


Figura 2.17: *Sample crack segmentation results: initial crack seeds (left) and after crack segmentation refinement (right).* [25].

## Filtro de Partículas

Por otra parte, algunas investigaciones han tratado este problema desde la perspectiva de probabilidades. Una de las propuestas usa como base la utilización de filtro de partículas, la cual en base a probabilidades, estima la posición de una partícula en  $t + 1$ , a partir de las condiciones en  $t$ . Hassan, Manthavan y Kamal utilizan la *Secuencia de Monte Carlo* para determinar el comportamiento de una grieta y como esta *crece* para determinar su posición dentro de una imagen [26]. En la Fig. 2.18 se puede apreciar como la secuencia de montecarlos propone distintos estados para la grieta y elimina los que no cumplan la condición.

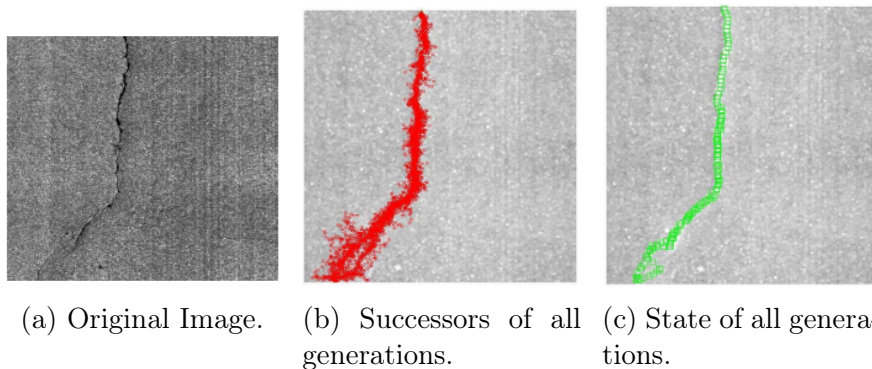


Figura 2.18: Experimental Results [26].

## Vecindarios o *Neighbourhood*

Avila, Begot, Duculty y Nguyen elaboraron un algoritmo de programación dinámica capaz de analizar las distintas trayectorias de una grieta a partir de cuatro vecindarios conectados orientados, lo cuales fueron formados a partir de ocho grafos orientados definidos por los autores. Este método propone buscar las cuatro distintas trayectorias posibles en vecindarios de largo  $d$ , así, las trayectorias que tengan la suma de intensidades más bajas son seleccionada. Posteriormente, utilizando probabilidades se decide si las trayectorias seleccionadas para un pixel en particular pertenecen a una grieta o a pavimento [27]. En la Fig. 2.19 se pueden apreciar los resultados obtenidos utilizando este método.

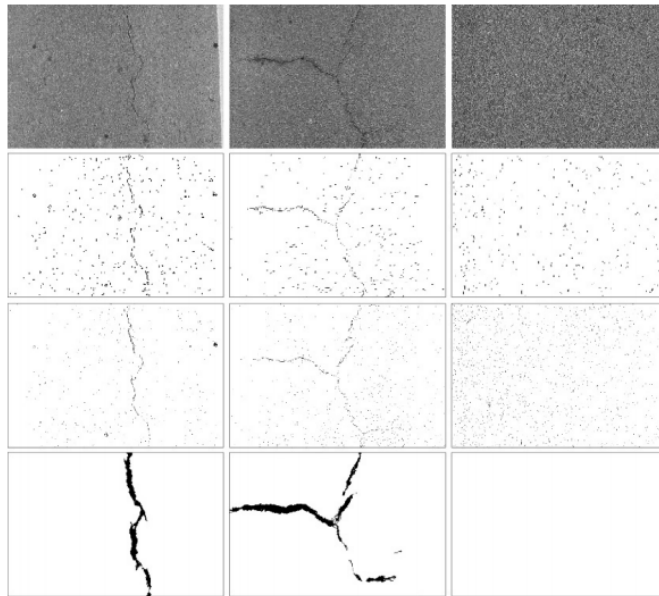


Figura 2.19: *Comparative results on static images. Row1: original images (the last without crack), row2: CWT Markov method, row3: Threshold, row4: FFA [27].*

## 2.2. Aspectos generales de caminos, deterioros y mediciones

### 2.2.1. Componentes de una carretera

A continuación se enlistarán los componentes de una carretera y su respectiva definición que son relevantes para la memoria:

- **Calzada:** Parte de la carretera destinada a la circulación de vehículos. Puede estar compuesta de uno o varios carriles.
- **Acera:** Superficie destinada a la circulación de peatones.
- **Carril:** División de la calzada delimitada por marcas viales.

- **Delimitación Vial:** Líneas de pintura blanca utilizadas para dividir partes en la carretera.
- **Grieta Activa:** Grieta en carretera no sellada.
- **Grieta Sellada:** Grieta en carretera que ya ha sido tratada.

### 2.2.2. Clasificación y Severidad de Grietas

A continuación se presentarán las clases de grietas existentes en pavimentos flexibles y las severidades que puedan tener, señalados por el Manual de Carreteras del Gobierno de Chile [28].

#### Grietas de Fatiga

Son una serie de fisuras y grietas interconectadas entre sí. Forman una *mallá de gallinero* o *piel de cocodrilo*. Un ejemplo se puede apreciar en la Fig. 2.20.

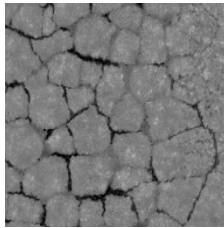


Figura 2.20: Ejemplo de grieta de fatiga [28].

#### Grietas Longitudinales

Son predominantemente paralelas al eje de la calzada. Preferentemente localizadas donde circula la mayor parte del tráfico. Un ejemplo se puede apreciar en la Fig. 2.21.

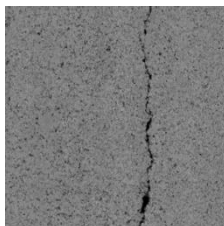


Figura 2.21: Ejemplo de grieta longitudinal [28].

## Grietas Transversales

Son predominantemente perpendiculares al eje de la calzada. Un ejemplo se puede apreciar en la Fig. 2.22.

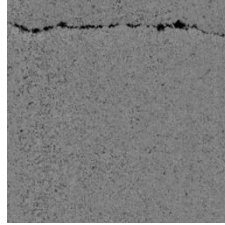


Figura 2.22: Ejemplo de grieta transversal [28].

## Niveles de Severidad

Las grietas presentes en pavimentos pueden clasificarse bajo varios criterios, siendo el más utilizado la severidad que esta posee. La severidad se determina a partir del ancho medio que posee una grieta. Los distintos niveles de severidad se enlistan a continuación:

1. **Baja:** Grietas de ancho medio menor a 3[mm].
2. **Media:** Grietas de ancho medio entre 3[mm] y 20[mm].
3. **Alta:** Grietas de ancho medio mayor a 20[mm].

La detección de grietas en carreteras tiene décadas de investigación, esto, debido a la gran variedad de pavimentos y condiciones en el mundo. Además, las mejoras en tecnología permiten la innovación de nuevas técnicas capaces de automatizar estos procesos con mayor precisión y eficacia. A continuación se presentaran algunas de las investigaciones realizadas para la detección de grietas en carreteras.

### 2.2.3. Laser Crack Measurement System

*LCMS* [4] es la tecnología más moderna utilizada en la actualidad para detectar grietas en carreteras. Está compuesta por dos láseres 3D de alta calidad capaces de medir de manera transversal una pista de carretera y con una precisión de 1[mm] a alta velocidad.

Los sensores del *LCMS* son capaces de adquirir dos perfiles de manera simultánea: *intensity* y *range*. La Fig. 2.23 muestra como se puede obtener información a partir de los sensores mencionados.



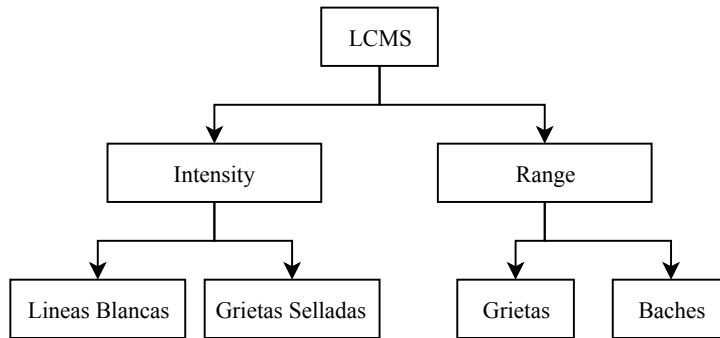
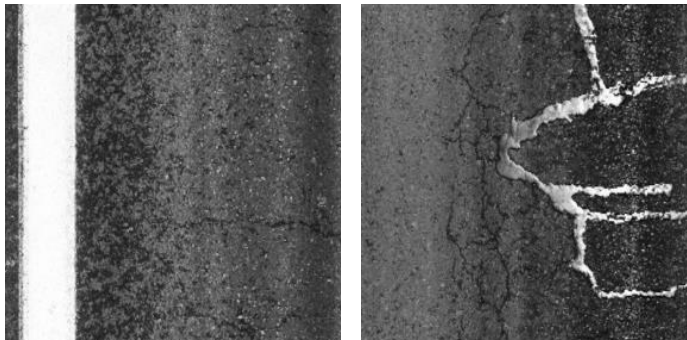


Figura 2.23: Esquema de información obtenida de los láseres del LCMS.

## Intensity

Las imágenes *intensity* [4] es la reflexión obtenida de uno de los láseres del *LCMS*, el resultado es similar al de sacar una foto pero con mayor precisión. El perfil de intensidad está diseñado principalmente para la detección de limitaciones en carreteras, es decir, líneas blancas pintadas en la carretera. Es fácil diferenciar a partir de estas imágenes la diferencia entre las limitaciones y la carretera. Por otra parte, las imágenes de intensidad proveen información relevante respecto a grietas que ya se encuentran selladas, las cuales suelen verse de color blanco (similar a las demarcaciones). Un ejemplo de los usos mencionados se puede apreciar en las Fig. 2.24a y Fig. 2.24b.



(a) Línea Blanca observada a partir de láser *intensity*.

(b) Grieta Sellada observada a partir de láser *intensity*.

Figura 2.24: Posibles detecciones a partir del láser *intensity*.

## Range

Por otra parte, la información obtenida a partir del sensor de *range* [4] es tridimensional. El láser del equipo mide la distancia entre este y la superficie, logrando una representación en escala de grises, en la cual, los colores más oscuros representan una superficie más baja que los colores más claros. Este perfil está diseñado

principalmente para la detección de grietas usando la variación de profundidad en los datos recolectados. De esta misma manera, es posible detectar baches u otros fenómenos que provoquen una diferencia en la superficie de la carretera. Un ejemplo de imagen *range* se puede apreciar en la Fig. 2.25.

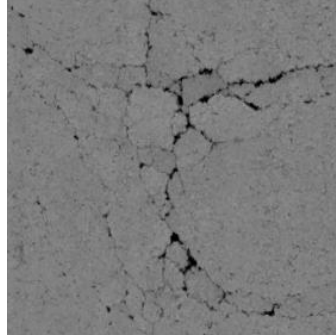


Figura 2.25: Grieta observada a partir de láser *range*

### Software Interno

Los dispositivos *LCMS* poseen un software interno [4] capaz de unir la información de ambos láseres y hacer una representación en 3D que es utilizada para la detección de grietas.

# Capítulo 3

## Desarrollo

### 3.1. Formalización del problema

El problema a abordar en la memoria se basa principalmente en la detección, clasificación y análisis de severidad de grietas presentes en imágenes capturadas por el *LCMS* de carreteras chilenas. Así, se entiende el problema, tal que, para una imagen de entrada  $\mathbf{x}$  (una imagen *range* de carretera), se obtiene una imagen de salida  $\mathbf{y}$ , que incluye la detección de las grietas presentes en la imagen  $\mathbf{x}$ , la clasificación y análisis de severidad de estas.

Como se mencionó en la Introducción, es sumamente relevante clasificar y analizar la severidad de las grietas. Esto es necesario debido a que es determinante entender cómo y en qué momento abordarlas.

Si bien los métodos mencionados en el Estado del Arte presentan soluciones para el problema de detección de grietas, estas, no dan relevancia al tiempo de procesamiento, clasificación o severidad de la grieta. En esta memoria se proponen métodos capaces de detectar, clasificar y analizar la severidad de grietas presentes en carreteras de asfalto.

### 3.2. Propuestas

El problema mencionado se puede dividir en tres partes: detección de grietas, clasificación de grietas y análisis de severidades. En esta memoria, se presentan dos propuestas para la etapa de detección, dos propuestas para la etapa de clasificación y una propuesta para la etapa de análisis de severidades.

La etapa de detección de grietas tiene como objetivo obtener una imagen *binarizada* entre pavimento y grietas. Se evaluará el desempeño de dos metodologías distintas; la primera, consiste en utilizar técnicas de procesamiento de imágenes de

bajo nivel, mientras que, en la segunda metodología se utilizan modelos convolucionales, particularmente, la red *pix2pix*.

La etapa de clasificación tiene como objetivo obtener la distintas clases de grietas presentes en una imagen, qué, como se mencionó en el Marco Teórico, pueden ser de fatiga, transversal o longitudinal. Similar a la etapa anterior, se emplearán dos metodologías distintas, donde; la primera, utiliza técnicas de análisis de componentes conectadas (cco), tales como, determinar la cantidad de bifurcaciones, el ángulo de inclinación, el ancho y el largo de una componente, entre otras características para determinar su clase, mientras que, la segunda metodología, utiliza una red convolucional de clasificación.

Finalmente, la etapa de análisis de severidades tiene como objetivo clasificar las distintas secciones de una grieta dentro de tres severidades; baja, media y alta. El método a evaluar calcula la distancia más cercana de un pixel de grieta a un pixel de pavimento, así, realizando un análisis de estas distancias, se determinará la severidad del pixel analizado.

Cada uno de los métodos utilizados en cada etapa será evaluado en rendimiento, es decir, precisión y velocidad. Para la primera etapa, la precisión será medida utilizando la métrica de *coeficiente Dice*. La segunda etapa será evaluada utilizando matrices de confusión. Finalmente, la tercera etapa será evaluada por un experto en el área de análisis de severidades. También, todas las etapas serán evaluadas en velocidad, es decir, se medirá el tiempo de ejecución de la propuesta.

### 3.3. Bases de datos utilizadas

Para desarrollar la memoria es necesario tener una gran cantidad de imágenes que permitan entrenar las distintas redes utilizadas en cada metodología y evaluar los resultados. Si bien en la literatura existen imágenes de grietas en carreteras (incluso con sus respectivos *target*), estas imágenes no representan el problema a solucionar en la memoria, pues, no son imágenes *range* (sino, más bien de tipo *intensity*, pues, son fotos) del *LCMS*, otra maquinaria similar o simplemente una cámara fotográfica. Debido a esto, es necesario conseguir y crear las bases de datos a trabajar en las propuestas. En las siguientes sub-secciones se describen las bases de datos utilizadas en la memoria.

#### 3.3.1. Imágenes *Range* e *Intensity*

La base de datos *base* utilizada en la memoria corresponde a imágenes de *range* e *intensity* de carreteras chilenas proporcionadas por el dispositivo *LCMS* de la empresa APSA Ltda. La base de datos está compuesta por mediciones realizadas en carreteras a lo largo de todo Chile y contiene más de 12 mil imágenes en total. Las imágenes corresponden a mediciones de 10 metros de un solo carril. Estas,

si bien pueden tener distintas resoluciones dependiendo del tipo de exportación seleccionado, en la base de datos utilizada, se utilizaron imágenes de  $1010 \times 2500$  [px]. En la figura 3.1 se presenta un ejemplo de las imágenes mencionadas.

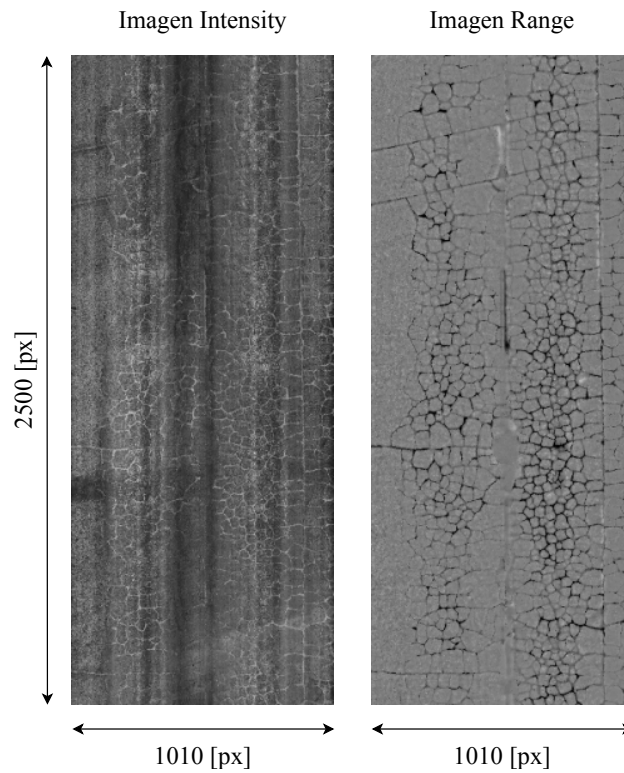


Figura 3.1: Ejemplo de imágenes *range* e *intensity* de la base de datos.

### 3.3.2. Base de Datos para Etapa de Detección

#### CRACK10M

Para evaluar los resultados obtenidos en los métodos utilizados en esta etapa, es necesario tener dos tipos de imágenes:

- Imagen de entrada: Imagen *range* original de carretera.
- Imagen de salida: Imagen *binarizada* entre grietas y pavimento.

Como las imágenes de *salida* **no se encuentran** en la base de datos del dispositivo de APSA, fue necesario crearlas **manualmente**. Para esto, se realizó un *labeling* en las imágenes con el programa de edición *Adobe Photoshop*.

Estas imágenes contienen la información de un carril de 10[m] de largo. Finalmente, se binarizaron 85 imágenes manualmente, del cual, 68 se utilizarán para entrenamiento y 17 para prueba.

En la Fig. 3.2 se puede apreciar un ejemplo de imagen de entrada y salida de la base de datos.

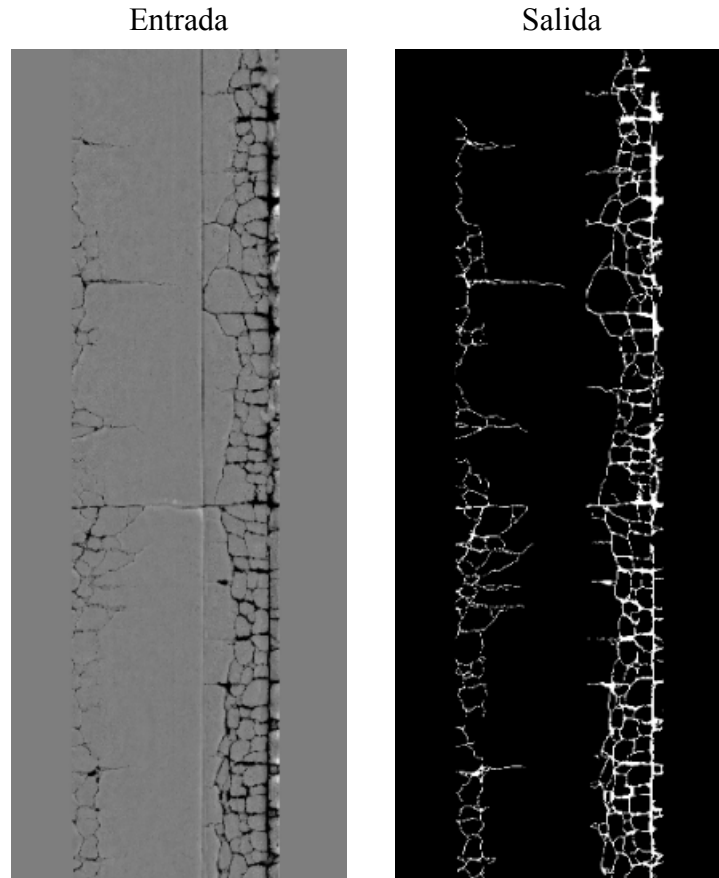


Figura 3.2: Ejemplo de imagen de entrada y salida para base de datos CRACK10M.

### CRACK256

Además, es necesario construir un conjunto de imágenes para entrenar la red *pix2pix*. Así, sobre el conjunto CRACK10M se utilizaron técnicas de ***Data Augmentation***. Específicamente, se dividió cada imagen *range* en sub-imágenes de  $256 \times 256$  [px]. A cada sub-imagen se le aplicó dos rotaciones ( $90^\circ$  y  $-90^\circ$ ) y tres volteadas o *flips* (vertical, horizontal y diagonal). Este proceso se puede apreciar en el ejemplo de la Fig. 3.3.

El resumen de estos conjuntos se puede apreciar en la Tabla 3.1.

Tabla 3.1: Resumen para conjuntos CRACK10M y CRACK256.

	Entrenamiento	Test	Total
CRACK10M	68	17	85
CRACK256	11.017	2.377	13.394

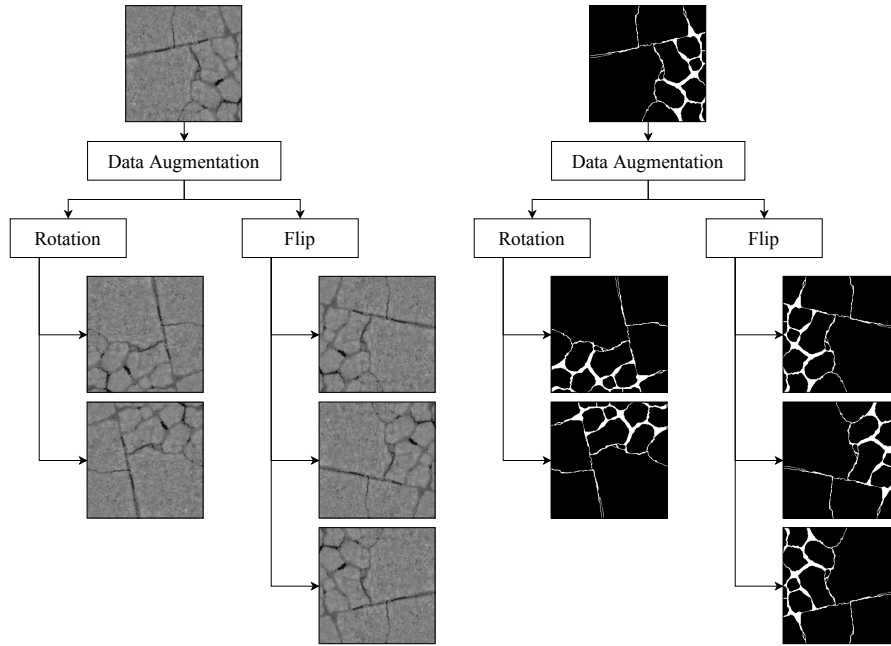


Figura 3.3: Ejemplo de aplicación de Data Augmentation en sub-imágenes de  $256 \times 256$  [px].

### 3.3.3. Base de Datos para Etapa de Clasificación

Para entrenar y evaluar los resultados los métodos utilizados en esta etapa, es necesario tener dos tipos de base de datos, una para el entrenamiento de una **CNN** y otra para la red **DeeplabV3**.

#### CNN

Esta red tiene como entrada la imagen *binarizada* de  $256 \times 256$  [px] y como salida, la clase de la grieta, la cual está representada numéricamente. Para esto se preparó un set de datos de 7674 imágenes con su respectiva clase. El resumen de la base de datos se presenta en la Tabla 3.2.

Tabla 3.2: Resumen de imágenes utilizadas para clasificación en CNN.

Clases	Cantidad	Label
Pavimento	1254	0
Grieta de Fatiga	2364	1
Grieta Transversal	2394	2
Grieta Longitudinal	1662	3

### 3.4. Pre-Procesamiento

Antes de comenzar con la detección, es necesario realizar un pre-procesamiento a las imágenes *range* del dispositivo LCMS. Este, consiste en hacer una filtración de las delimitaciones viales presentes en las imágenes *intensity* sobre las imágenes *range*. Cabe recordar, que la información de delimitaciones viales y grietas selladas se encuentran solo en las imágenes *intensity*, por lo que, es necesario de ambas para poder realizar este filtrado.

La filtración de delimitaciones viales y todo lo externo que no pertenezca al carril en sí, tiene dos justificaciones. La primera, es la cantidad de ruido que presentan las delimitaciones viales, acera, etc., esto, debido a que las imágenes *range* son compuestas en base a la distancia entre el láser y el pavimento, el cual puede verse afectado por el grosor de la pintura en las delimitaciones, la misma pintura quebrajada u otros elementos externos a la textura del carril. Segundo, al cuantificar las grietas presentes en una carretera, es necesario tener cuidado de no contabilizar dos veces una misma grieta (lo cual puede ocurrir al medir dos carriles contiguos), es por esto, que es estrictamente necesario eliminar todo aquello que no pertenezca al carril analizado en cuestión.

Para lograr el filtrado, primero se identifican las regiones blancas en la imagen *intensity* que tengan las condiciones de una delimitación vial, es decir, rectangular y alargada, con un ángulo de inclinación menor a  $60^\circ$ , y que, estuvieran en los extremos del carril. Al analizar las propiedades de estas regiones, se obtiene la mínima elipse que la contiene y sus características, tales como: el eje mayor, eje menor, centro e inclinación. Dada esta información, se realiza una aproximación de las esquinas de la región de una delimitación vial. Sea  $(x_0, y_0)$  el centro de la elipse,  $\theta$  el ángulo de inclinación,  $y_\alpha$  el eje mayor y  $x_\alpha$  el eje menor. Las esquinas de la delimitación vial son similares a las esquinas de la elipse, así:

- Esquina superior izquierda:  
 $e_{s.i.} \sim (x_{s.i.}, y_{s.i.}) = (x_0 - \frac{1}{2} \cdot \cos(\theta) \cdot x_\alpha, y_0 - \frac{1}{2} \cdot \cos(\theta) \cdot y_\alpha).$
- Esquina superior derecha:  
 $e_{s.d.} \sim (x_{s.d.}, y_{s.d.}) = (x_0 + \frac{1}{2} \cdot \cos(\theta) \cdot x_\alpha, y_0 - \frac{1}{2} \cdot \cos(\theta) \cdot y_\alpha).$
- Esquina inferior izquierda:  
 $e_{i.i.} \sim (x_{i.i.}, y_{i.i.}) = (x_0 - \frac{1}{2} \cdot \cos(\theta) \cdot x_\alpha, y_0 + \frac{1}{2} \cdot \cos(\theta) \cdot y_\alpha).$
- Esquina inferior derecha:  
 $e_{i.d.} \sim (x_{i.d.}, y_{i.d.}) = (x_0 + \frac{1}{2} \cdot \cos(\theta) \cdot x_\alpha, y_0 + \frac{1}{2} \cdot \cos(\theta) \cdot y_\alpha).$

Para las delimitaciones viales del lado izquierdo de la carretera, se utilizan las esquinas inferior y superior derecha, mientras que, para las del lado derecho, se utilizan las esquinas izquierdas. Utilizando estas esquinas, se calculó la recta que pasa por las esquinas pares correspondientes y se dividió la imagen en carril y 'otro', eliminando todo lo que no fuera carril. Esta filtración puede entenderse con mayor claridad en la Fig. 3.4.



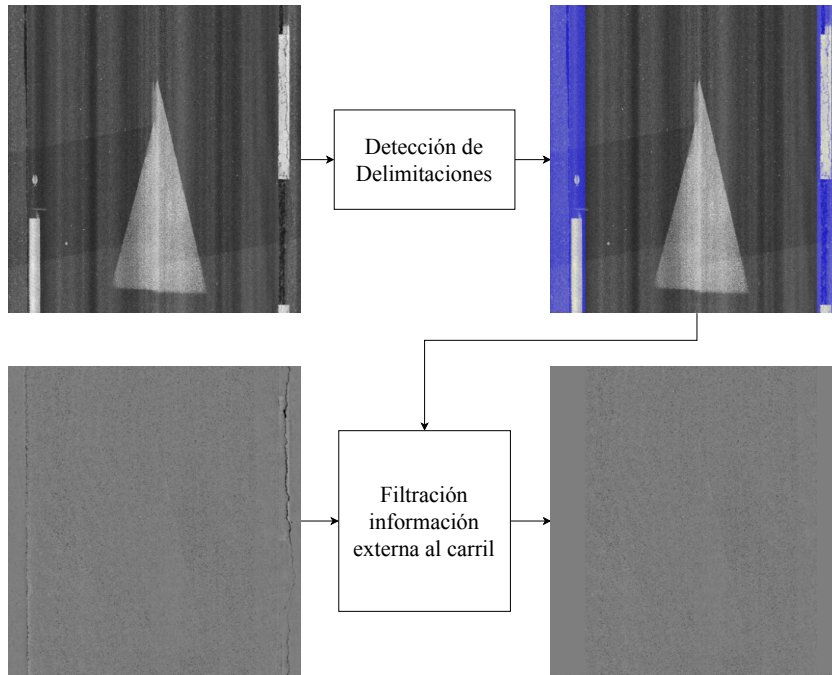


Figura 3.4: Filtración de delimitaciones vial e información irrelevante.

### 3.5. Etapa de Detección

Como ya se ha mencionado, esta etapa tiene como objetivo la *binarización* entre grieta y pavimento para imágenes *range* del LCMS. Esto se puede entender con el esquema presentado en la Fig. 3.5.

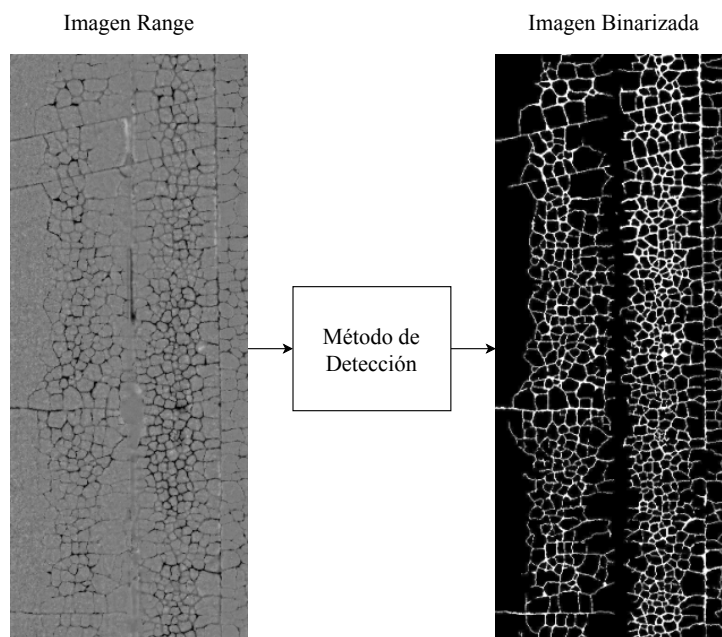


Figura 3.5: Esquema de Etapa de Detección.

A continuación, en las siguientes sub-secciones, se presentarán dos propuestas distintas que abordan esta etapa.

### 3.5.1. Procesamiento de Imágenes

El primer método aborda el problema en dos fases; la primera, intenta *mejorar* la calidad de las imágenes para destacar las grietas presentes en ella y eliminar el ruido del pavimento, la segunda fase, trata la detección en sí, es decir, aplica las técnicas finales para obtener la imagen *binarizada* final.

#### Primera Fase

El procesamiento realizado en esta fase se puede apreciar con detalle en la Fig. 3.6.

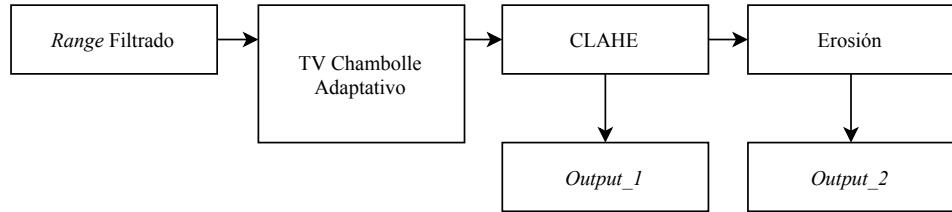


Figura 3.6: Esquema de técnicas utilizadas en la etapa de Procesamiento de Imágenes.

Esta fase comienza con la aplicación de *TV Chambolle*, una técnica de *denoising* para eliminar el ruido presente en las imágenes. Dado que, en un carril pueden coexistir hasta dos tipos de composición en la composición de pavimentos, es necesario aplicar esta técnica de forma *localizada*. Para ello, se agregaron dos procesos al aplicar esta técnica; primero, se segmentó la imagen en sub-imágenes de  $256 \times 256$  [px], segundo, el peso del *denoising* aplicado a cada sub-imagen está sujeto a la cantidad normalizada de píxeles con nivel  $noise_{level}$  (que es el nivel de intensidad en el cual convive mayoritariamente el ruido) ponderado por  $\alpha$ . Así, el peso utilizado en TV Chambolle está dado por la Ecuación 3.1.

$$weight = \alpha \cdot noise_{level} \quad (3.1)$$

Posteriormente, se utiliza la técnica de *CLAHE*, el cual destaca los píxeles de grietas por sobre el pavimento. Se utilizan *kernels* de tamaño  $(\frac{h}{10}, w)$  y un *clip limit* de 0,005. Finalmente, se aplica erosión como última técnica, la que provoca un aumento en el grosor de las grietas, aunque también lo hace con el ruido en el carril, para esto, es necesario utilizarla de manera adecuada. Cabe destacar que es necesario guardar los resultados obtenidos tanto al aplicar *CLAHE* (*Output\_1*), como, al aplicar erosión (*Output\_2*).

## Segunda Fase

La segunda fase, de detección, se basa en la utilización de *histéresis*. Las técnicas que componen esta etapa se pueden apreciar en orden en la Fig. 3.7.

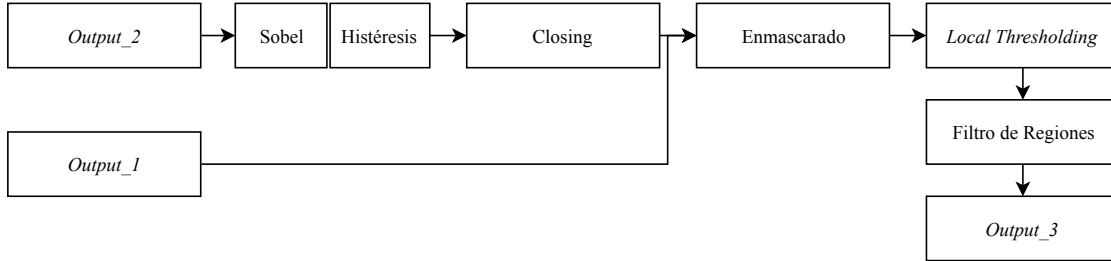


Figura 3.7: Esquema de técnicas utilizadas en la etapa de Detección.

La histéresis cumple la función de detectar los bordes de las grietas presentes en la imagen. Esta técnica detecta la relación entre los píxeles de una imagen (generalmente bordes) utilizando dos umbrales. En la propuesta se realiza una detección de bordes en las grietas utilizando *sobel*, que posteriormente son filtrados con histéresis. Los umbrales bajo y alto para la histéresis se determinan como  $T_L = 5$  y  $T_H = 25$ , respectivamente.

Al identificar los bordes que corresponden a grietas, se utiliza la operación de *closing* para cerrar y juntar los espacios entre los bordes que se encuentren a una distancia cercana, para esto, se utilizó una estructura de disco de tamaño 10. Este proceso crea una máscara, que es utilizada para recuperar la información de la imagen mejorada por *CLAHE* (*Output\_1*), la cual permite tener una mejor visualización de las grietas sin afectar su severidad. Las zonas que no fueron recuperadas por la máscara son rellenadas por la intensidad de moda presente en *Output\_1*.

Posteriormente, se aplica la operación de *local thresholding* para diferenciar la grieta del pavimento de manera binaria, esto se implementa con un *kernel* de tamaño 25, es decir, toma una decisión en relación a los 25 vecinos más cercanos del píxel en cuestión. Además, se filtran las regiones con áreas pequeñas que puedan incluir ruido a la detección.

### 3.5.2. Red *pix2pix*

El segundo método propuesta para resolver el problema de detección, plantea el entrenamiento de la red GAN *pix2pix*.

#### Arquitectura

La arquitectura de la red *pix2pix* se implementó sin cambios respecto a la original presentada en [19] y presentada en la Fig. 2.13.

## Entrenamiento

El set de datos utilizado fue el presentado en la Sección 3.3.2, el cual contiene 15421 pares de imágenes *range* y su *binarización*, ambas, de tamaño  $256 \times 256$  [px]. El entrenamiento de la red se realizó por 200 épocas. En cada época se presentan a la red 12336 imágenes, las cuales fueron divididas en *batches* de tamaño 5. Además, se utilizó un  $\lambda = 100$  en el *loss* del generador. Un esquema resumen de lo mencionado se puede presenciar en la Fig. 3.8.

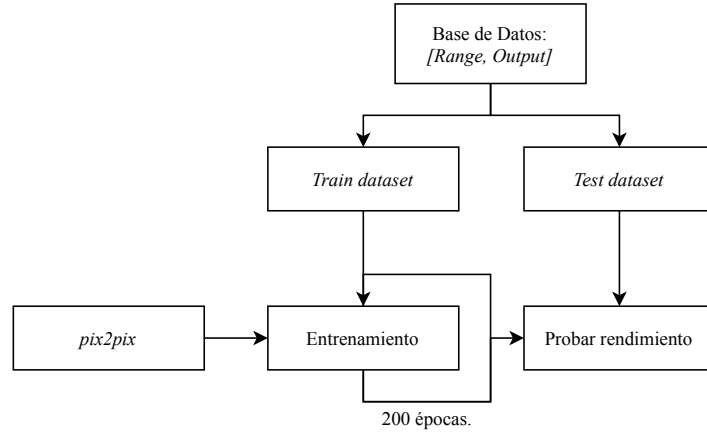


Figura 3.8: Esquema de entrenamiento de red *pix2pix*.

Finalmente, un ejemplo simplificado de como funciona la red se puede apreciar en la Fig. 3.9.

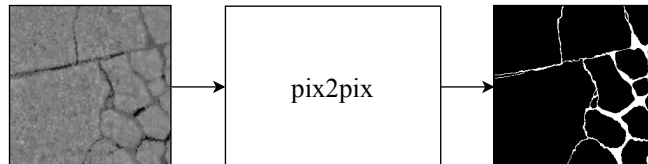


Figura 3.9: Ejemplo de funcionamiento de red *pix2pix*.

## 3.6. Etapa de Clasificación

### 3.6.1. Análisis de Componentes Conectadas

La primera metodología propone clasificar una grieta a partir de las características presentes en cada región. En [29] se propone una metodología para detectar bifurcaciones en imágenes vasculares, la cual es adaptada a la memoria.

Dada las posibles discontinuidades que pueda presentar una grieta, se disminuye la resolución de estas, así, se pueden eliminar las discontinuidades pequeñas en una grieta. En la metodología, se disminuye la resolución original de la grieta a una

10 veces menor. La lógica utilizada para realizar el escalado consiste en analizar la imagen en ventanas de tamaño  $10 \times 10$  [px]. Si dentro de esta ventana existe al menos 1 [px] correspondiente a grieta, toda la ventana es marcada como grieta. Así, al disminuir la resolución de esta nueva imagen no se perderá información de grietas.

Una vez obtenida la información de las grietas en una resolución menor, se obtiene el *esqueleto* de cada grieta, así, se elimina cualquier pixel que no entregue información relevante para la clasificación de esta. El esqueleto se puede obtener fácilmente utilizando los métodos de *skeletonize*.

De forma simplificada, se puede determinar que una grieta de fatiga es una grieta que posee 1 bifurcación o más, así, cualquier grieta que no cumpla esta condición, es transversal o longitudinal dada su orientación. Si bien esto es válido, es necesario considerar la posibilidad de que dos (o más) grietas de distinta clase se conecten en un punto, por ejemplo, una grieta de fatiga con una grieta longitudinal. Dado estos casos, es necesario realizar un análisis extra.

Para determinar la cantidad de bifurcaciones que posee una grieta, basta con recorrer el esqueleto de esta, cada pixel  $(x_p, y_p)$  se considera una bifurcación si cumple las siguientes condiciones:

1. Al seleccionar una ventana de tamaño  $3 \times 3$ , centrada en  $(x_p, y_p)$ , su vecindario posee al menos 3 pixeles de clase grieta **no colindantes** entre ellos.
2. Al dibujar una circunferencia de radio 5 [px] y centrada en  $(x_p, y_p)$ , esta intercepta con al menos 3 pixeles que corresponden al mismo esqueleto del pixel de la grieta analizada.

Estas condiciones se pueden apreciar en las figuras 3.10 y 3.11, respectivamente.

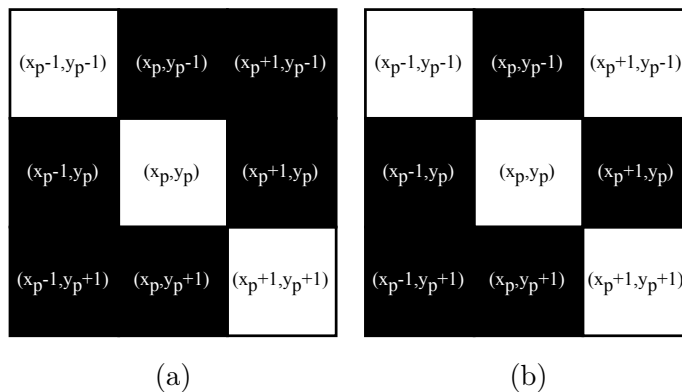


Figura 3.10: Ejemplo para condición 1. (a) Ejemplo de ventana para pixel  $(x_p, y_p)$  que no es bifurcación, (b) Ejemplo de ventana para pixel  $(x_p, y_p)$  que si es una bifurcación.

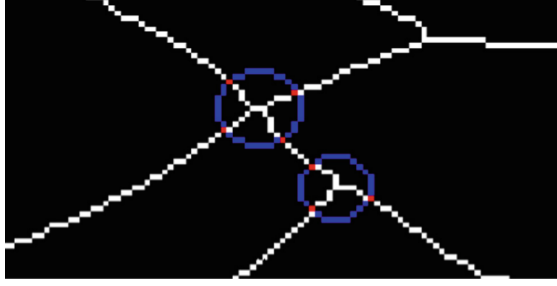


Figura 3.11: Ejemplo de intersecciones posibles al trazar un círculo centrado en un pixel  $(x_p, y_p)$  que corresponde a una bifurcación [29].

### Clasificación

Como ya se mencionó, el conteo de bifurcaciones tan solo determina si una grieta es (en base) de fatiga o no, es por esto, que además de esta información, es necesario analizar la orientación de una grieta. Esta, se puede obtener al recuperar la mínima elipse que envuelve una región o grieta, así, la orientación de la elipse determinara si la grieta analizada es transversal o longitudinal.

Sea  $g_i$  una grieta  $i$ ,  $\gamma_{g_i}$  la cantidad de bifurcaciones que posee y  $\theta_{g_i}$  la orientación de la mínima elipse que la envuelve:

1. Si  $\gamma_{g_i} \geq \gamma_f$ ,  $g_i$  es una grieta de fatiga.
2. Si  $\gamma_{g_i} < \gamma_f$  y  $\theta_{g_i} \geq 45^\circ$ ,  $g_i$  es una grieta longitudinal.
3. Si  $\gamma_{g_i} < \gamma_f$  y  $\theta_{g_i} < 45^\circ$ ,  $g_i$  es una grieta transversal.

Posteriormente, es necesario realizar un análisis extra, pues, como ya se explico al comienzo de la metodología, puede existir la posibilidad de que dos (o más) grietas de distinta clase posean un pixel común. De ser este el caso, necesariamente esta grieta fue clasificada (en base) como una grieta de fatiga, ya que, este pixel común sería una bifurcación. Para considerar estos casos, se eliminan todos los pixeles correspondientes a bifurcaciones en las grietas de fatiga y se analizan todas las secciones de grieta que componen una grieta de fatiga.

Finalmente, sea una  $s_{i,j}$  la sección  $j$  de una grieta de fatiga  $i$ ,  $\theta_{s_{i,j}}$  su orientación y  $l_{s_{i,j}}$  el largo de la grieta:

1. Si  $l_{s_{i,j}} \geq 3$  y  $\theta_{s_{i,j}} \geq 70^\circ$ ,  $s_{i,j}$  es una sección de grieta longitudinal.
2. Si  $l_{s_{i,j}} \geq 3$  y  $\theta_{s_{i,j}} \leq 20^\circ$ ,  $s_{i,j}$  es una sección de grieta transversal.
3. Cualquier otra condición es considerara como una sección de grieta de fatiga.

### 3.6.2. CNN

El segundo método consiste en utilizar una red convolucional como clasificador. La red recibe como entrada una imagen de  $256 \times 256$  [px], la cual, puede contener ninguna, una o más grietas. Finalmente, la red entrega un resultado numérico que indica una clase para la imagen de entrada. Cabe recordar que existen 4 posibles resultados:

- 0: Pavimento.
- 1: Fatiga.
- 2: Transversal.
- 3: Longitudinal.

#### Arquitectura

La CNN a utilizar esta compuesta por 4 capas convolucionales, 4 capas de *max pooling*, una *flatten* y dos *dense*. Esta estructura se puede apreciar en detalle en la Tabla 3.3.

Tabla 3.3: Detalle de estructura de CNN para clasificación.

Layer	Output
Capa Convolucional	(None, 256, 256, 32)
Max Pooling	(None, 128, 128, 32)
Capa Convolucional	(None, 128, 128, 64)
Max Pooling	(None, 64, 64, 64)
Capa Convolucional	(None, 64, 64, 128)
Max Pooling	(None, 32, 32, 128)
Capa Convolucional	(None, 32, 32, 256)
Max Pooling	(None, 16, 16, 256)
Flatten	(None, 65536)
Dense	(None 512)
Dense	(None, 1)

#### Entrenamiento

El set de datos utilizado fue el presentado en la Sección 3.3.3, el cual contiene 7674 imágenes de tamaño  $256 \times 256$  [px] y su respectiva clase. El entrenamiento de la red se realizó por 50 épocas. En cada época se presentan a la red 6139 imágenes, las cuales fueron divididas en *batches* de tamaño 128.

### 3.7. Etapa de Análisis de Severidades

El análisis de severidades es independiente del método de detección o clasificación utilizado. Para realizar este proceso, es necesario definir cuantos pixeles determinarán el tipo de severidad. La escala de conversión fue definida como:  $1[px] = 4[mm]$ . Así, para grietas de ancho menor a  $1[px]$  se considera severidad baja, entre  $1[px]$  y  $5[px]$  severidad media, y finalmente, mayor a  $5[px]$ , severidad alta (Tabla 3.4).

Tabla 3.4: Definición de severidades en milímetros y pixeles.

	Severidad		
	Baja	Media	Alta
Milímetros	$< 3$	$\geq 3$ y $< 20$	$\geq 20$
Pixeles	$= 1$	$> 1$ y $\leq 5$	$> 5$

Para lograr este objetivo, el algoritmo calcula la distancia más cercana que tiene cada punto en una grieta al pavimento (en pixeles). Aquí, es necesario hacer un análisis extra, pues, la distancia calculada implica una distancia mínima en el vecindario del pixel analizado. Así, un pixel de distancia  $3[px]$  es considerado como severidad alta, pues, significa que posee al menos  $2[px]$  de grieta en cada uno de sus vecindarios (lo que implicaría tener un ancho mínimo de  $5[px]$ ). Esto se puede apreciar mejor en la Fig. 3.12.

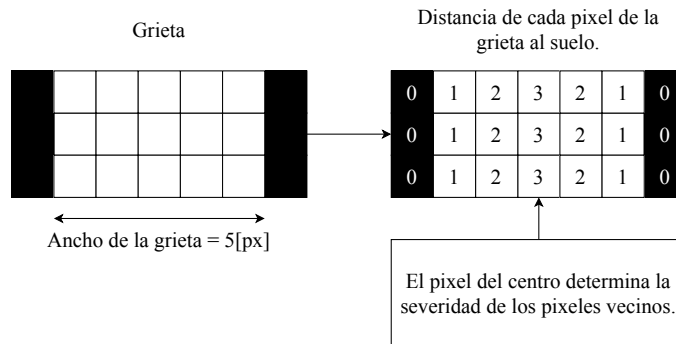


Figura 3.12: Análisis de severidad dada la distancia de entre pixel de grieta y pavimento.

### 3.8. Post-Procesamiento



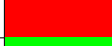
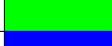




#### 3.8.1. Visualización

Los resultados de las etapas anteriores: detección, clasificación y análisis de severidades, son presentados en distintas visualizaciones. Para la clasificación se



utilizaran los colores rojo, verde y azul para representar las clases de fatiga, transversal y longitudinal respectivamente. Mientras que, para el análisis de severidades se presentará un mapa de calor en tres niveles (para los tres tipos de severidades), así, entre más claro es el color en el mapa, mayor es su severidad. Finalmente, un resumen de los colores mencionados se puede apreciar en la Tabla 3.5.

Tabla 3.5: Colores para cada clase y severidad.

		Color
Detección	Pavimento	
	Grieta	
Clase	Fatiga	
	Transversal	
	Longitudinal	
Severidad	Baja	
	Media	
	Alta	

# Capítulo 4

## Resultados y Discusión

### 4.1. Resultados

A continuación se presentan los resultados y métricas de evaluación de desempeño utilizados para las distintas etapas y metodologías desarrolladas. Específicamente, para la etapa de detección, los sistemas se evaluaron utilizando *Dice Coefficient* (DC) con distintas áreas precisión. Para la etapa de clasificación, las propuestas son evaluadas utilizando una matriz de confusión. Finalmente, para la etapa de análisis de severidades, se entrega la opinión de un experto en severidades para evaluar los resultados obtenidos. Para cada etapa y propuesta, se presentan ejemplos gráficos de los resultados obtenidos.

Es necesario especificar que las métricas presentadas en este capítulo fueron realizadas en un computador con las siguientes características principales: CPU I5-8400, RAM 32GB y GPU Nvidia 1070ti Hybrid de 8GB.

#### 4.1.1. Etapa de Detección

En esta sección se presentan los resultados obtenidos en la primera etapa de la memoria. La evaluación se realiza sobre las bases de datos CRACK256 y CRACK10M. En la Tabla 4.1 se encuentran los resultados para la métrica de *Dice Coefficient* ( $DC_n$ ), con  $n$ : el nivel áreas de precisión exigido en la medición. Además, en la Tabla 4.2, se presenta el tiempo promedio que tarda en detectar cada propuesta (P.d.I o Procesamiento de Imágenes, y pix2pix).

Tabla 4.1: *Dice Coefficient* en etapa de detección.

BD	Propuesta	DC <sub>1x1</sub>	DC <sub>8x8</sub>	DC <sub>16x16</sub>
CRACK256 <sub>test</sub>	pix2pix	0.52 ± 0.19	0.75 ± 0.21	0.80 ± 0.20
	P.d.I.	-	-	-
CRACK10M <sub>test</sub>	pix2pix	0.59 ± 0.09	0.79 ± 0.05	0.81 ± 0.06
	P.d.I.	0.51 ± 0.09	0.69 ± 0.05	0.72 ± 0.12

Tabla 4.2: Tiempo en etapa de detección.

BD	Propuesta	Tiempo [s]
CRACK256	pix2pix	0.04 ± 0.00
	P.d.I.	-
CRACK10M	pix2pix	1.58 ± 0.03
	P.d.I.	7.26 ± 0.32

## Visualizaciones

A continuación, en las Figuras 4.1, 4.2, 4.3 y 4.4 se presentan los resultados visuales obtenidos para las propuestas de Procesamiento de Imágenes y *pix2pix*.

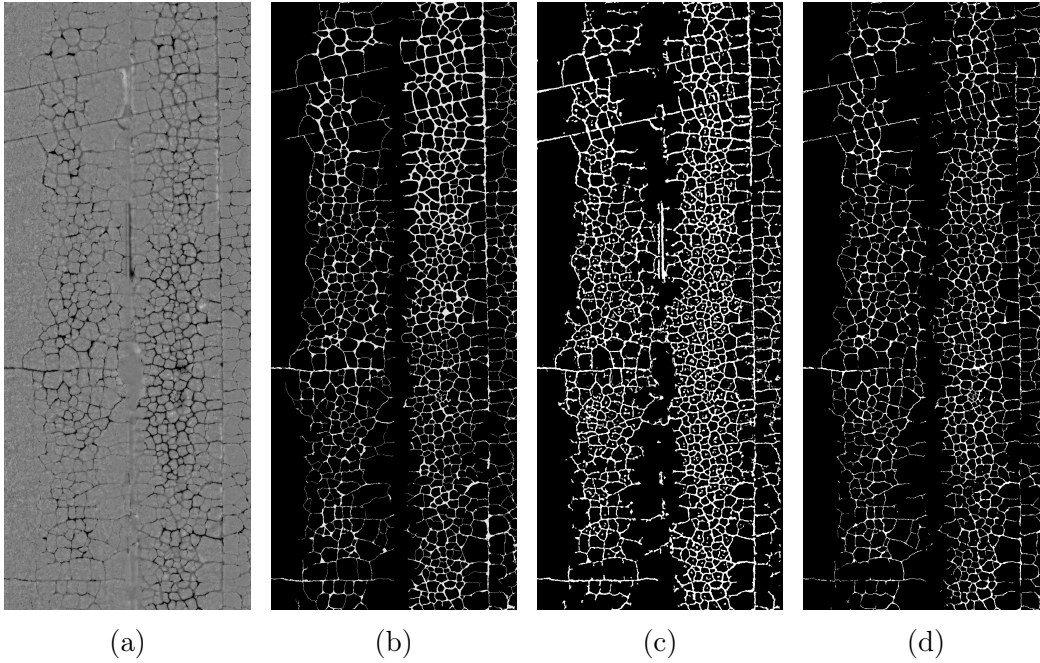


Figura 4.1: Ejemplo 1. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados Procesamiento de Imágenes, (d) Resultados *pix2pix*.

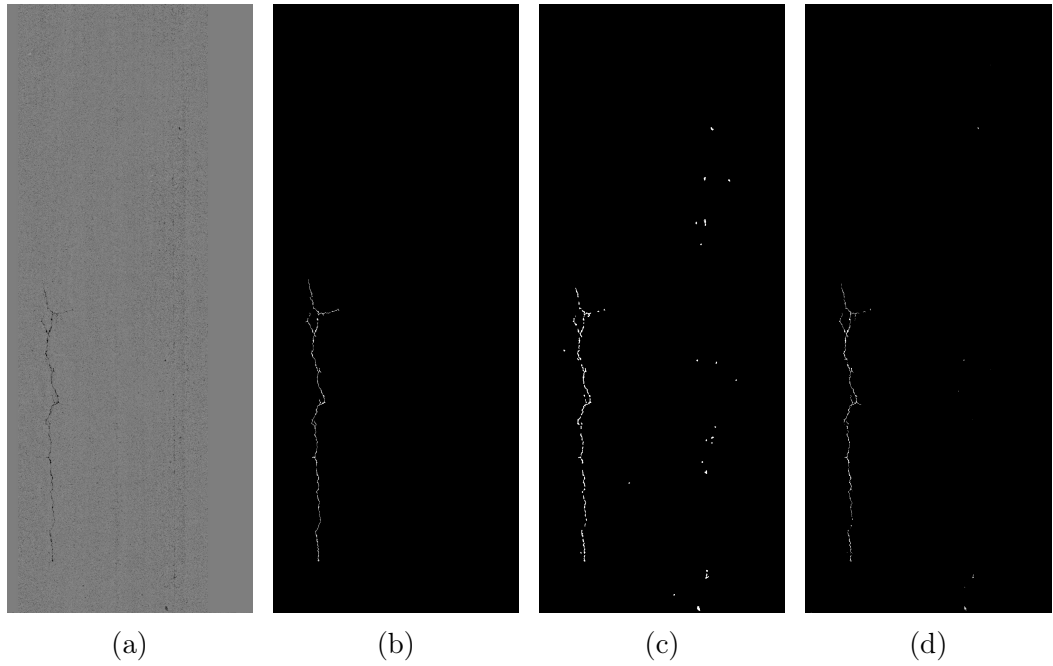


Figura 4.2: Ejemplo 2. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados Procesamiento de Imágenes, (d) Resultados *pix2pix*.

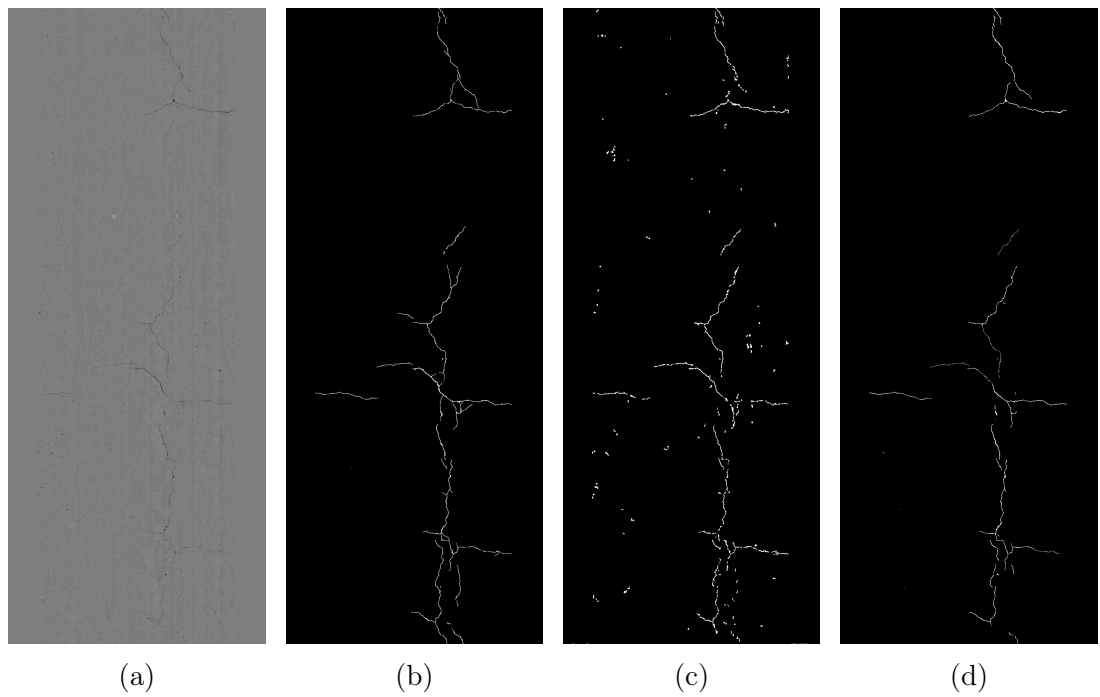


Figura 4.3: Ejemplo 3. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados Procesamiento de Imágenes, (d) Resultados *pix2pix*.

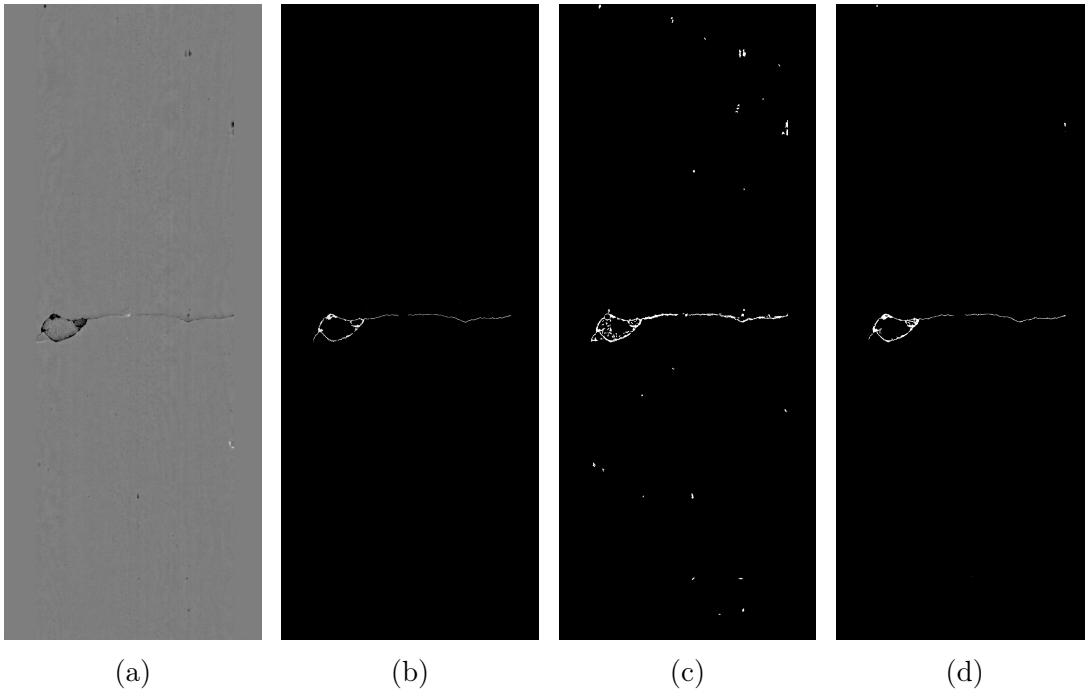


Figura 4.4: Ejemplo 4. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados Procesamiento de Imágenes, (d) Resultados *pix2pix*.

#### 4.1.2. Etapa de Clasificación

En esta sección se presentan los resultados obtenidos en la segunda etapa de la memoria. En las Tablas 4.3 y 4.4 se presentan las matrices de confusión de la clasificación obtenida por las propuestas de Análisis de CCO y CNN.

Tabla 4.3: Matriz de Confusión para clasificación obtenida con Análisis de CCO.

		Predicción			Precisión [%]
		Fatiga	Transversal	Longitudinal	
Real	Fatiga	95.8 %	2.6 %	1.6 %	96 %
	Transversal	21.5 %	78.0 %	0.5 %	78 %
	Longitudinal	31.1 %	1.5 %	67.4	67 %
					<b>80 %</b>

Tabla 4.4: Matriz de Confusión para clasificación obtenida con CNN.

		Predicción			Precisión [%]
		Fatiga	Transversal	Longitudinal	
Real	Fatiga	92.2 %	2.8 %	5 %	92 %
	Transversal	27.0 %	69.2 %	3.8 %	69 %
	Longitudinal	42.6 %	5.8 %	51.6 %	52 %
					<b>71 %</b>

Además, en la Tabla 4.5 se presentan los tiempos promedio que tarda cada propuesta.

Tabla 4.5: Tiempo en etapa de clasificación.

	Tiempo [s]
CNN	$0.31 \pm 0.01$
CCO	$0.63 \pm 2.52$

## Visualizaciones

A continuación, en las Figuras 4.5, 4.6, 4.7 y 4.8 se presentan los resultados visuales obtenidos para las propuestas de CNN y Análisis de CCO.

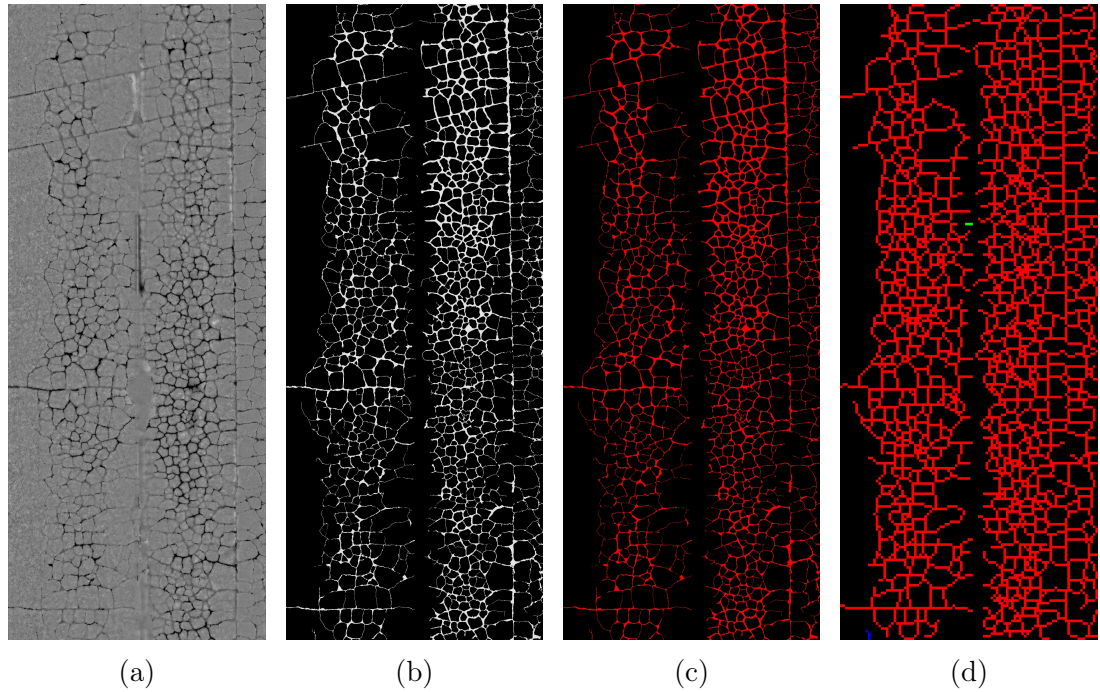


Figura 4.5: Ejemplo 1. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados CNN, (d) Resultados Análisis de CCO.

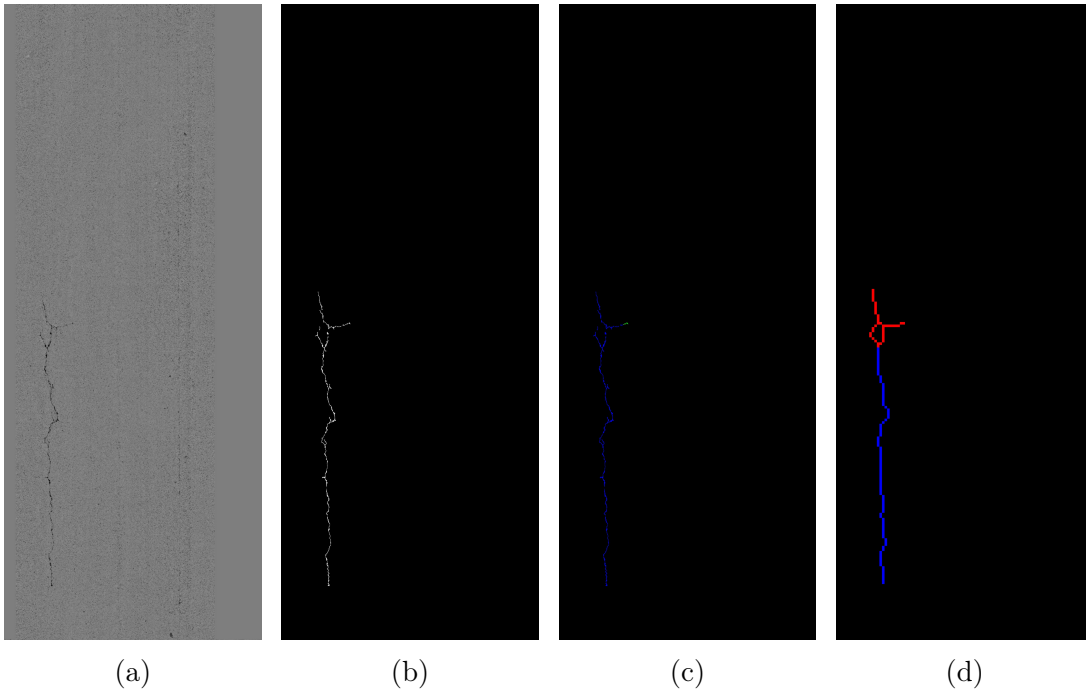


Figura 4.6: Ejemplo 2. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados CNN, (d) Resultados Análisis de CCO.

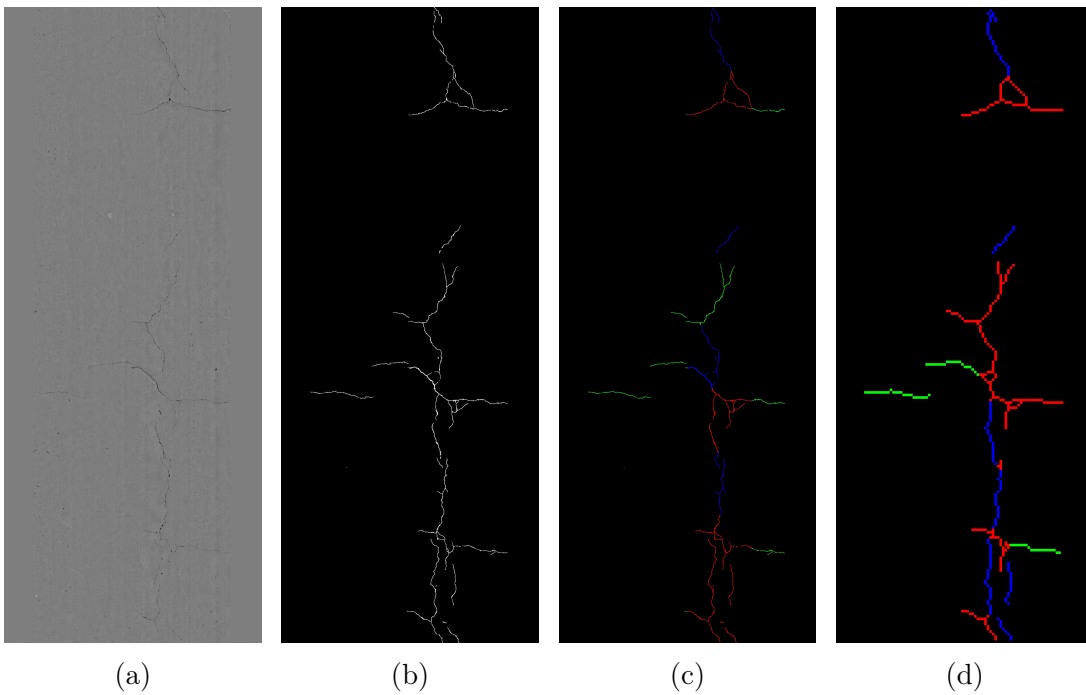


Figura 4.7: Ejemplo 3. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados CNN, (d) Resultados Análisis de CCO.

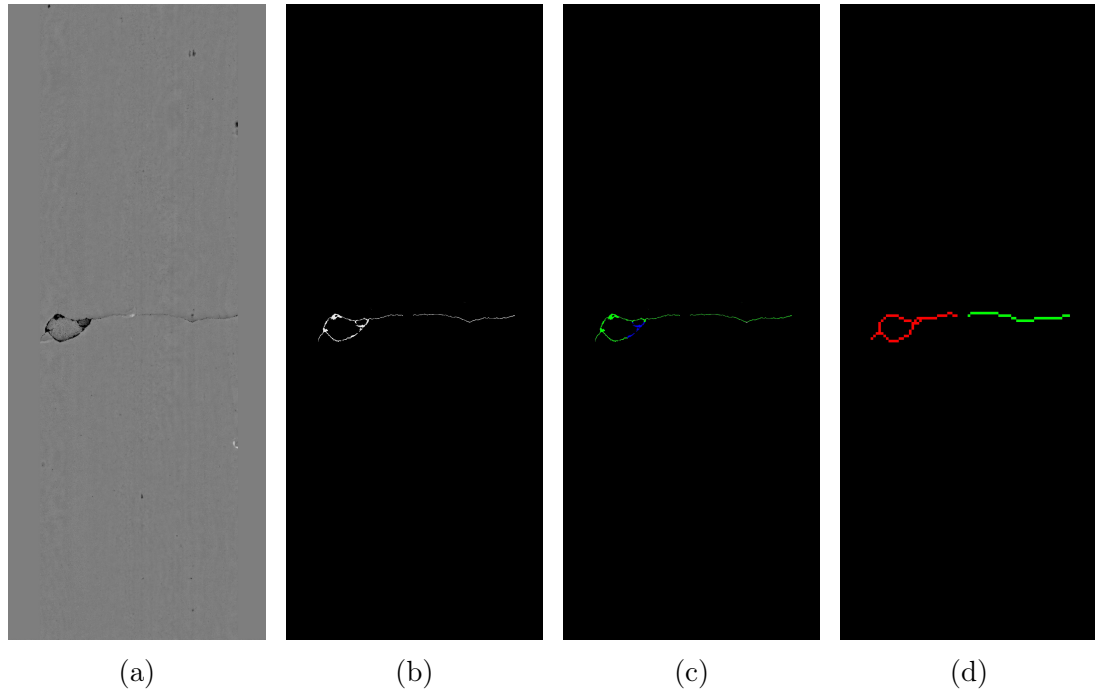


Figura 4.8: Ejemplo 4. (a) Imagen *range*, (b) Detección Esperada, (c) Resultados CNN, (d) Resultados Análisis de CCO.

### 4.1.3. Etapa de Análisis de Severidades

En esta sección se presentan los resultados obtenidos en la tercera etapa de la memoria.

La propuesta realizada para esta etapa se constituye por una medición, y, no se posee una base de datos como punto de comparación y medición de resultados. Dadas estas condiciones, es necesario considerar la opinión de un experto en el área de análisis de severidades de grietas.

Ricardo Garrido, Sub-Gerente de Ingeniería en APSA Ltda. realizó una evaluación de los resultados presentados por esta propuesta, considerando los siguientes puntos:

- Cerca del 92 % de las imágenes evaluadas presentan resultados completamente satisfactorios.
- Existen algunas dudas respecto a la clasificación entre evaluaciones de baja y media severidad.

Finalmente, en la Tabla 4.6 se puede apreciar el tiempo promedio de procesamiento para esta propuesta.



Tabla 4.6: Tiempo en etapa de análisis de severidades.

Tiempo [s]
$0.22 \pm 0.02$

## Visualizaciones

A continuación, en las Figuras 4.9, 4.10, 4.11 y 4.12 se presentan los resultados visuales obtenidos para el algoritmo de Análisis de Severidad.

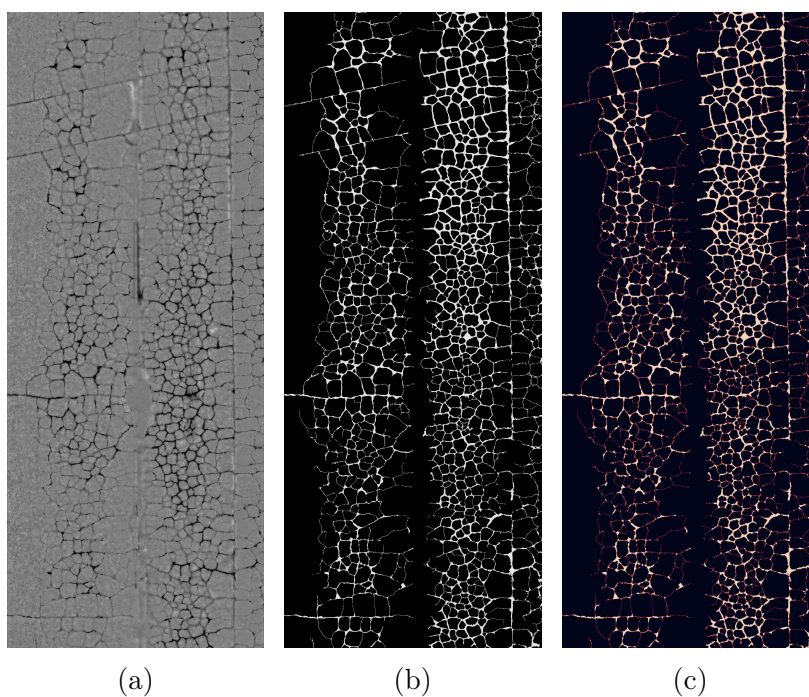


Figura 4.9: Ejemplo 1. (a) Imagen *range*, (b) Grietas presentes en (a), (c) Resultados Análisis de Severidad

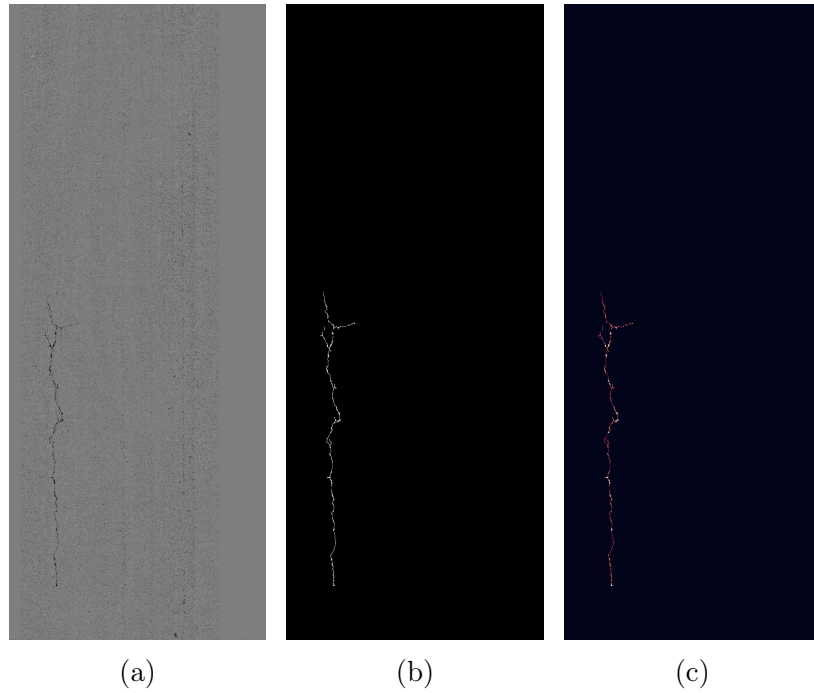


Figura 4.10: Ejemplo 2. (a) Imagen *range*, (b) Grietas presentes en (a), (c) Resultados Análisis de Severidad

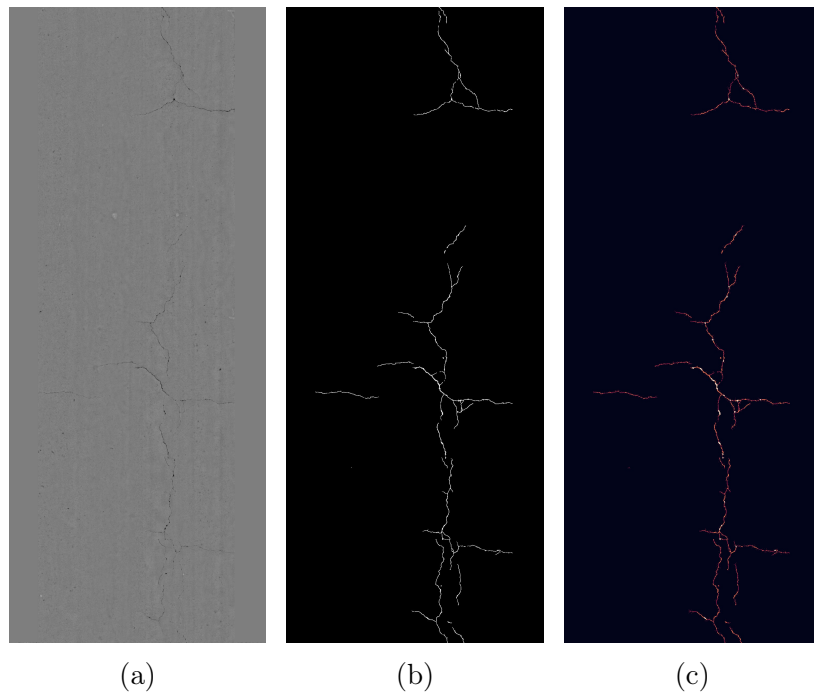


Figura 4.11: Ejemplo 3. (a) Imagen *range*, (b) Grietas presentes en (a), (c) Resultados Análisis de Severidad

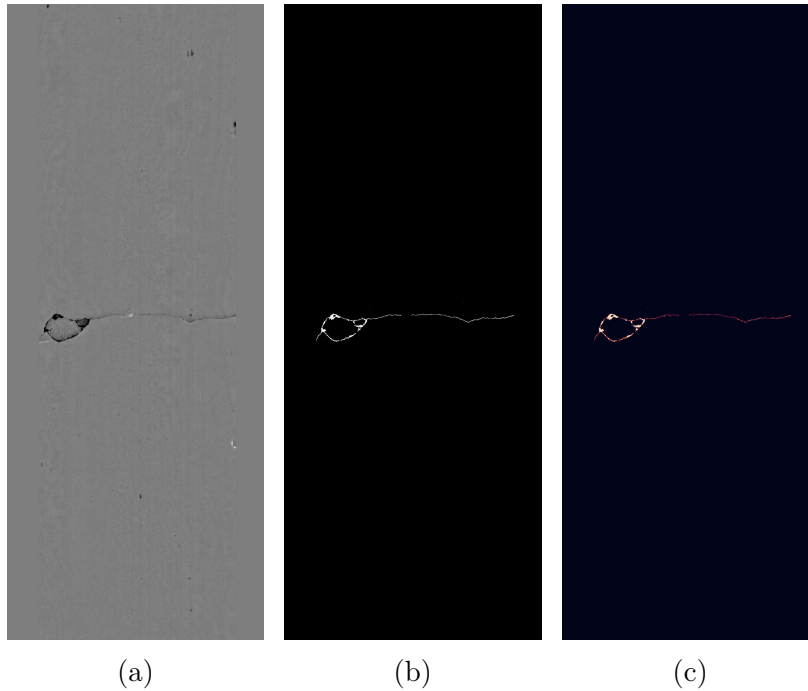


Figura 4.12: Ejemplo 4. (a) Imagen *range*, (b) Grietas presentes en (a), (c) Resultados Análisis de Severidad

## 4.2. Discusión

A continuación se discuten los resultados obtenidos para cada propuesta en cada etapa. Además, se contrastan los resultados visuales con las métricas proporcionadas. Finalmente, también se mencionarán las ventajas y desventajas de cada propuesta.

### 4.2.1. Etapa de Detección

#### Propuesta 1: Procesamiento de Imágenes

Respecto a la primera propuesta para el problema de detección, se puede apreciar que la métricas obtenidas no fueron mucho más bajas en comparación a la segunda propuesta. Si bien, se esperaba que los resultados fueran inferiores, al observar los resultados de manera visual se puede apreciar que el rendimiento de la propuesta no es insatisfactorio, al contrario, la propuesta presenta una gran capacidad de detección en grietas, pero, posee problemas de precisión, ya que, las grietas detectadas son más anchas de lo que debería, lo cual, es sumamente importante para el posterior análisis de severidades. Lo mencionado se puede apreciar en el ejemplo presentado en la Fig. 4.13, donde, a pesar de que la propuesta detecto gran mayoría de las grietas esperadas, estas no fueron precisas.

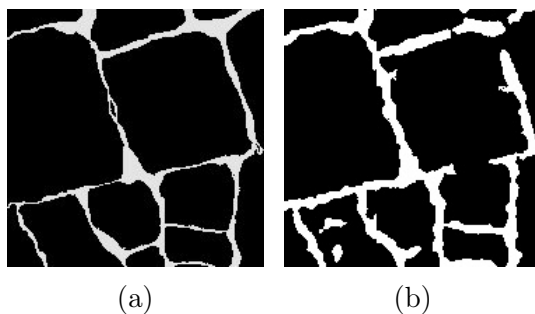


Figura 4.13: Ejemplo de detección. (a) Detección esperada, (b) Detección obtenida con PdI.

Además de la imprecisión mencionada, la propuesta también presenta problemas al momento de filtrar el ruido presente. Esto se puede apreciar con claridad en los ejemplos presentados en la Fig. 4.14. Las características mencionadas influyen directamente en la baja evaluación obtenida por las métricas.

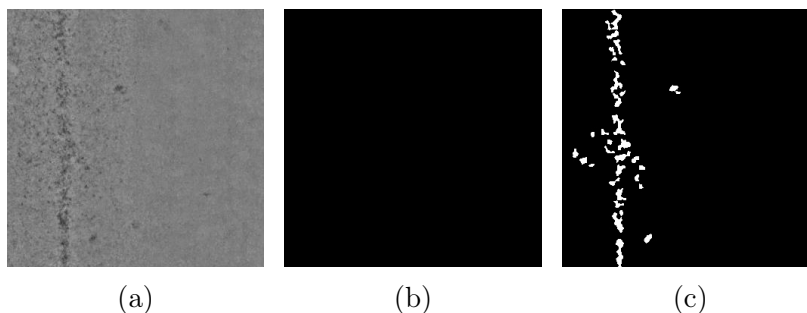


Figura 4.14: Ejemplo de detección ruidosa. (a) Imagen *Range*, (b) Detección esperada, (c) Detección obtenida con PdI.

Por otro lado, la aplicación de una serie de técnicas de procesamientos de imágenes en bloque provoca un aumento en el tiempo de procesamiento, además, técnicas como *TV Chambolle* son dependientes a la cantidad de ruido presente en la imagen, por lo que, si una imagen es muy ruidosa, tardará más de lo esperado. Esta característica puede tener problemas en áreas del norte del país, donde, como se mencionó en la Introducción de la memoria, la carretera suele estar cubierta por capas de arena, que los sensores del dispositivo LCMS detectan como ruido. En la Fig. 4.15 se presentan un ejemplo de carretera con y sin ruido en el asfalto.

Finalmente, es importante mencionar que la gran desventaja de esta metodología es que no realiza una interpretación sobre las imágenes presentadas. Esto afecta en ocasiones donde pueden aparecer manchas, superficies desniveladas u otras situaciones, en las cuales, un cambio en la tonalidad del asfalto o grises más oscuros no representan una grieta y aún así, son detectadas por la propuesta, pues no sabe interpretarlas como tal.

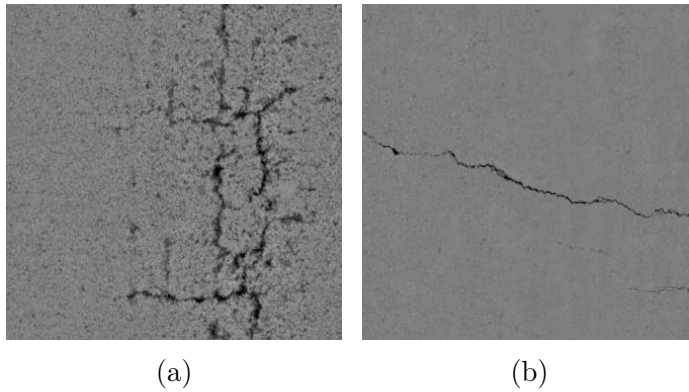


Figura 4.15: Ejemplo de grietas en carreteras con y sin ruido. (a) Imagen *Range* con ruido, (b) Imagen *Range* sin ruido.

## Propuesta 2: Red *pix2pix*

Respecto a la segunda propuesta para el problema de detección, se puede apreciar que las métricas obtenidas fueron altas incluso para niveles de áreas de precisión bajas, como  $8 \times 8$ . La diferencia en la medición para áreas de  $1 \times 1$  y  $8 \times 8$  indica que las grietas que debían ser detectadas, fueron en su mayoría detectadas, pero, no con una precisión exacta.

Aun así, si bien las métricas presentadas en la Tabla 4.1 son bastante altas, no presentan resultados completamente satisfactorios para la detección. Estos, se puede asociar a dos factores, los cuales se enumeran a continuación:

1. **Base de Datos:** Como se menciona en este trabajo, la base de datos utilizada para entrenar la red *pix2pix* y posteriormente evaluarla, fue realizada manualmente solo para fines de la memoria. Ante esta situación, pueden haber errores en la exactitud de algunos grosores y grietas a detectar deseadas. A esto, se suma la cantidad de imágenes utilizadas en el entrenamiento, si bien se utilizaron técnicas de *Data Augmentation* para incrementar el tamaño de la base de datos, esto no es comparable con tener una base de datos realmente más extensa y completa. Las técnicas utilizadas tan solo cumplen la función de repetir una misma imagen, modificándola de tal manera que parezca ser una distinta, pero, sigue siendo la misma. La red necesita de muchas más imágenes para poder obtener resultados realmente satisfactorios.
2. **Diversidad de Grietas:** Algo que va de la mano con el punto anterior, es la diversidad de grietas presentes en asfalto. En esta memoria se trataron de abordar la mayor cantidad de formas, severidades y profundidades de grietas, pero, el detectarlas todas, implica tener una base de datos mucho más extensa, que puedan representar de manera contundente esta diversidad. En las Figuras 4.16 y 4.17 se puede apreciar como la red funciona de forma muy satisfactoria para grietas con una profundidad notoria, mientras que, para grietas que son casi imperceptibles (incluso para el ojo humano), la red no es capaz de detectarlas con mucha precisión. Esta diversidad provoca una

variedad en los resultados, y por lo tanto, una disminución en el porcentaje de precisión presentado.

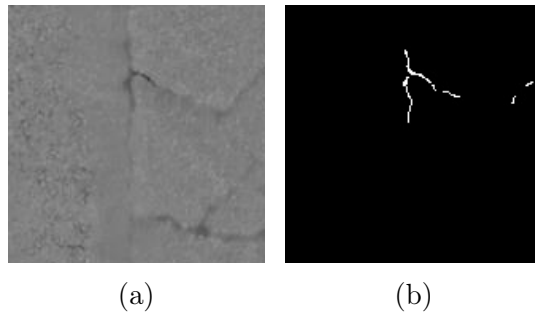


Figura 4.16: Ejemplo de grieta con poca profundidad. (a) Imagen *Range*, (b) Detección obtenida por la red *pix2pix*.

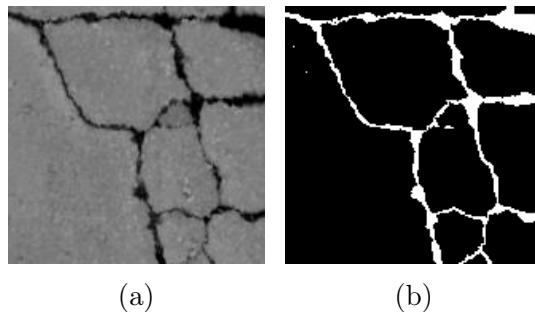


Figura 4.17: Ejemplo de grieta con alta profundidad. (a) Imagen *Range*, (b) Detección obtenida por la red *pix2pix*.

Por otra parte, al observar los resultados visuales, se puede apreciar como la precisión en la detección es muchísimo más exacta en comparación al método de Procesamiento de Imágenes, beneficiando las mediciones que se obtendrían con el algoritmo desarrollado para la etapa de Análisis de Severidades. Un ejemplo de esto se puede apreciar en la Fig. 4.18, en la cual, al realizar la misma comparación que con PdI en la Fig. 4.13, la precisión de la red es mucho mayor.

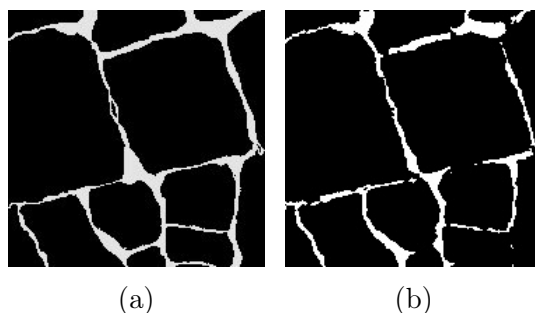


Figura 4.18: Ejemplo de detección. (a) Detección esperada, (b) Detección obtenida con *pix2pix*.

Por otra parte, los resultados obtenidos presentan poco o nada de ruido en relación a la primera propuesta, lo cual no demanda un trabajo de post procesamiento

para eliminarlo. Al igual que el ejemplo anterior, en 4.19 se puede apreciar que al realizar la misma comparación que en 4.14, la red es capaz de diferenciar entre el ruido presente en el asfalto y una grieta.

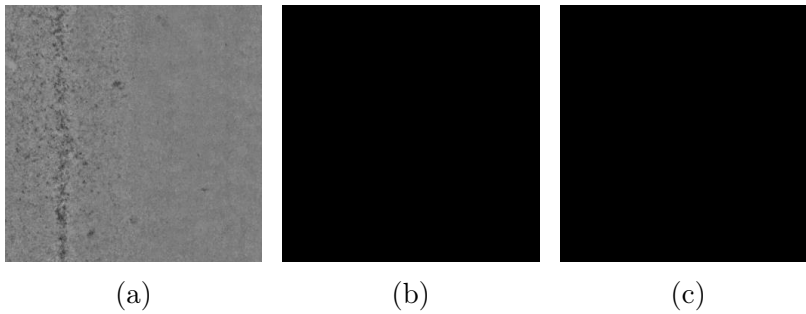


Figura 4.19: Ejemplo de detección ruidosa. (a) Imagen *Range*, (b) Detección esperada, (c) Detección obtenida con *pix2pix*

Una de las grandes ventajas de esta propuesta, es que el tiempo de procesamiento es constante, pues, el tiempo de procesamiento es independiente de la cantidad de grietas, ruido, u otro factores que pueden estar presentes en la imagen a analizar. Siendo capaz de procesar 10 [m] de carretera en aproximadamente 1,6 [s], un tiempo bastante bajo.

Finalmente, utilizar Inteligencia Artificial contrae muchos beneficios, dentro de estos uno de los más importantes es su capacidad de interpretar. La red propuesta es capaz de reconocer si un pixel oscuro corresponde a una grieta, ruido, mancha, etc. dependiendo del contexto en el que se encuentra y no solo de su tonalidad. Además de poder detectar distintos tipos de grietas, formas y contrastes.

## 4.2.2. Etapa de Clasificación

### Propuesta 1: Análisis de CCO

Respecto a la primera propuesta para la etapa de clasificación, en la Tabla 4.3 se puede apreciar alto porcentaje de precisión al clasificar grietas de fatiga y una disminución en la clasificación de grietas transversales y longitudinales.

Los resultados obtenidos son bastante satisfactorios, pues, el método propuesto cumple con las características deseadas para la clasificación de grietas, incluso, si esta falla. Un acierto del 96[%] para las grietas de fatiga es altamente satisfactorio, por lo que, no se entrará en detalle respecto a esa clasificación.

Por otro lado, los resultados para la clasificación de grietas transversales y longitudinales, a pesar de ser más bajos, tienen mucho sentido y son satisfactorios por las razones a continuación enumeradas:

1. En la clasificación existen dos situaciones en las cuales se puede considerar que una grieta es longitudinal o transversal.
  - Primero, que sea una grieta sin bifurcaciones, lo cual indica que no es de fatiga. De ser así, la propuesta determina a partir de su inclinación si esta es longitudinal o transversal. En caso de equivocarse, el error asociado es mínimo, pues, en términos de medición y evaluación, una grieta longitudinal y transversal tienen la misma unidad de medida [m], mientras que, una grieta de fatiga se mide en [m<sup>2</sup>].
  - Segundo, que sea una extensión de una grieta de fatiga, es decir, que nazca o viva en esta. En este caso, la propuesta siempre tiene presente que la base de la grieta detectada es una grieta de fatiga y determina a partir del tamaño y orientación de esta extensión si corresponde a una grieta longitudinal o transversal. El error puede presentarse en casos donde la grieta sea clasificada como fatiga en vez de los otros casos, bajo esta condición, se entendería que la extensión es muy pequeña, y por lo tanto, el error asociado es pequeño. También, pudo haber sido clasificada de manera errónea entre longitudinal o transversal, lo cual volvería al primer caso mencionado.

Una de las pequeñas desventajas que tiene utilizar esta propuesta es el tiempo de procesamiento no constante. La metodología utilizada, detecta las bifurcaciones y analiza todas las secciones de CCO relacionadas con estas, lo que la hace dependiente de la cantidad presente en cada una de las imágenes. Por otra parte, esta desventaja no tiene gran impacto, pues, no son muchas las imágenes que contienen grandes cantidades de bifurcaciones, al contrario. Es por esto, que a pesar de lo mencionado, la propuesta tiene un tiempo promedio de 0.6[s], un tiempo absolutamente aceptable. En la Fig. 4.20 se pueden apreciar dos casos, una imagen con gran cantidad de bifurcaciones y una con pocas, además, en la Tabla 4.7, se puede apreciar la diferencia en el tiempo de procesamiento asociado a estas imágenes.

Tabla 4.7: Comparación en tiempos de procesamiento en imágenes con poca y muchas bifurcaciones.

	Tiempo
Fig. 4.20a	0.3 [s]
Fig. 4.20b	22.3 [s]

Finalmente, también se puede apreciar en los resultados obtenidos la diferencia cuantitativa en la cantidad de ejemplos analizados para cada clase, lo cual, también provoca un pequeño sesgo en los resultados de la clasificación.



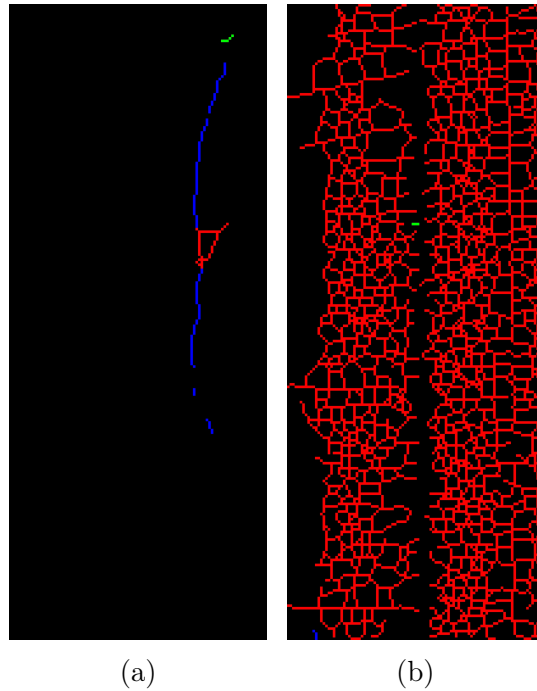


Figura 4.20: Ejemplo de diferencias de bifurcaciones. (a) Grietas con pocas bifurcaciones, (b) Grietas con muchas bifurcaciones.

## Propuesta 2: CNN

Respecto a los resultados obtenidos mediante el uso de Redes Convolucionales, se puede apreciar que si bien los resultados no son insatisfactorios, pues, en la Tabla 4.4 se puede observar una precisión promedio del 71[%] y hasta un 92[%] en grietas de fatiga, no son del todo precisos.

Una de las desventajas de esta propuesta, es que, la clasificación de grietas se realiza por áreas. Este funcionamiento puede provocar errores al momento de clasificar áreas que contengan información de más de una grieta, pues, todos los pixeles que convivan en el área analizada, son clasificadas bajo la misma clase. Así, la clase predominante en un área es la que determinara como clasificar todo el conjunto. Esto, puede traer problemas de medición no controlables como en la primera propuesta. Ejemplos de estas situaciones se pueden apreciar en las Figuras 4.21a y 4.21b, en la primera, se puede apreciar como la red no logra detectar la grieta transversal que nace de la fatiga, mientras que, en la segunda, no logra determinar donde empieza o termina un tipo de grieta, pues clasifica un área completa.

Por otro lado, una de las grandes ventajas, tal como la red *pix2pix* para la primera etapa, es su capacidad de procesamiento constante, tardando aproximadamente 0.3[s] en clasificar 10[m] de carretera, un tiempo bastante bajo para estas dimensiones.

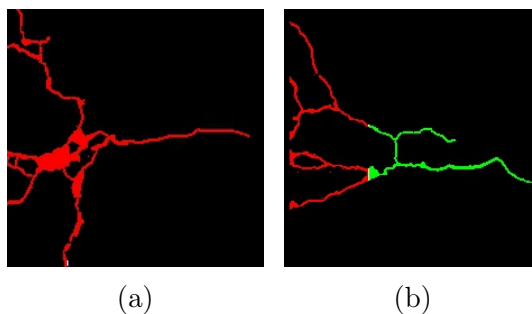


Figura 4.21: Ejemplos de problemas al clasificar por área con CNN. (a) Error al identificar distintas grietas, (b) Error al determinar inicio y término de una clase de grieta.

### 4.2.3. Análisis de Severidades

Respecto a los resultados obtenidos para la etapa de análisis de severidades, se puede apreciar que la evaluación del experto destaca una gran efectividad y precisión en la clasificación de estas. Son muy pocos los casos en los cuales se duda si su determinación es correcta.

Es necesario mencionar que la etapa de análisis de severidades se encuentra altamente ligada a la etapa de detección, pues, si la primera realiza detecciones con anchos erróneos, la severidad tendrá resultados no satisfactorios.

Finalmente, se puede apreciar que los tiempos de procesamiento son bastante bajos, tardando cerca de 0.3[s] en analizar 10[m] de carretera.

# Capítulo 5

## Conclusiones y Trabajo Futuro

### 5.1. Conclusiones

El trabajo realizado en la memoria aborda las problemáticas de detección, clasificación y análisis de grietas en imágenes de carreteras de asfalto capturadas por el LCMS. Las propuestas abordadas permiten un avance en la velocidad y precisión a la hora de realizar los planes de conservación de carreteras a nivel mundial.

Para solucionar estas problemáticas, se presentaron propuestas para cada una de las etapas mencionadas. Para la etapa de detección se exploraron las siguientes propuestas:

- **Procesamiento de Imágenes:** Esta propuesta consiste en la implementación en serie de distintas técnicas de procesamiento de imágenes que permiten mejorar la calidad y visualización de las grietas.
- **Red *pix2pix*:** Esta propuesta consiste en la implementación una red generativa adversaria, capaz de extraer las características que representan las grietas presentes en distintos contextos, de esta manera, generar una imagen con la detección deseada.

Posterior a la obtención de resultados y sus respectivos análisis para cada propuesta, se concluye que la mejor propuesta es la segunda, es decir, la red *pix2pix*. Esto, no solo se fundamenta en los desempeños obtenidos, los cuales fueron mejores que la primera propuesta tanto en detección como en tiempo, sino también, por la gran consistencia que posee. Es necesario destacar que la segunda propuesta tuvo un desempeño satisfactorio a pesar de la limitada base de datos utilizada para su entrenamiento y evaluación. A esto, se puede agregar la gran facultad de interpretar y determinar la clase de los distintos píxeles presentes en las imágenes no tan solo por su intensidad, sino también, por el contexto en el que vive.

Para la segunda etapa, de clasificación, se exploraron las siguientes dos propuestas:

- **Análisis de CCO:** La primera propuesta consiste en la implementación de un algoritmo capaz de analizar las componentes conectadas presentes en la detección binarizada de grietas. El algoritmo cuantifica las bifurcaciones presentes y a partir de características morfológicas determina la clase de cada componente.
- **CNN:** La segunda propuesta consiste en la implementación de una red neuronal convolucional de clasificación. La cual extrae características espaciales y determina la clase preponderante en áreas cuadradas de carretera.

De las propuestas presentadas para la etapa de clasificación, se concluye que la primera es la que presenta un mejor desempeño. Esto se fundamenta primero en los resultados obtenidos, en los cuales, el análisis de CCO obtiene un mejor desempeño respecto a la CNN. Además, una característica sumamente importante y que se enfatiza en la discusión de estos resultados, es que: el error asociado al desacierto de clase tiene un efecto mínimo en la evaluación práctica del algoritmo, es decir, incluso cuando la propuesta falla, el error tiene consecuencias aceptables dentro del marco de clasificación y evaluación de grietas en la vida real. Es necesario recalcar el objetivo de implementar las propuestas mencionadas y que están asociadas a una mejor optimización de recursos y tiempos a la hora de evaluar y reparar carreteras. Además, si bien el tiempo tiene un papel fundamental en la evaluación de desempeño, las diferencias entre las propuestas es mínima, pues, el análisis de CCO tan solo tarda, en promedio, 0.3[s] más, algo casi imperceptible.

Para la última etapa, de análisis de severidades, se realizó una propuesta. Esta, consiste en calcular la distancia entre un pixel de grieta al pixel de pavimento más cercano. A partir de esta medición, se expande la distancia central correspondiente al ancho de la grieta y se determina si esta pertenece a una severidad baja, media o alta. Esta metodología tuvo resultados bastante satisfactorios, los cuales fueron corroborados por un experto en el área de análisis de severidades de grietas sobre asfalto. Esta propuesta además de ser precisa, tiene un tiempo promedio constante.

Respecto a los objetivos específicos propuestos para esta memoria, se lograron satisfacer cada uno de ellos, pues:

- Se diseñó e implementó una propuesta para la etapa de detección basada en técnicas de procesamiento de imágenes.
- Se implementó una propuesta para la etapa de detección basada en la arquitectura de red *pix2pix*, la cual tuvo mejor desempeño respecto a la propuesta de procesamiento de imágenes.
- Se construyó de manera manual, una base de datos para el entrenamiento y evaluación de la red *pix2pix*, la cual, utilizando técnicas de *Data Augmentation*, contiene cerca de 13 mil imágenes de grietas con su respectiva *segmentación*.
- Se diseñó e implementó una propuesta para la etapa de clasificación basada en análisis de componentes conectadas, la cual tuvo un mejor desempeño para esta etapa.
- Se diseñó e implementó una propuesta para la etapa de clasificación basada

en una red neuronal convolucional basada en cuatro capas convolucionales y cuatro capas de *max pooling*.

- Se construyó una base de datos para entrenar y evaluar la red convolucional creada, la cual contiene cerca de 8 mil imágenes con su respectivo *label*.
- Se diseñó un algoritmo capaz de evaluar la severidad de las grietas detectadas en la etapa anterior, la cual tuvo un desempeño altamente satisfactorio.
- Se visualizaron los resultados obtenidos para todas las etapas, siendo estos, fáciles de diferenciar y analizar.

Las distintas problemáticas abordadas en cada una de las etapas de la memoria fueron resueltas de manera exitosa. El trabajo realizado en esta memoria contribuye en gran medida en la detección y evaluación de grietas en carreteras de asfalto, permitiendo optimizar el tiempo y uso de recursos en la revisión de los deterioros de los pavimentos. Además, las propuestas presentadas permiten la automatización de procesos que actualmente deben ser re-evaluados visualmente.

Es importante destacar el gran impacto que genera el tiempo de procesamiento presentado, el cual ha logrado disminuir tiempos de hasta dos semanas con equipos actuales como el LCMS, a un máximo de 24 horas, es decir, se podrían evaluar 14 veces más kilómetros de carretera.

Finalmente, se concluye que en el trabajo presentado no tan solo se proponen nuevos métodos para la resolución de las problemáticas ya mencionadas, sino también, un mejor desempeño, lo que se traduce en mantener las carreteras de Chile y el mundo de manera más eficiente y eficaz.

## 5.2. Trabajo Futuro

El trabajo realizado en esta memoria presenta un gran avance para abordar la problemática de detección y análisis de grietas. Los resultados obtenidos son satisfactorios, pero se encuentran limitados al trabajo específico realizado acá en la memoria. Dada estas circunstancias, se propone el siguiente trabajo futuro:

- Respecto a la etapa de detección, se propone confeccionar una base de datos mucho más extensa, precisa y completa para entrenar y evaluar la red *pix2pix*. Capaz de prescindir lo máximo posible de la utilización de técnica de *Data Augmentation*.
- Para la etapa de clasificación, se propone la realización de una propuesta basada en redes de segmentación semántica, como, *DeepLabV3*.
- Extender e implementar las soluciones presentadas en esta memoria a imágenes de grietas en carretera capturadas por distintos medios, como: tecnologías diferentes al LCMS, fotos capturadas por cámaras digitales o celulares, etc.
- Realizar pre-procesamiento mediante redes de segmentación semántica para filtrar áreas no deseadas, tales como: tapas de alcantarilla, resaltos, delimita-

ciones viales, tachas reflectantes, etc., que pueden perjudicar a la detección pura de grietas.

- Agregar propuesta para la detección de grietas selladas en imágenes *intensity* del LCMS.
- Implementar las propuestas presentadas para la detección de grietas y deterioros en pavimentos de hormigón.

# Bibliografía

- [1] Noble, J. (8 de mayo de 1994). World's Oldest Paved Road Found in Egypt. *The New York Times*, pp. 13.
- [2] Red Caminera de Chile, Ministerio de Obras Públicas. (2017). Carta Caminera.
- [3] Instituto Nacional de Estadísticas, Ministerio de Economía. (2017). Compendio Estadístico.
- [4] Laurent, J., François, J., Lefebvre, D., Savard, Y. (2016). 3D laser road profiling for the automated measurement of road surface conditions and geometry. Pavemetrics Systems inc., Canada.
- [5] Rudin, L., Osher, S., Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4), 259-268.
- [6] Chambolle, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision*, 20(1), 89-97.
- [7] Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., . . . Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer Vision, Graphics and Image Processing*, 39(3), 355-368.
- [8] Serra, J. (1986). Introduction to mathematical morphology. *Computer Vision, Graphics and Image Processing*, 35(3), 283-305.
- [9] Irwin Sobel, Gary Feldman. (2015). An Isotropic 3x3 Image Gradient Operator. Unpublished.
- [10] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6), 679-698.
- [11] Burger, W., Burge, M. (2013). *Principles of Digital Image Processing: Advanced Methods* (2013 ed., Undergraduate Topics in Computer Science). London: Springer London.
- [12] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model

- for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <https://doi.org/10.1007/BF00344251>
- [13] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code. *Neural Comput.* 1, 4, 541–551. <https://doi.org/10.1162/neco.1989.1.4.541>
- [14] Krizhevsky, A., Sutskever, I., Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- [15] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation.
- [16] Long, J., Shelhamer, E., Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *arXiv.org*. <http://search.proquest.com/docview/2081826840/>
- [17] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.
- [18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). Generative Adversarial Networks. *arXiv.org*. <http://search.proquest.com/docview/2084531261/>
- [19] Isola, P., Jun-Yan, Z., Zhou, T., Efros, A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. *arXiv.org*. <http://search.proquest.com/docview/2076424618/>
- [20] Dice, L. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3), 297–302. <https://doi.org/10.2307/1932409>
- [21] Chinchor, N. (1992). MUC-4 Evaluation Metrics. *Proc. of the Fourth Message Understanding Conference*, 22–29. <https://www.aclweb.org/anthology/M92-1002.pdf>
- [22] Stehman, S. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1), 77–89. [https://doi.org/10.1016/S0034-4257\(97\)00083-7](https://doi.org/10.1016/S0034-4257(97)00083-7)
- [23] Correia, Paulo, Oliveira, Henrique. (2010). Automatic Crack Detection On Road Imagery Using Anisotropic Diffusion And Region Linkage. *Zenodo*.
- [24] Quan, Y., Sun, J., Zhang, Y., Zhang, H. (2019). The Method of the Road Surface Crack Detection by the Improved Otsu Threshold. *Tianjin, China, 2019*, pp. 1615-1620.
- [25] Oliveira, H., Lobato Correia, P. (2017). 2017 25th European Signal Processing



Conference (EUSIPCO) (Vol. 2017). EURASIP.

- [26] Hassan, N., Mathavan, S., Kamal, K. (2017). 2017 23rd International Conference on Automation and Computing (ICAC). Chinese Automation and Computing Society in the UK - CACSUK.
- [27] Avila, M., Begot, S., Duculty, F., Nguyen, T. (2014). 2014 IEEE International Conference on Image Processing (ICIP). IEEE.
- [28] Dirección de Vialidad, Ministerio de Obras Públicas. (2018). Manual De Carreteras: Volumen N7, Mantenimiento Vial.
- [29] Serrano C., J., Martínez-Santos, J. (2018). Advances in Computing [electronic resource]: 13th Colombian Conference, CCC 2018, Cartagena, Colombia, September 26–28, 2018, Proceedings (1st ed. 2018.). Springer International Publishing. <https://doi.org/10.1007/978-3-319-98998-3>