



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

**IDENTIFICACIÓN DE DAÑO EN PANELES COMPUESTOS MEDIANTE
ALGORITMO DE APRENDIZAJE PROFUNDO SUPERVISADO CON
ARQUITECTURA W-NET**

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MECÁNICO

RAÚL ALEJANDRO CARREÑO BIGGS

PROFESORA GUÍA:
VIVIANA I. MERUANE NARANJO

MIEMBROS DE LA COMISIÓN:
RAFAEL O. RUIZ GARCÍA
ENRIQUE A. LÓPEZ DROGUETT

ESTE TRABAJO HA SIDO PARCIALMENTE FINANCIADO POR:
PROYECTO FONDECYT 1170535

SANTIAGO DE CHILE
2021

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA
POR: RAÚL ALEJANDRO CARREÑO BIGGS
FECHA: 2021
PROF. GUÍA: VIVIANA I. MERUANE NARANJO

IDENTIFICACIÓN DE DAÑO EN PANELES COMPUESTOS MEDIANTE ALGORITMO DE APRENDIZAJE PROFUNDO SUPERVISADO CON ARQUITECTURA W-NET

Hoy en día, el *Deep Learning* (DL) está presente en gran parte de las industrias, permitiendo que computadoras realicen cálculos complejos en un lapso de tiempo reducido. Tanta es la incorporación del DL en las industrias, que gracias al avance tecnológico en procesamiento gráfico (GPU) y *cloud computing*, ésta rama del *Machine Learning* es un gran foco de investigación actual.

Este trabajo hará uso de ésta herramienta vanguardista en el campo de la detección de daño, más específicamente daño por delaminación en paneles compuestos. Los paneles compuestos son considerados uno de los materiales de ingeniería más utilizados en las industrias de precisión, donde es vital la confiabilidad y no sacrificar las propiedades mecánicas por intentar reducir el peso. El daño por delaminación es el modo de falla más característico de estos paneles, y realizar la detección del daño con precisión es el objetivo que se persigue al implementar el DL en este campo.

El enfoque principal de este trabajo es realizar una primera aproximación a una comparación entre dos tipos de algoritmos de DL supervisados para la identificación de daño en paneles compuestos. Para esto, el algoritmo W-Net es optimizado en modo supervisado (entrenamiento con etiquetas) con dos funciones de costo (IoU Loss y Dice Loss) y luego es implementado a imágenes que representan daño por delaminación, mismas imágenes donde fue implementado el algoritmo supervisado U-Net con anterioridad. La comparación se realizó en aspectos de desempeño de predicción, medido con cuatro métricas de evaluación (IoU Coefficient, Dice Coefficient, Probabilistic Rand Index y Variation of Information), y recursos invertidos en la obtención de los resultados.

Del estudio se concluye que el algoritmo W-Net supervisado requiere una mayor capacidad de cómputo que el algoritmo U-Net al realizar el mismo entrenamiento. En referencia al desempeño de cada algoritmo, los resultados indican que el algoritmo supervisado U-Net es más eficaz al evaluar datasets numéricos, mientras el algoritmo W-Net supervisado obtiene predicciones más precisas al evaluar datasets experimentales.

Dedicado a mi madre Alejandra, a mi pareja Ana y a todos mis amigos que me apoyaron en esta etapa.

Agradecimientos

Antes que todo agradezco a mi madre, quién me educó y me convirtió en el hombre que soy hoy, y es gracias a ella que estoy superando esta etapa de mi vida. A Raksha también.

Dentro de todos mis amigos que estuvieron presentes en mi etapa universitaria, mencionaré los que más incidencia tuvieron en este proceso. Aun así, agradezco a cada uno de mis amigos, porque sin ellos no habría podido afrontar este desafío de la manera en que lo hice.

A Bryan, con quien más compartí en la universidad, en tiempo y diversos aspectos. Considerado el hermano que no tuve y quien más me ayudó a crecer como persona, profesional y amigo.

A Felipe, Jose, Francisco y Guille, mis compañeros de aventuras, acompañándonos desde un completo en gorbea, hasta las celebraciones personales de cada uno. Siempre tendrán en mi un amigo con quien contar.

A Vicente, Valentina, Jose C. y Joaquín, con quienes compartí más mis últimos años de universidad por los ramos que tomamos juntos. Fueron un gran apoyo en lo personal y lo académico.

A Sebastián e Ignacia, por su amistad incondicional y con quienes puedo ser yo mismo sin miedo a ser juzgado de ninguna manera.

A Carol, Nicole, Jorge, Ronald y Harold, familia más cercana que estuvo siempre presente y pendiente por si necesitaba algo.

A Valeria DM, Linda, Fabi, Valentina T, Ian y amigos de futbol/baby en general.

A Ana, con quien he compartido la vida de una forma totalmente diferente este último año y medio. Con quien comparto una confidencialidad que traspasa todo tipo de barreras, desde problemas personales hasta la simple distancia que nos separa físicamente. Sin tí apoyándome éste último periodo de mi proceso universitario no lo hubiera logrado de tan buena manera.

Gracias a la profesora Viviana por su apoyo incondicional en este trabajo y al profesor Rafael por su disposición para apoyarme cuando lo necesité.

A Manuel Elgueta por su interés y apoyo, incondicional y desinteresado, en mi desarrollo personal y profesional. A María Elisa, Héctor, Enrique, Andrés y Fernando Elgueta, por su apoyo en lo familiar y profesional.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	2
1.3. Alcances	2
2. Antecedentes	3
2.1. Paneles compuestos	3
2.2. Daño por delaminación	3
2.3. Algoritmos supervisados y no supervisados	5
2.4. Algoritmo U-Net	5
2.5. Algoritmo W-Net	6
2.6. Métricas de evaluación	9
2.6.1. Intersection-over-Union (IoU)	9
2.6.2. Dice Coefficient (Dice)	10
2.6.3. Probabilistic Rand index o Rand score (PRI)	10
2.6.4. Variation of Information o Mutual Information (VI)	11
3. Metodología	12
3.1. Estudio de metodología U-Net	12
3.2. Arquitectura personalizada e hiperparámetros de entrenamiento para el algoritmo W-Net supervisado	15
3.2.1. Construcción de la arquitectura personalizada	15
3.2.2. Hiperparámetros de entrenamiento	16
3.3. Entrenamiento y obtención de resultados	16
4. Resultados	18
4.1. Modelo entrenado U-Net	18
4.1.1. Dataset numérico	18
4.1.2. Dataset experimental	18
4.2. W-Net supervisado: IoU Loss	20
4.2.1. Optimización de hiperparámetros	20
4.2.2. Dataset numérico	21
4.2.3. Dataset experimental	21
4.3. W-Net supervisado: Dice Loss	23
4.3.1. Optimización de hiperparámetros	23
4.3.2. Dataset numérico	24
4.3.3. Dataset experimental	25
4.4. Resumen	27

4.4.1. Dataset numérico	27
4.4.2. Dataset experimental	30
5. Análisis de resultados	33
6. Conclusiones	36
Bibliografía	38
7. Anexos	39
7.1. Predicciones dataset numérico - U-Net: IoU Loss	39
7.2. Proceso de optimización de hiperparámetros - W-Net (S): IoU Loss	44
7.3. Predicciones dataset numérico - W-Net (S): IoU Loss	47
7.4. Proceso de optimización de hiperparámetros - W-Net (S): Dice Loss	52
7.5. Predicciones dataset numérico - W-Net (S): Dice Loss	55

Índice de Tablas

3.1.	Configuración del modelo para segmentación de imágenes.	13
3.2.	Daño normalizado de los cuatro casos experimentales y el rendimiento de predicción de daño utilizando el algoritmo U-Net y método de thresholding.	14
3.3.	Configuración inicial para pruebas del W-Net supervisado.	17
4.1.	Resultados del dataset numérico de testing con el modelo entrenado U-Net.	19
4.2.	Resultados del dataset experimental con el modelo entrenado U-Net.	20
4.3.	Configuración óptima del modelo W-Net supervisado: IoU Loss.	21
4.4.	Resultados del dataset numérico de testing con el modelo W-Net supervisado: IoU Loss.	22
4.5.	Resultados del dataset experimental con el modelo W-Net supervisado: IoU Loss.	23
4.6.	Configuración óptima del modelo W-Net supervisado: Dice Loss.	24
4.7.	Resultados del dataset numérico de testing con el modelo W-Net supervisado: Dice Loss.	26
4.8.	Resultados del dataset experimental con el modelo W-Net supervisado: Dice Loss.	26
4.9.	Comparación de resultados del dataset numérico - IoU Coefficient.	27
4.10.	Comparación de resultados del dataset numérico - Dice Coefficient.	28
4.11.	Comparación de resultados del dataset numérico - Adjusted Rand score.	28
4.12.	Comparación de resultados del dataset numérico - Adjusted Mutual Info score.	29
4.13.	Comparación de resultados del dataset experimental - Caso 1.	30
4.14.	Comparación de resultados del dataset experimental - Caso 2.	30
4.15.	Comparación de resultados del dataset experimental - Caso 3.	31
4.16.	Comparación de resultados del dataset experimental - Caso 4.	32
7.1.	Resultados de optimización de la técnica de regularización para el algoritmo W-Net supervisado: IoU Loss.	44
7.2.	Resultados de optimización de learning rate para el algoritmo W-Net supervisado: IoU Loss.	45
7.3.	Resultados de optimización de cantidad de canales para el algoritmo W-Net supervisado: IoU Loss.	46
7.4.	Selección del modelo para obtención de resultados con el algoritmo W-Net supervisado: IoU Loss.	46
7.5.	Resultados de optimización de la técnica de regularización para el algoritmo W-Net supervisado: Dice Loss.	52
7.6.	Resultados de optimización de learning rate para el algoritmo W-Net supervisado: Dice Loss.	53
7.7.	Resultados de optimización de cantidad de canales para el algoritmo W-Net supervisado: Dice Loss.	54
7.8.	Selección del modelo para obtención de resultados con el algoritmo W-Net supervisado: Dice Loss.	54

Índice de Ilustraciones

2.1.	Panel compuesto tipo sándwich con núcleo “honeycomb” de aluminio.	4
2.2.	Esquema que muestra daño por delaminación circular en un panel compuesto tipo sándwich.	4
2.3.	Ejemplo de segmentación semántica.	5
2.4.	Esquema original de la arquitectura “U-Net” [4].	6
2.5.	Arquitectura “W-Net” creada por Xide y Brian. [5]	7
2.6.	Esquema general de la figura original y la predicción hecha por el algoritmo, ambas utilizadas por el método IoU	10
2.7.	Esquema gráfico que explica la formulación de la métrica “Dice Coefficient”.	10
3.1.	Comparación de resultados obtenidos por el algoritmo U-Net y método de thresholding.	13
3.2.	Daño experimental identificado utilizando el algoritmo U-Net con la configuración propuesta. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17. Los círculos rojos indican el daño real.	14
3.3.	Esquema de la arquitectura personalizada propuesta para estudio del algoritmo W-Net en modo supervisado.	16
4.1.	Predicciones realizadas por modelo entrenado U-Net. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.	19
4.2.	Daño experimental identificado utilizando el modelo entrenado disponible del U-Net. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17. Los círculos rojos indican el daño real.	19
4.3.	Imágenes representativas de predicción con la técnica <i>Batch Normalization + Dropout (20 %)</i> en proceso de optimización del W-Net supervisado: IoU Loss.	20
4.4.	Función de costo a lo largo de la fase de entrenamiento del modelo óptimo seleccionado.	21
4.5.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.	22
4.6.	Daño experimental identificado utilizando el modelo W-Net supervisado: IoU Loss. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17.	22
4.7.	Comportamiento de la función de costo “Dice Loss” en dos entrenamientos con la técnica <i>Batch Normalization + Dropout (20 %)</i> , esto en el proceso de optimización del W-Net supervisado: Dice Loss.	23

4.8.	Predicciones realizadas por los entrenamientos 1 y 5 con la técnica <i>Batch Normalization + Dropout (20%)</i> , esto en el proceso de optimización del W-Net supervisado: Dice Loss.	24
4.9.	Función de costo a lo largo de la fase de entrenamiento del modelo óptimo seleccionado.	25
4.10.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.	25
4.11.	Daño experimental identificado utilizando el modelo W-Net supervisado: Dice Loss. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17.	26
4.12.	Comparación de resultados del dataset numérico - IoU Coefficient.	27
4.13.	Comparación de resultados del dataset numérico - Dice Coefficient.	28
4.14.	Comparación de resultados del dataset numérico - Adjusted Rand score.	29
4.15.	Comparación de resultados del dataset numérico - Adjusted Mutual Info score.	29
4.16.	Comparación de resultados del dataset experimental - Caso 1.	30
4.17.	Comparación de resultados del dataset experimental - Caso 2.	31
4.18.	Comparación de resultados del dataset experimental - Caso 3.	31
4.19.	Comparación de resultados del dataset experimental - Caso 4.	32
7.1.	Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 2.	39
7.2.	Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 3.	40
7.3.	Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 4.	41
7.4.	Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 5.	42
7.5.	Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 6.	43
7.6.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 2.	47
7.7.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 3.	48
7.8.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 4.	49
7.9.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 5.	50
7.10.	Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 6.	51
7.11.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 2.	55
7.12.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 3.	56
7.13.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 4.	57

7.14.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 5.	58
7.15.	Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 6.	59

Capítulo 1

Introducción

El presente trabajo de título busca continuar la exploración y utilización del aprendizaje profundo (Deep Learning) en el campo de la detección de daño presente en componentes estructurales. La tecnología evoluciona día a día y permite que los computadores mejoren su eficiencia en términos de procesamiento de datos, eficiencia que hace 30 años nadie hubiera creído que sería posible conseguir. En particular, los avances en unidades de procesamiento gráfico (GPU) para los computadores, y en *cloud computing* (almacenamiento de datos en la nube) han impulsado el desarrollo del *Deep Learning*, logrando introducirlo cada día más en varias industrias para automatizar sus procesos.

Por su parte, los paneles compuestos representan un importante foco de investigación para el *Deep Learning* dadas las numerosas aplicaciones de estos paneles en varias industrias de precisión, las que requieren de una alta confiabilidad y estructuras ligeras que no sacrifiquen sus propiedades mecánicas por buscar un menor peso (e.g. aeroespacial, aeronáutica, automotriz, entre otras). Es así como la detección de daño toma un papel importante, donde un modo de falla característico de estos componentes es la falla por delaminación, la cual consiste en una separación local entre el núcleo (*core*) y las capas externas (*layers*) del panel compuesto, generando pérdida de cohesión y propiedades mecánicas en las zonas afectadas.

Ahora, los algoritmos de *Deep Learning* pueden ser clasificados en dos categorías: supervisados y no supervisados, donde la principal diferencia entre ambas clasificaciones es la existencia de etiquetas (*labels*) en el subconjunto de datos de entrenamiento para los algoritmos supervisados, y que los algoritmos no supervisados evitan.

Este trabajo tiene un enfoque mixto entre algoritmos supervisados y no supervisados, ya que se implementa el algoritmo “W-Net”, categorizado como algoritmo no supervisado de tipo *auto-encoder*, en su modo supervisado, es decir, se toma la arquitectura del algoritmo pero se realiza el entrenamiento con etiquetas. Para la construcción y estructuración del algoritmo se utilizó el software gratuito Python. El algoritmo es entrenado con dos funciones de costo, “IoU Loss y Dice Loss, e implementado a imágenes en escala de grises que contienen la información de daño por delaminación en paneles compuestos. Luego, el desempeño de los resultados obtenidos por el algoritmo fue medido con los métodos de evaluación *Intersection-over-Union (IoU)*, *Dice Coefficient* (segmentación de imágenes), *Probabilistic Rand Index* y *Variation of Information* (Clustering), y comparados con el desempeño logrado por el U-Net en el mismo set de imágenes para determinar ventajas y desventajas de ambos algoritmos.

1.1. Motivación

En la época actual, el *Deep Learning* está generando avances tecnológicos en todos los sectores de la industria, y es una herramienta vanguardista cuando hablamos de detección de daño en estructuras. Es por esto que, aplicar y evaluar el desempeño del algoritmo no supervisado W-Net, en modo supervisado, en este campo de la industria, persigue la idea de profundizar el conocimiento del *Deep Learning*.

Por otro lado, existe un estudio hecho con anterioridad, en el cual se implementa el algoritmo supervisado U-Net en el mismo grupo de imágenes numéricas y experimentales. Esto incentiva la comparación científica entre ambas clases de algoritmos: supervisados y no supervisados, con el fin de analizar las ventajas y desventajas de cada uno en esta aplicación. Para tal propósito, este trabajo sirve solo como primera aproximación.

1.2. Objetivos

Objetivo general: Identificar posibles zonas con daño de tipo “delaminación” en paneles compuestos, esto por medio de un algoritmo de aprendizaje profundo supervisado, cuya red posee una estructura similar a la del algoritmo W-Net original.

Objetivos específicos:

- Diseñar y estructurar el algoritmo personalizado con la red W-Net utilizando el programa Python.
- Entrenar el algoritmo con las funciones de costo “IoU Loss” y “Dice Loss”
- Evaluar ambos entrenamientos con las métricas “IoU coefficient”, “Dice coefficient”, “Probabilistic Rand Index” y “Variation of Information”.
- Comparar resultados con los obtenidos por el algoritmo U-Net en el estudio anterior.

1.3. Alcances

Este trabajo tiene su comienzo en la recuperación de las imágenes, ya utilizadas en un estudio anterior, las cuales contienen la información de los índices de daño. La metodología de obtención de los dataset de imágenes puede ser consultada en [1].

El trabajo finalizará con el análisis de los resultados obtenidos y con la comparación, referente a efectividad y metodología (recursos computacionales invertidos en obtener los resultados), entre el algoritmo W-Net (en modo supervisado) y el algoritmo supervisado U-Net.

Capítulo 2

Antecedentes

En este capítulo se presenta una breve discusión bibliográfica de los antecedentes básicos, con el fin de contextualizar al lector sobre las bases en las que se sustenta este proyecto. Luego, se profundizará bibliográficamente en los antecedentes que son fundamentales para llevar a cabo el desarrollo de este trabajo.

2.1. Paneles compuestos

Cuando se habla de la estructura de un sistema mecánico (e.g. un avión) en ingeniería, se hace referencia a los materiales que soportan y protegen a los componentes principales, y que además dan forma y/o restringen el movimiento del sistema. Hoy en día, los principales componentes estructurales en las industrias de precisión son los paneles compuestos, ya que son considerablemente más livianos a otros materiales de ingeniería y no sacrifican las propiedades mecánicas para cumplir con los objetivos estructurales que se les exige.

Existe una gran variedad de paneles compuestos hoy en día, entre los cuales se encuentran los paneles compuestos de tipo “Sándwich”, que son utilizados en todo el mundo en un sinnúmero de aplicaciones. Los paneles compuestos tipo sándwich están formados por dos capas externas (*layers*) cubriendo un panel interno llamado núcleo (*core*), el cuál puede ser homogéneo o no homogéneo. Las capas externas se unen al núcleo mediante algún adhesivo.

Para este estudio se utilizará, como objeto experimental, un panel compuesto tipo sándwich de dimensiones 0.35 m x 0.25 m x 0.021 m, con un núcleo no homogéneo tipo “Panel de abeja” (*Honeycomb*) de aluminio, como el que se muestra en la **Figura 2.1**. Además, las capas externas serán adheridas al núcleo con resina epoxy. Los detalles del montaje experimental y las características del panel compuesto pueden ser consultados en la referencia [1].

2.2. Daño por delaminación

Una de las causas más comunes y características de falla en paneles compuestos tipo sándwich es la separación local entre una capa externa y el núcleo, como se esquematiza en la **Figura 2.2**. Este modo de falla puede producir una reducción de la capacidad de carga y la integridad estructural del panel. El inicio del proceso de delaminación puede presentarse en



Figura 2.1: Panel compuesto tipo sándwich con núcleo “honeycomb” de aluminio.

cualquier etapa de la vida del panel, desde su proceso de manufactura hasta el fin de su vida útil.

Hoy en día, con los avances científicos en el campo de la identificación de daño se puede prescindir de un modelo sano (sin daño) con el que comparar los resultados obtenidos con el panel experimental [2], facilitando la implementación de las técnicas de *Deep Learning*, tanto en procesos de mantenimiento en la industria, como en estudios con fines de desarrollo científico.

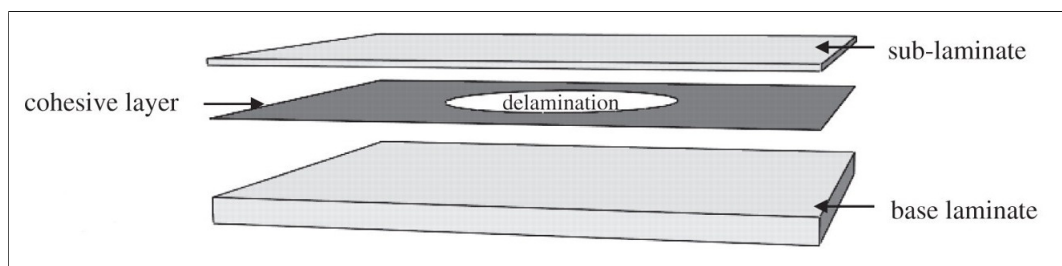


Figura 2.2: Esquema que muestra daño por delaminación circular en un panel compuesto tipo sándwich.

La obtención de los índices de daño por delaminación en los paneles compuestos experimentales de este trabajo siguió la metodología que se resume a continuación:

1. Obtención de los modos de vibración experimentales mediante un sistema high-speed 3D DIC [2], [3].
2. Estimación de la curvatura de los modos de vibración utilizando un proceso de regresión Gaussiana (*Gaussian Process*). Los valores de los parámetros de escala (s_x^2 y s_y^2) y la varianza del ruido (σ_n^2) se extraen de [2], ya que el caso de aplicación es el mismo.
3. Obtener los índices de daño aplicando el método “*Gapped Smoothing*” como se describe en [1], [2] y [3].

Dado que la obtención de los índices de daño escapa al alcance de este trabajo, no se profundizará más en esta sección. Las imágenes resultantes que contienen la información del daño del panel serán directamente utilizadas en el algoritmo W-Net supervisado.

2.3. Algoritmos supervisados y no supervisados

Una de las aplicaciones más frecuentes del *Deep Learning* hoy en día es la clasificación y segmentación semántica de imágenes. La clasificación de imágenes tiene como objetivo determinar a cual categoría, dentro de las creadas por el algoritmo, corresponde el contenido de una o varias imágenes, por ejemplo, determinar si en la imagen hay un perro o un gato. Por otro lado, la segmentación semántica consiste en agrupar los píxeles de las imágenes en distintos conjuntos para diferenciarlos (sin saber a que corresponde cada grupo de píxeles), como en el ejemplo de la **Figura 2.3**, donde se agrupan los píxeles que corresponden al paisaje y los píxeles que no corresponden a ese conjunto, esto sin categorizar los conjuntos como “paisaje” y “persona”, sino que simplemente como dos conjuntos diferentes de píxeles. Dentro de las aplicaciones más relevantes de la segmentación semántica de imágenes se encuentra la conducción autónoma de vehículos, la generación de imágenes médicas y el campo de la inspección industrial.



Figura 2.3: Ejemplo de segmentación semántica.

Como se mencionó anteriormente, los algoritmos de *Deep Learning* se pueden clasificar, a grandes rasgos, en dos categorías: supervisados y no supervisados. Mientras los algoritmos supervisados utilizan etiquetas con las “respuestas” de las predicciones para guiar el entrenamiento, los algoritmos no supervisados se basan en la agrupación de características para clasificar o segmentar una imagen. Por un lado, el procedimiento para la obtención de las etiquetas (de casos experimentales) que utilizan los algoritmos supervisados suele requerir de mucho trabajo, pero por el otro lado, los algoritmos no supervisados requieren una mayor cantidad de cálculos para llevar a cabo el entrenamiento sin etiquetas, siendo un factor muy importante la potencia (hardware) del computador que realizará este entrenamiento.

2.4. Algoritmo U-Net

Este algoritmo supervisado fue publicado el año 2015 por Olaf, Philipp y Thomas [4], quienes buscaban una arquitectura y método de entrenamiento eficientes que pudieran realizar segmentación semántica con unas pocas imágenes. El algoritmo fue postulado al *ISBI challenge 2012* para segmentación semántica en un dataset con imágenes de microscopio electrónico y, usando la misma arquitectura entrenada en imágenes de microscopios de luz transmitida, consiguieron ganar el *ISBI cell tracking challenge 2015*.

La arquitectura del U-Net se muestra en la **Figura 2.4**, que cuenta con 23 capas de convolución distribidas en una etapa de contracción y una de expansión. La etapa de contracción consiste en la repetida aplicación de 2 convoluciones 3x3, cada una seguida por una

capa ReLU, y luego una capa de Max-Pooling 2x2 con “stride” igual a 2. En cada paso de contracción se duplica el número de canales de características. Por otro lado, cada paso de la etapa de expansión consiste en una expansión del mapa de características, seguida por una convolución 2x2 (“up-convolution”), donde se divide a la mitad el número de canales, una concatenación con el mapa de características correspondiente de la etapa de contracción, y dos convoluciones 3x3, cada una seguida por una capa ReLU. Además, cuenta con una capa final de convolución 1x1 para segmentar en la cantidad de clases deseada.

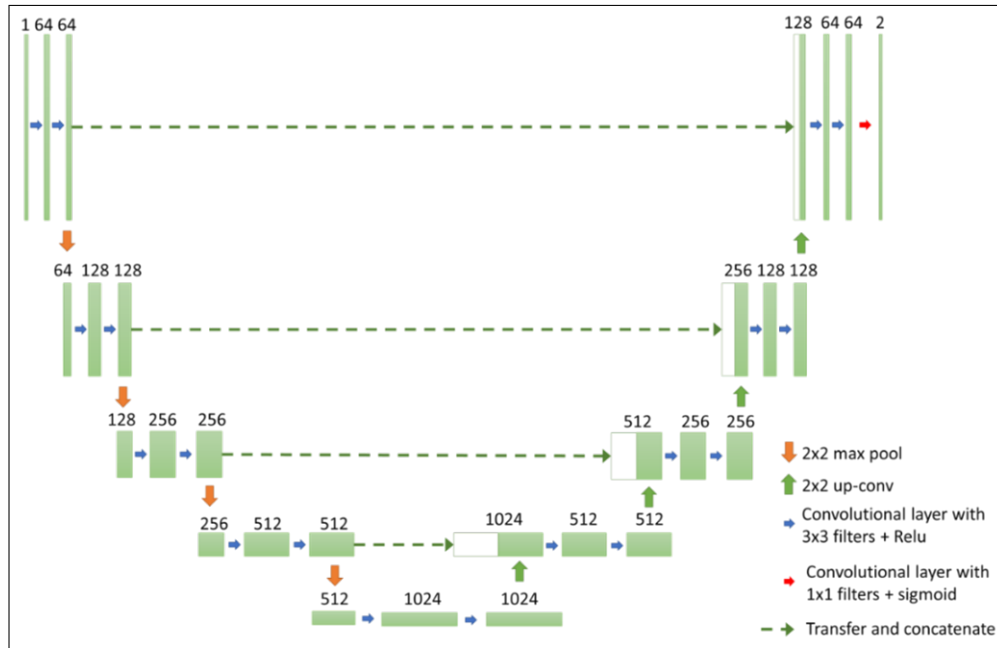


Figura 2.4: Esquema original de la arquitectura “U-Net” [4].

2.5. Algoritmo W-Net

Xide y Brian et al.[5] se basaron en la arquitectura “U-Net”, utilizada en segmentación semántica de imágenes biomédicas para diseñar, en el año 2017, el algoritmo “W-Net” (cuya arquitectura se muestra en la **Figura 2.5**). La motivación fue segmentar imágenes de manera no supervisada para evitar la recolección de una gran cantidad de imágenes etiquetadas, las que son requeridas para el entrenamiento en el caso de algoritmos supervisados y que, por lo demás, conlleva un gran trabajo obtenerlas.

Para ello, crearon esta estructura que está compuesta por dos etapas, cada una con arquitectura similar a la del U-Net, obteniendo un algoritmo no supervisado catalogado como tipo *auto-encoder*. La primera etapa (denominada “U_{Enc}”) se encarga de “codificar” la imagen, capturando las características de la misma y creando mapas de predicción densa. Luego, la segunda etapa (denominada “U_{Dec}”) puede “decodificar” la imagen, es decir, reconstruirla a partir de la base de predicción densa.

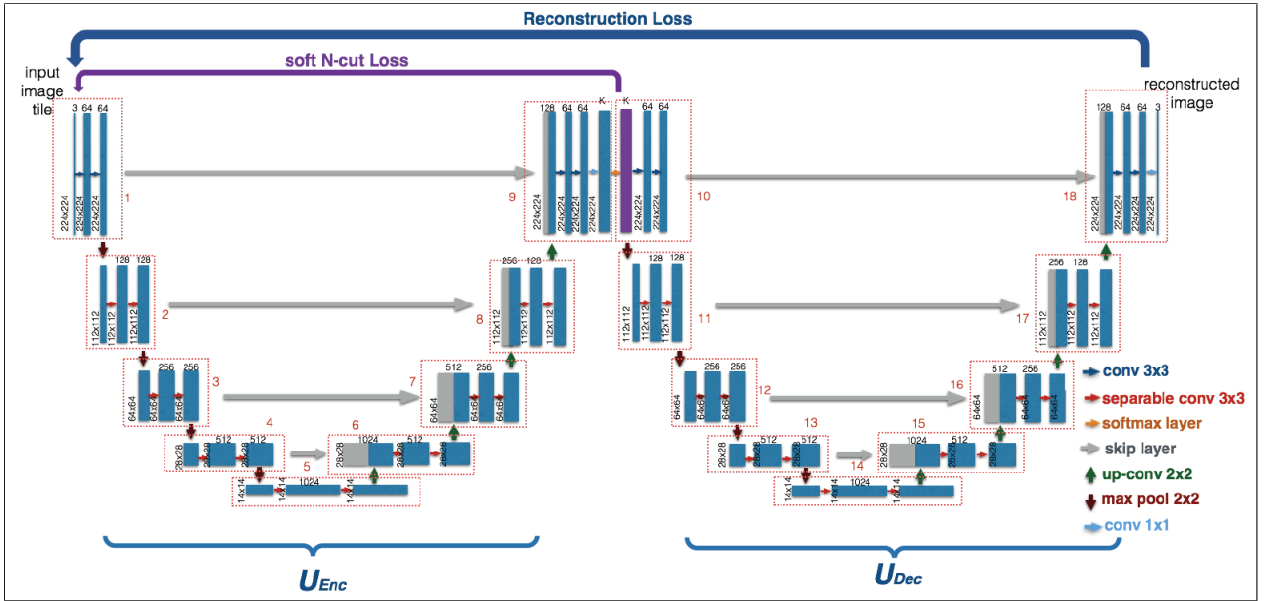


Figura 2.5: Arquitectura “W-Net” creada por Xide y Brian. [5]

En particular, la arquitectura diseñada por Xide y Brian cuenta con estas dos etapas (“ U_{Enc} ” y “ U_{Dec} ”), cada una dividida en 9 módulos (18 en total), los que se encuentran representados por rectángulos rojos en la **Figura 2.5**. Todos los módulos están formados por dos capas de convolución 3x3 (específicamente “Depthwise Separable Convolution”, exceptuando los módulos 1, 9, 10 y 18 con convolución normal), cada una seguida por una capa ReLU y una capa de Batch Normalization. Los módulos están conectados por capas de Max-Pooling 2x2 o capas de convolución transpuesta (“Transpose Convolution”) 2x2 dependiendo si corresponde a una operación de contracción (primera mitad de la etapa) o expansión (segunda mitad de la etapa) de la imagen, reduciendo a la mitad o duplicando las dimensiones de la imagen respectivamente. También, se duplica el número de canales de características en el tramo de contracción, mientras que en el tramo de expansión se reduce a la mitad. Para los tramos de expansión, los módulos utilizan como entrada la unificación o “concatenación” del resultado de la convolución transpuesta con la salida del correspondiente módulo de contracción, esto con el fin de recuperar información que se pierde durante el tramo de contracción. Por último, al final de cada etapa se realiza una convolución 1x1 (seguida por una capa Softmax en el caso de la primera etapa) para traducir la información recolectada por los canales de características en el número de clases de segmentación deseado (para la primera etapa), y el número de canales de color de la imagen reconstruida (para la segunda etapa).

En cuanto al entrenamiento, el algoritmo W-Net cuenta con dos funciones de costo (*loss function*). La primera es la función “Soft Normalized Cut Loss” (**Ecuaciones 2.1 a 2.3**), versión diferenciable de la función “Normalized Cut Loss”, que calcula la asociación y disociación entre grupos de píxeles mediante la siguiente formulación matemática.

$$J_{soft-Ncut}(V, K) = \sum_{k=1}^K \frac{cut(A_k, V - A_k)}{assoc(A_k, V)} \quad (2.1)$$

$$J_{soft-Ncut}(V, K) = K - \sum_{k=1}^K \frac{assoc(A_k, A_k)}{assoc(A_k, V)} \quad (2.2)$$

$$J_{soft-Ncut}(V, K) = K - \sum_{k=1}^K \frac{\sum_{u \in V} p(u = A_k) \cdot \sum_{v \in V} \omega(u, v) p(v = A_k)}{\sum_{u \in V} p(u = A_k) \cdot \sum_{t \in V} \omega(u, t)} \quad (2.3)$$

Donde $p(u = A_k)$ mide la probabilidad de que el pixel u pertenezca a la clase A_k y $\omega(u, v)$ representa la “similaridad” entre los pixeles u y v . Por otra parte, las funciones de corte ($cut(A, B)$) y asociación ($assoc(A, B)$) están definidas, para A y B conjuntos disjuntos, como se muestra en las **Ecuaciones 2.4 y 2.5**.

$$cut(A, B) = \sum_{u \in A, v \in B} \omega(u, v) \quad (2.4)$$

$$assoc(A, V) = \sum_{u \in A, v \in V} \omega(u, v) \quad (2.5)$$

Al entrenar la etapa “ U_{Enc} ” para minimizar el valor de la función “*Soft N-Cut Loss*”, simultáneamente se puede minimizar la disociación normalizada y maximizar la asociación normalizada entre las clases. De esta manera se puede obtener la mejor segmentación de la imagen para el número de clases deseado.

Ahora, la segunda función de costo es llamada “función de reconstrucción” y se muestra en la **Ecuación 2.6**. Ésta función tiene por objetivo comparar la imagen reconstruida con la original para que la segmentación contenga la mayor información posible de la imagen original.

$$J_{reconstr} = \| X - U_{Dec}(U_{Enc}(X; W_{Enc}); W_{Dec}) \|_2^2 \quad (2.6)$$

Donde W_{Enc} denota los parámetros del encoder, W_{Dec} denota los parámetros del decoder, y X es la imagen original.

Xide y Brian proponen el entrenamiento del W-Net utilizando ambas funciones de manera semi-simultanea, con el fin de balancear el trade-off entre la precisión de la reconstrucción y la consistencia en la segmentación. A continuación, el **Algoritmo 1** muestra el procedimiento de entrenamiento propuesto mencionado.

Ahora, es importante recordar que para este trabajo se utilizará la arquitectura del W-Net y se evaluará como algoritmo supervisado, dejando de lado las funciones de costo y el método de entrenamiento propuesto en el trabajo original del W-Net. La construcción de este “W-Net supervisado” se detalla en el capítulo 3.

Algoritmo 1: Entrenamiento de un “minibatch” del W-Net.

1. **procedure** W-Net ($X; U_{\text{Enc}}, U_{\text{Dec}}$)
2. **for** número de iteraciones de entrenamiento **do**:
3. Tomar un “minibatch” de nuevas imagenes originales x
4. Actualizar U_{Enc} minimizando $J_{\text{soft-Ncut}}$
5. **Solo actualizar** U_{Enc}
6. Actualizar toda la red W-Net minimizando J_{reconstr}
7. **Actualizar ambos** U_{Enc} y U_{Dec}
8. **return** U_{Enc}

2.6. Métricas de evaluación

Existe un gran número de métodos de evaluación para algoritmos que trabajan la segmentación semántica de imágenes. Por ejemplo, en el trabajo de Xide y Brian se utilizan tres: *Segmentation covering*, *probabilistic Rand index* y *variation of information*. Para este trabajo en particular, se utilizarán 4 métodos: “Intersection over Union” (IoU), “Dice coefficient” (Dice), “Probabilistic Rand Index” (PRI) y “Variation of Information” (VI). La selección de estos métodos se debe en parte a que son simples e intuitivos, pero principalmente debido a su utilización en los trabajos directamente relacionados a este [1], [5].

2.6.1. Intersection-over-Union (IoU)

El método IoU toma la figura original que se busca predecir y la predicción entregada por el algoritmo, y con ellas calcula, como se muestra en la **Ecuación 2.7**, la intersección sobre la unión.

$$IoU = \frac{TP}{FP + TP + FN} = \frac{\text{Área de intersección}}{\text{Área de unión}} \quad (2.7)$$

Donde TP, FP y FN corresponden al número de pixeles “verdaderos positivos”, “falsos positivos” y “falsos negativos” respectivamente. La **Figura 2.6** apoya el entendimiento de como opera este método desde una perspectiva más gráfica.

Entonces, el método entrega un coeficiente normalizado para evaluar el desempeño del algoritmo. Así, si la predicción es perfecta, el método entregará como resultado el valor 1, y en el caso de que la predicción sea totalmente errónea (ningún pixel de la predicción pertenece al grupo de pixeles originales) el método arrojará un coeficiente igual a 0.

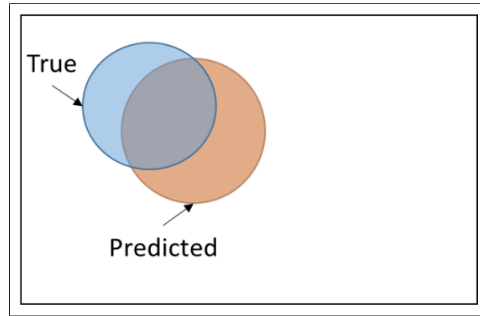


Figura 2.6: Esquema general de la figura original y la predicción hecha por el algoritmo, ambas utilizadas por el método IoU

2.6.2. Dice Coefficient (Dice)

Este método es bastante similar al IoU, pero su formulación es diferente. De igual manera, se toma la figura original y la predicción que realiza el algoritmo. Con ellas calcula lo siguiente.

$$Dice = \frac{2 \cdot TP}{FP + 2 \cdot TP + FN} = \frac{2 \cdot \text{Área de intersección}}{\text{Área ambas figuras}} \quad (2.8)$$

Este cálculo se vuelve más intuitivo al ver el esquema explicativo de la **Figura 2.7** que se muestra a continuación.

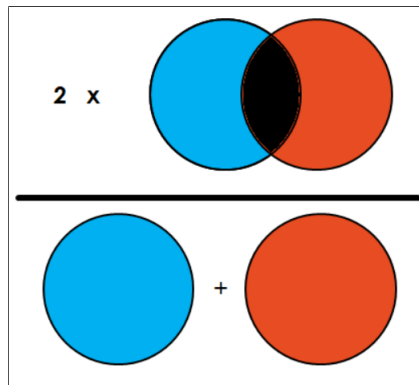


Figura 2.7: Esquema gráfico que explica la formulación de la métrica “Dice Coefficient”.

Al igual que el IoU, el método Dice Coefficient entrega un valor entre 0 y 1, donde el 0 representa una predicción totalmente errónea y el valor 1 se alcanza cuando la predicción es perfecta.

2.6.3. Probabilistic Rand index o Rand score (PRI)

Originalmente, esta métrica se introdujo al aprendizaje profundo para evaluación de “clustering” (agrupamiento) general. El método calcula una similitud medida entre 2 agrupamientos, considerando todos los pares de píxeles y contando los pares que están asignados en el mismo o diferente grupo, en la predicción del algoritmo y la etiqueta real.

En este trabajo se utiliza específicamente la métrica “*adjusted rand score*” (ARI) de la

librería “scikit learn” [6]. La formulación matemática que sigue esta métrica es la siguiente.

$$ARI = \frac{(RI - E(RI))}{(\max(RI) - E(RI))} \quad (2.9)$$

Donde $E(RI)$ es la esperanza o valor esperado del Rand score y RI es el Rand score calculado como indica la **Ecuación 2.10**.

$$RI = \frac{\text{N}^\circ \text{ of agreeing pairs}}{\text{N}^\circ \text{ of pairs}} \quad (2.10)$$

De este modo, el “adjusted Rand score” asegura obtener un valor entre 0 y 1. Mientras el valor 0 representa un etiquetado al azar, independiente del número de clusters o pixeles, y el valor 1 representa un agrupamiento idéntico.

2.6.4. Variation of Information o Mutual Information (VI)

Al igual que el *Rand index*, el “*variation of information score*” también fue creado con el propósito de comparación entre clusterings. En particular, este método mide la distancia entre dos segmentaciones en términos de su entropía condicional promedio.

Se hará uso de la métrica específica “*adjusted mutual info score*” (AMI) de “scikit learn” [6], la cual ajusta el hecho de que la “*Mutual Information*” (MI) es generalmente mayor para un clustering con gran número de clusters, sin importar si efectivamente hay más información compartida o no. Para dos clusterings U y V , el AMI está dado por la siguiente ecuación.

$$AMI(U, V) = \frac{[MI(U, V) - E(MI(U, V))]}{[\frac{H(U)+H(V)}{2} - E(MI(U, V))]} \quad (2.11)$$

Donde $E(MI(U, V))$ es la esperanza o valor esperado del “*Mutual info score*” y $H(U)$, $H(V)$ representan las entropías de los clusterings U y V respectivamente. La fórmula utilizada por scikit learn para calcular el MI se muestra en la **Ecuación 2.12**.

$$MI(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \quad (2.12)$$

Aquí, los valores que entrega la métrica también tienen como máximo valor 1. Para una comparación entre dos clusterings idénticos el resultado será 1, mientras que para etiquetados al azar se espera obtener valores cercanos a 0, pudiendo ser incluso negativos.

Capítulo 3

Metodología

La metodología de este estudio se divide en tres secciones. En primer lugar, se realiza un estudio de la metodología utilizada en el trabajo del U-Net [1]. Luego, en la segunda sección, se construye una arquitectura personalizada del W-Net, teniendo en consideración el estudio ya realizado, el cuál orienta en la definición de los hiperparámetros que se busca optimizar. Finalmente, se detalla el procedimiento para el entrenamiento y la obtención de resultados del algoritmo W-Net supervisado.

3.1. Estudio de metodología U-Net

Con el afán de realizar una comparación fidedigna, se decidió estudiar la metodología implementada en el trabajo del U-Net antes de personalizar la arquitectura del W-Net. Aquí, construyen una arquitectura base similar a la propuesta por el paper original del U-Net [4], y que se muestra en la **Figura 2.4**. Luego, se propuso indagar algunas alternativas en cuanto a: técnicas de regularización (para prevenir el *overfitting*, además del *EarlyStopping*), *learning rate* del optimizador, cantidad de convoluciones realizadas por bloque y cantidad de filtros (o canales de características), todo esto con el fin de investigar que configuración consigue mejores resultados.

Para realizar la búsqueda de la mejor configuración se utilizó un dataset numérico con 2800 imágenes para entrenamiento, 175 imágenes para validación y 525 imágenes con fines de testeo, todas redimensionadas de 51x71 píxeles a 96x128 píxeles. Se decidió utilizar el optimizador **Adam**, en un principio con un *learning rate* de 0.00001. Además, se utilizó la función de pérdida *IoU Loss*, que equivale a $1 - \text{IoU}$, y se configuró un *EarlyStopping* con paciencia de 50 épocas. Dada la variabilidad existente en este tipo de pruebas, se decidió realizar 5 entrenamientos para cada caso.

El primer paso fue definir que estrategia de regularización entregaba mejores resultados. Se evaluaron 3 casos: Sin regularización, *Batch Normalization* y *Dropout* de 20%, donde los resultados con *Batch Normalization* superaron a los de las demás estrategias de regularización.

Luego, como análisis de sensibilidad, se estudió cual valor de *learning rate* se adecuaba mejor al entrenamiento con un límite máximo de 1000 épocas. Aquí fueron evaluados 4 casos: 10^{-6} , 10^{-5} , 10^{-4} y 10^{-3} . El *learning rate* que obtuvo mejores resultados fue el de valor 10^{-4} .

Después, se realizó el estudio entre la cantidad de convoluciones por bloque, siendo opciones una, dos y tres convoluciones por bloque, resultando en que la opción original con dos convoluciones por bloque consiguió los mejores resultados.

Finalmente, se estudió la cantidad de canales que debían tener las convoluciones de cada bloque. Las opciones eran las siguientes: 16-64-128-256-512, 32-64-128-256-512 y 64-128-256-512-1024. En este aspecto, la última configuración obtuvo los mejores resultados.

Con esto se obtuvo la mejor configuración para realizar el entrenamiento y conseguir los mejores resultados, la cual se presenta a continuación en la **Tabla 3.1**.

Tabla 3.1: Configuración del modelo para segmentación de imágenes.

Optimizador	Adam
Learning rate	0.0001
Estrategia de regularización	Batch Normalization
Función de costo	IoU Loss
Cantidad de canales	64-128-256-512-1024
Número de convoluciones por bloque	2

Una vez que el algoritmo fue optimizado con el dataset numérico, se obtuvieron los resultados de éste dataset y se separaron las imágenes con sus resultados, agrupándolos según su rango de daño normalizado. Los resultados se muestran en la **Figura 3.1**.

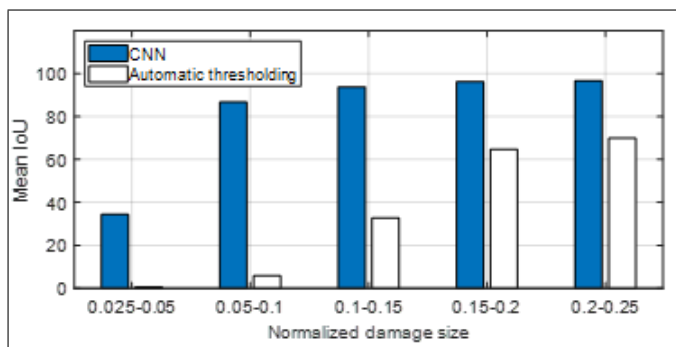
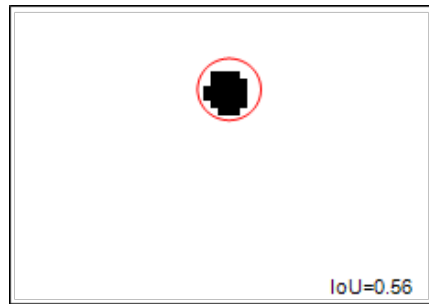


Figura 3.1: Comparación de resultados obtenidos por el algoritmo U-Net y método de thresholding.

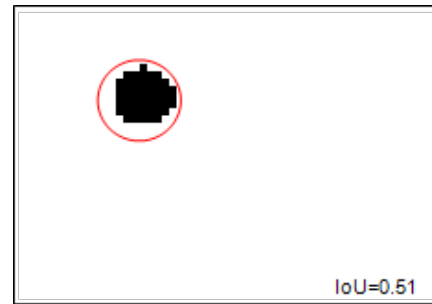
Después, se validó la generalidad de los resultados implementando la configuración óptima en un set de imágenes experimental. De aquí se obtuvieron los índices de daño que se muestran en la **Tabla 3.2**, y las predicciones visibles en la **Figura 3.2**.

Tabla 3.2: Daño normalizado de los cuatro casos experimentales y el rendimiento de predicción de daño utilizando el algoritmo U-Net y método de thresholding.

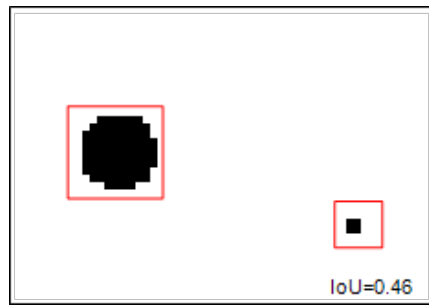
Caso	Daño normalizado			IoU	
	Daño normalizado 1	Daño normalizado 2	Forma	CNN	Automatic thresholding
1	0.09	-	Circular	0.56	0.47
2	0.12	-	Circular	0.51	0.46
3	0.14	0.07	Cuadrada	0.46	0.46
4	0.11	0.17	Circular	0.60	0.65



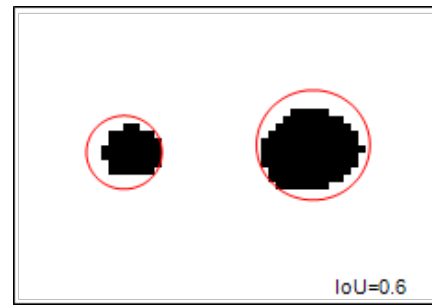
a Caso 1



b Caso 2



c Caso 3



d Caso 4

Figura 3.2: Daño experimental identificado utilizando el algoritmo U-Net con la configuración propuesta. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17. Los círculos rojos indican el daño real.

3.2. Arquitectura personalizada e hiperparámetros de entrenamiento para el algoritmo W-Net supervisado

Para la realización de este trabajo, se debe crear una arquitectura personalizada que se adecúe al caso de estudio y seleccionar cuales serán los hiperparámetros de entrenamiento para optimizar el modelo.

3.2.1. Construcción de la arquitectura personalizada

Se consideró tomar como base la arquitectura del W-Net, presente en la **Figura 2.5**, y se modificó su característico sistema de entrenamiento tipo autoencoder, con dos funciones de costo que trabajan semi-simultáneamente (ambas se computan una vez por cada batch, pero no al mismo tiempo), por una sola función de costo al final de la estructura, esto para convertirlo en un algoritmo supervisado. Para este trabajo, se realizó el entrenamiento del algoritmo dos veces, optimizando con dos funciones de costo diferentes: “IoU Loss” y “Dice Loss”.

Al igual que la arquitectura original, la propuesta de W-Net supervisado se divide en 18 módulos, cada uno compuesto por dos convoluciones “Depthwise Separable” 3x3 exceptuando, de igual manera, los bloques 1, 9, 10 y 18 que efectúan convoluciones normales, ambas seguidas de una capa ReLU. Esta arquitectura también mantiene como conexión entre módulos capas de Max-Pooling 2x2 y “Transpose Convolution” 2x2, para tramos de contracción y expansión respectivamente, lo que produce el mismo fenómeno de reducir a la mitad o duplicar las dimensiones de la imagen. También fue necesario replicar el proceso de concatenación en los tramos de expansión debido a la utilización de la convolución transpuesta en estos módulos.

Ahora, dentro de las modificaciones realizadas a la arquitectura, se encuentra el cambio de dimensiones de la imagen de entrada, fijándolas en 64x80 píxeles. La selección de estos valores se debe al proceso de reducción, donde las dimensiones de la imagen se deben reducir a la mitad 4 veces, siendo necesario que las dimensiones iniciales sean divisibles por 16. Desde luego, se podrían haber utilizado imágenes de mayor tamaño para aumentar la precisión del estudio, pero fue necesario considerar la capacidad de cómputo del ordenador, y haciendo pruebas preliminares, se determinó que éstas eran las mayores dimensiones posibles para realizar el estudio.

Otra modificación menor, aunque no menos importante, es que el número de canales K presente en los bloques 9 y 10 fue fijado en $K = 2$, ya que los resultados pierden sentido para otros valores de K .

Las últimas modificaciones corresponden al número de canales y la cantidad de convoluciones por módulo. La razón que motiva éstas modificaciones será explicada con más detalle al momento de mencionar cuales serán los hiperparámetros a optimizar.

De esta manera, la arquitectura propuesta para el estudio del algoritmo W-Net supervisado queda definida como se muestra gráficamente en la **Figura 3.3**.

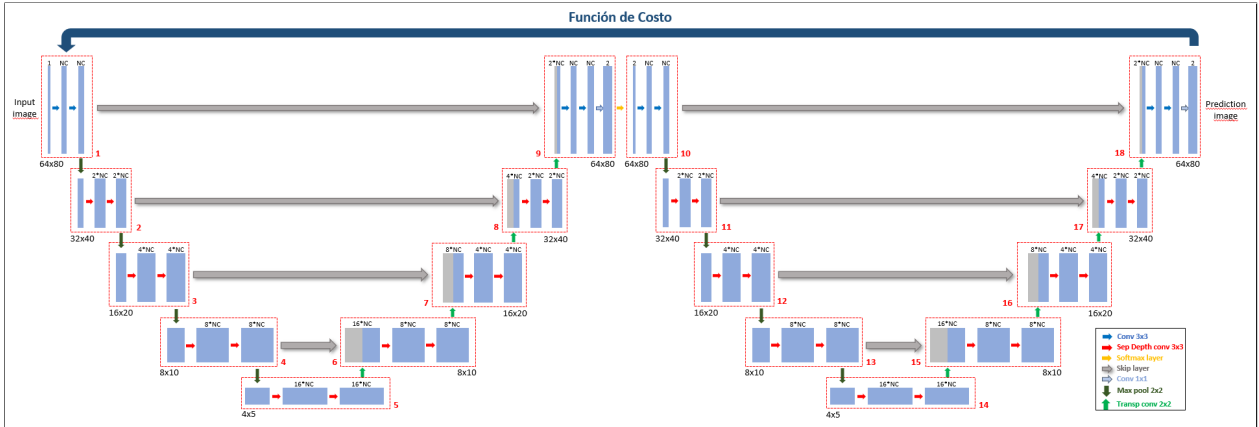


Figura 3.3: Esquema de la arquitectura personalizada propuesta para estudio del algoritmo W-Net en modo supervisado.

3.2.2. Hiperparámetros de entrenamiento

Como se dijo en un principio, se buscó realizar un procedimiento similar al efectuado en el trabajo del U-Net. Para esto, se consideró hacer las mismas pruebas sobre las estrategias de regularización, *learning rate* y cantidad de canales. Se añadió la prueba “*Batch Normalization + Dropout* de 20%” a las ya existentes en la categoría de estrategias de regularización, esto debido a que el trabajo original del W-Net propone esta estrategia y es un caso de estudio interesante. Además, por el apremio del tiempo se decidió no realizar las pruebas de diferentes números de convoluciones por bloque, asumiendo que lo propuesto en la arquitectura original del W-Net y los mejores resultados obtenidos del trabajo del U-Net no entregaban la misma conclusión (de utilizar dos convoluciones por bloque) por mera casualidad.

3.3. Entrenamiento y obtención de resultados

Antes de obtener los resultados de este trabajo, y ya que se dispone del modelo entrenado del U-Net, se extrajeron los resultados de este modelo por cuenta propia, ya que se trabajará con más métricas de evaluación de las que se usaron en dicho trabajo.

Ahora, el dataset de imágenes numéricas y experimentales utilizado en este trabajo es el mismo que se utilizó en el trabajo del U-Net, tanto para entrenamiento como para obtención de resultados. Las imágenes del dataset numérico fueron redimensionadas de 51x71 píxeles a 64x80 píxeles. Se utilizó el mismo optimizador (Adam) y el mismo *learning rate* inicial ($LR = 0.00001$). Como se mencionó anteriormente, el procedimiento de entrenamiento y obtención de resultados es realizado para dos funciones de costo: “IoU Loss” y “Dice Loss”.

Para comenzar la optimización de hiperparámetros se define la configuración inicial que se visualiza en la **Tabla 3.3**.

Tabla 3.3: Configuración inicial para pruebas del W-Net supervisado.

Parámetro	Configuración
Tamaño de la imagen	64x80x1
Optimizador	Adam
Learning rate	0.0001
Loss Function	IoU/Dice Loss
Estrategia de regularización	Sin regularización
Cantidad de filtros	32-64-128-256-512
Paciencia del EarlyStopping	50 épocas
Batch size	25

Debido a la variabilidad existente en cada entrenamiento, se realizaron 5 entrenamientos por cada prueba. La optimización de los hiperparámetros se realizó en el orden específico que se describe a continuación.

1. Estrategia de regularización

- a) Sin regularización
- b) *Batch Normalization*
- c) *Dropout* (20 %)
- d) *Batch Normalization* y *Dropout* (20 %)

2. Learning rate

- a) LR = 0.000001
- b) LR = 0.00001
- c) LR = 0.0001
- d) LR = 0.001

3. Cantidad de canales

- a) 16-64-128-256-512
- b) 32-64-128-256-512
- c) 64-128-256-512-1024

El entrenamiento se realizó con el dataset numérico, en particular, los sets de entrenamiento y validación. Una vez determinada la configuración óptima, se implementan los cinco modelos entrenados en el set de testing y se escoge, para la obtención de resultados, el que obtenga el resultado más cercano al promedio de estos modelos.

Finalmente, se extraen los resultados del set numérico de testing y del dataset experimental, éste último para validar la generalidad de los resultados. En cada dataset se obtienen las evaluaciones de las métricas para calificar el desempeño y las predicciones gráficas para calificar la precisión.

Capítulo 4

Resultados

Antes de exponer los resultados del trabajo, es necesario dejar en evidencia que no se muestran imágenes de la totalidad de ellos, debido a la gran cantidad de imágenes y entrenamientos realizados. Los resultados ilustrados aquí son los resultados finales y los que presenten algún fenómeno interesante que comentar en el capítulo de análisis.

En cuanto al formato de presentación, este capítulo está dividido en cuatro secciones, una destinada a los resultados obtenidos por el modelo entrenado disponible del U-Net, dos donde se muestran los resultados del modelo W-Net supervisado entrenado con las dos funciones de costo, y el último destinado a mostrar un resumen de los resultados expuestos en las tres secciones anteriores, éste último con fines comparativos. A su vez, cada sección está dividida en tres subsecciones para presentar los resultados del proceso de optimización, los resultados del dataset numérico y los resultados del dataset experimental por separado, exceptuando la sección de resultados del U-Net, el cual no fue optimizado en este trabajo y estará dividido solo en dos subsecciones.

4.1. Modelo entrenado U-Net

4.1.1. Dataset numérico

Dado que el modelo entrenado del U-Net está disponible, se obtuvieron directamente los resultados de las métricas para el dataset numérico de testing.

A continuación, La **Figura 4.1** ilustra algunas predicciones de referencia junto con sus respectivas imágenes originales y etiquetas, las que fueron escogidas al azar pero intentando que se representen los resultados de todo el dataset. Más predicciones son mostradas en el capítulo de **Anexos 7.1**. Por su parte, la **Tabla 4.1** muestra los resultados de las métricas obtenidos por este modelo entrenado.

4.1.2. Dataset experimental

También se obtuvieron directamente los resultados de los cuatro casos del dataset experimental, los que se visualizan gráficamente en la **Figura 4.2** y numéricamente en la **Tabla 4.2**.

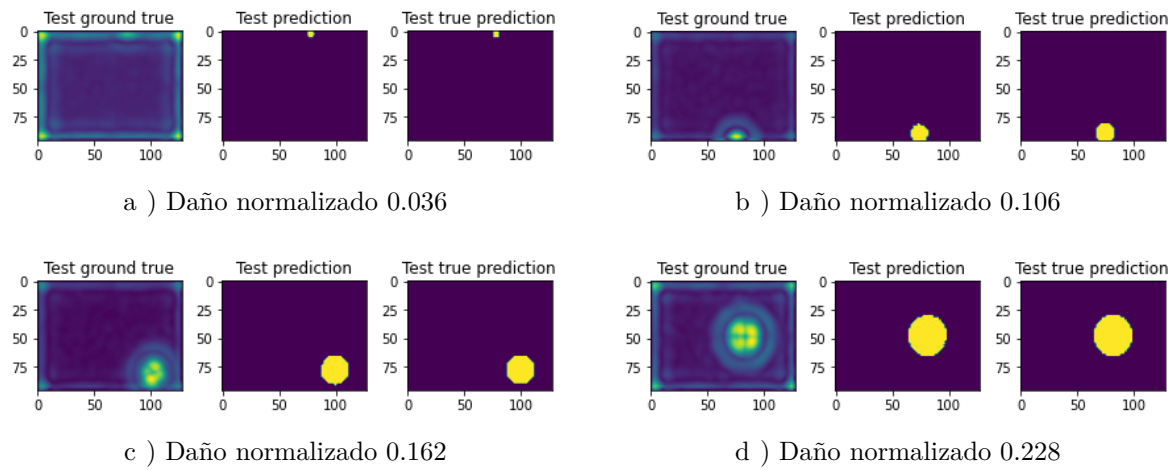


Figura 4.1: Predicciones realizadas por modelo entrenado U-Net. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.

Tabla 4.1: Resultados del dataset numérico de testing con el modelo entrenado U-Net.

Intervalo	Índice de daño	U-Net: IoU Loss			
		IoU	Dice	PRI	VI
Intervalo 1	0 - 0.025	0,0009	0,0013	0,9656	0,9617
Intervalo 2	0.025 - 0.05	0,5233	0,665	0,4171	0,3672
Intervalo 3	0.05 - 0.1	0,858	0,9234	0,9223	0,8611
Intervalo 4	0.1 - 0.15	0,9494	0,974	0,9657	0,925
Intervalo 5	0.15 - 0.2	0,9617	0,9805	0,9775	0,9421
Intervalo 6	0.2 - 0.25	0,967	0,9832	0,9788	0,944
Dataset Completo	0 - 0.25	0,9456	0,972	0,9053	0,8683

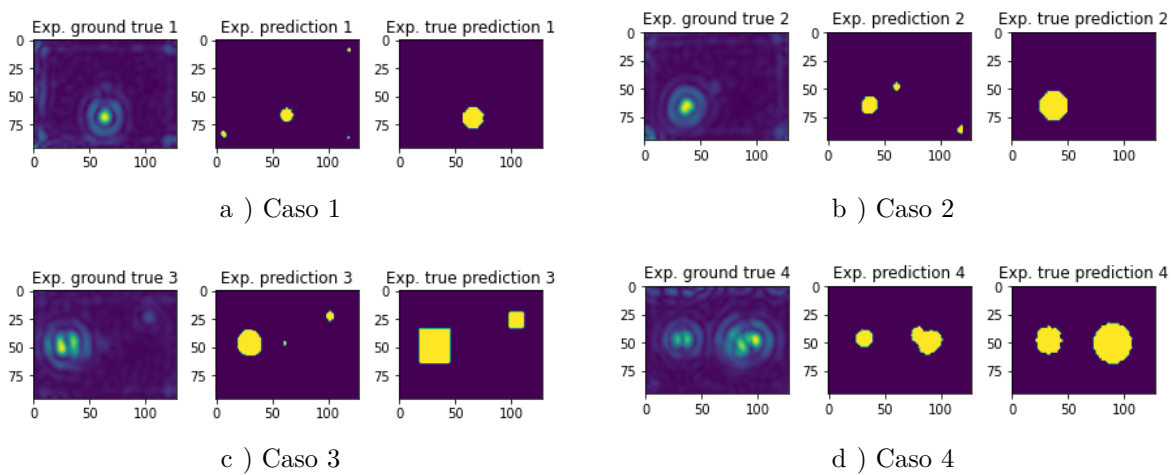


Figura 4.2: Daño experimental identificado utilizando el modelo entrenado disponible del U-Net. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17. Los círculos rojos indican el daño real.

Tabla 4.2: Resultados del dataset experimental con el modelo entrenado U-Net.

Casos	Daño normalizado	Figura	U-Net: IoU Loss			
			IoU	Dice	PRI	VI
Caso 1	0.09	Circular	0,3331	0,4997	0,4809	0,3204
Caso 2	0.12	Circular	0,3093	0,4725	0,4412	0,2785
Caso 3	0.14 y 0.07	Cuadrada	0,4189	0,5904	0,5354	0,4148
Caso 4	0.11 y 0.17	Circular	0,4471	0,6180	0,5446	0,4308

4.2. W-Net supervisado: IoU Loss

4.2.1. Optimización de hiperparámetros

El primer paso de la optimización consistió en definir cual de las técnicas de regularización, entre las propuestas, conseguía los mejores resultados. Para esta función de costo, la opción **Sin regularización** obtuvo los mejores resultados, alcanzando un coeficiente IoU promedio de 0,916 (**Anexos, Tabla 7.1**). Ahora, algo interesante que surgió en este paso de la optimización fueron los resultados de la opción “Batch Normalization + Dropout (20%)”, que en cuanto a resultados numéricos no es relevante, pero las predicciones tienen un punto interesante de discusión que se abordará en el respectivo capítulo de análisis.

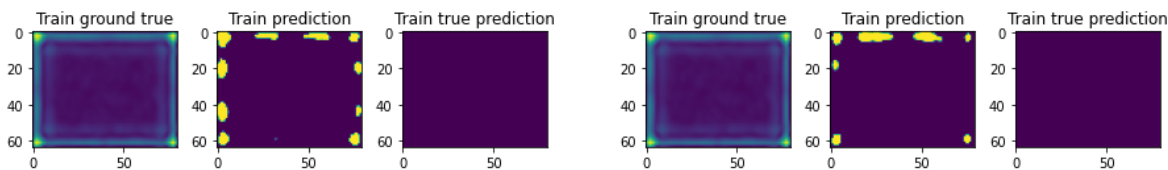


Figura 4.3: Imágenes representativas de predicción con la técnica *Batch Normalization + Dropout (20%)* en proceso de optimización del W-Net supervisado: IoU Loss.

Luego, se definió el learning rate adecuado para que el entrenamiento fuera óptimo, y entre las opciones evaluadas, el que entregó mejores resultados fue el **LR = 0,0001**, sobrepasando a las demás opciones con un coeficiente promedio IoU de 0,924 (**Anexos, Tabla 7.2**). Un hecho a tener en cuenta es que los entrenamientos con las opciones **LR = 0,00001** y **LR = 0,000001** no convergieron antes de llegar a la época límite.

Finalmente, se optimizó el hiperparámetro “Cantidad de canales” donde los resultados indicaron que la segunda opción, con **32-64-128-256-512** canales, entrega los mejores resultados, llegando al coeficiente promedio IoU de 0,927 (**Anexos, Tabla 7.3**). Es importante destacar en este apartado que la tercera opción, con **64-128-256-512-1024** canales, no logró finalizar el entrenamiento debido a la falta de memoria RAM (Hardware).

Así, la configuración del modelo W-Net supervisado, optimizado con la función de costo *IoU Loss*, y que será evaluado en el dataset numérico de testing y el experimental, quedó definida como se muestra en la **Tabla 4.3**.

Tabla 4.3: Configuración óptima del modelo W-Net supervisado: IoU Loss.

Optimizador	Adam
Learning rate	0.0001
Estrategia de regularización	Sin regularización
Función de costo	IoU Loss
Cantidad de canales	32-64-128-256-512
Número de convoluciones por bloque	2

4.2.2. Dataset numérico

Como primer resultado, se evaluó el dataset numérico de testing con los 5 modelos entrenados por la configuración óptima. El promedio de coeficientes IoU que alcanzó esta configuración fue de 0,9204 (**Anexos, Tabla 7.4**) y, en base a esto, se escogió el modelo con el tercer entrenamiento (con $\text{IoU} = 0,9202$) para la obtención de los demás resultados, tanto con el dataset numérico como con el dataset experimental. A continuación se muestra el comportamiento de la función de costo a lo largo de la fase de entrenamiento.

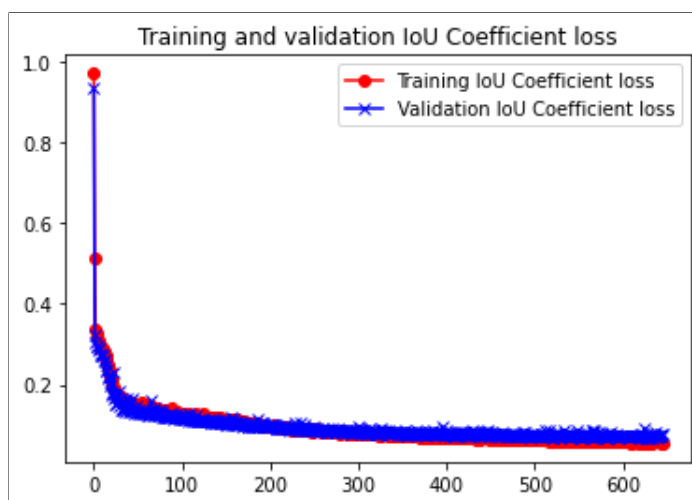


Figura 4.4: Función de costo a lo largo de la fase de entrenamiento del modelo óptimo seleccionado.

Luego, con el modelo seleccionado, se extraen los resultados numéricos que se presentan en la **Tabla 4.4**. También se muestran predicciones de referencia (corresponden a las predicciones de las mismas imágenes mostradas en la sección de resultados del U-Net) en la **Figura 4.5** para representar de manera gráfica los resultados obtenidos. Más predicciones pueden ser consultadas en el capítulo **Anexos 7.3**.

4.2.3. Dataset experimental

Al evaluar los cuatro casos del dataset experimental, se obtuvieron las predicciones ilustradas, junto con sus imágenes originales y sus etiquetas, en la **Figura 4.6**. Los valores obtenidos por cada métrica se muestran en la **Tabla 4.5**.

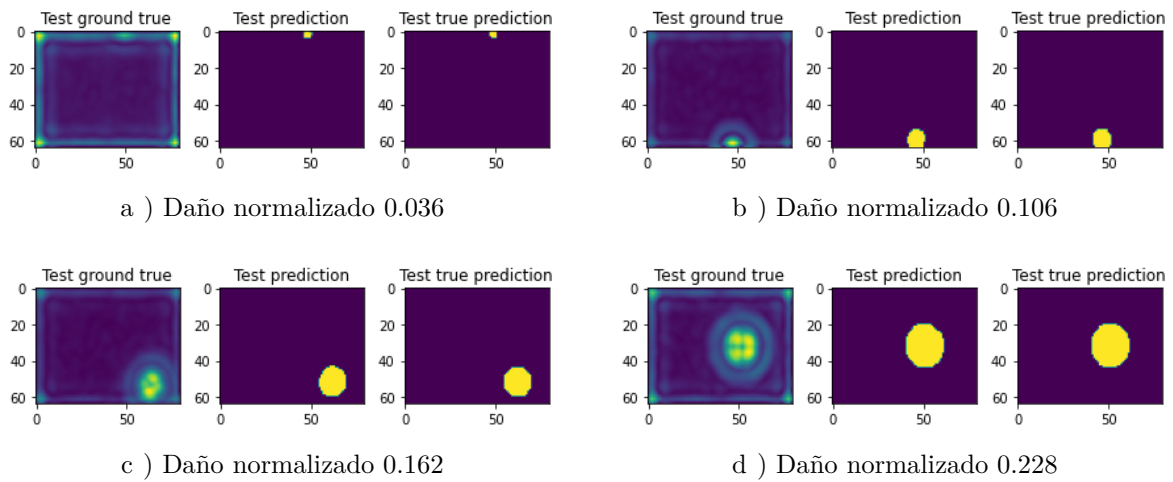


Figura 4.5: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.

Tabla 4.4: Resultados del dataset numérico de testing con el modelo W-Net supervisado: IoU Loss.

Intervalo	Índice de daño	W-Net (S): IoU Loss			
		IoU	Dice	PRI	VI
Intervalo 1	0 - 0.025	0,0008	0,0008	0,8518	0,81
Intervalo 2	0.025 - 0.05	0,1086	0,1932	0,8737	0,8294
Intervalo 3	0.05 - 0.1	0,7976	0,8863	0,8569	0,8168
Intervalo 4	0.1 - 0.15	0,9249	0,9609	0,8832	0,8426
Intervalo 5	0.15 - 0.2	0,9452	0,9718	0,8354	0,7939
Intervalo 6	0.2 - 0.25	0,9516	0,9752	0,7898	0,7484
Dataset Completo	0 - 0.25	0,9202	0,9584	0,8495	0,8082

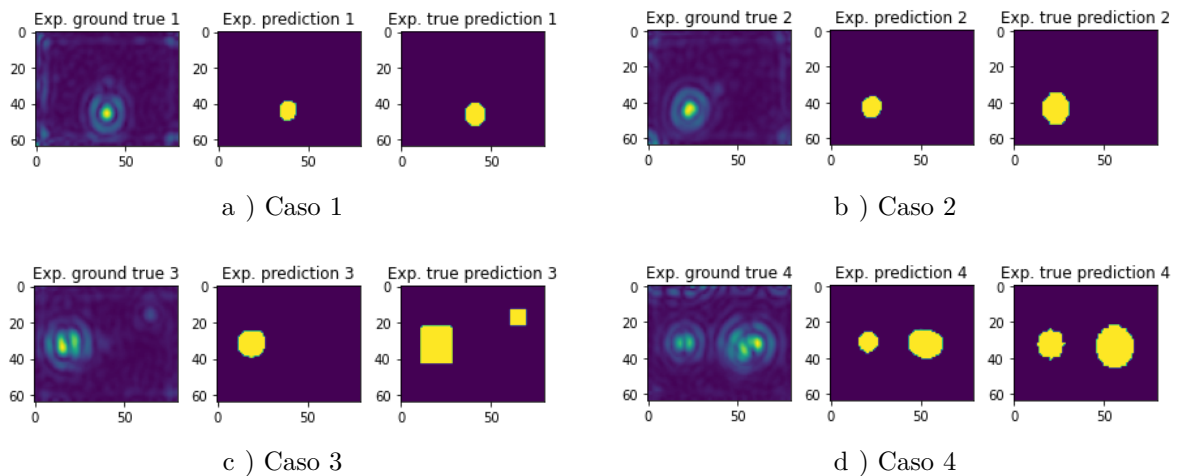


Figura 4.6: Daño experimental identificado utilizando el modelo W-Net supervisado: IoU Loss. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17.

Tabla 4.5: Resultados del dataset experimental con el modelo W-Net supervisado: IoU Loss.

Casos	Daño normalizado	Figura	W-Net (S): IoU Loss			
			IoU	Dice	PRI	VI
Caso 1	0.09	Circular	0,4942	0,6615	0,6475	0,4709
Caso 2	0.12	Circular	0,4953	0,6625	0,6378	0,5225
Caso 3	0.14 y 0.07	Cuadrada	0,4146	0,5861	0,5310	0,4266
Caso 4	0.11 y 0.17	Circular	0,5347	0,6968	0,6232	0,4649

4.3. W-Net supervisado: Dice Loss

4.3.1. Optimización de hiperparámetros

Siguiendo la misma metodología que con la función de costo IoU Loss, lo primero es definir cual estrategia de regularización obtiene los mejores resultados. Para el caso de la *Dice Loss*, la mejor opción es “Sin regularización”, la cual obtuvo un promedio de coeficiente Dice de 0,964 (**Anexos, Tabla 7.5**). Al igual que para el entrenamiento realizado por la función IoU Loss, la opción “Batch Normalization + Dropout (20%)” obtuvo resultados interesantes de analizar. En las siguientes figuras se ilustrarán el comportamiento de la función de costo a lo largo de los entrenamientos y predicciones realizadas por distintos entrenamientos para representar el comportamiento.

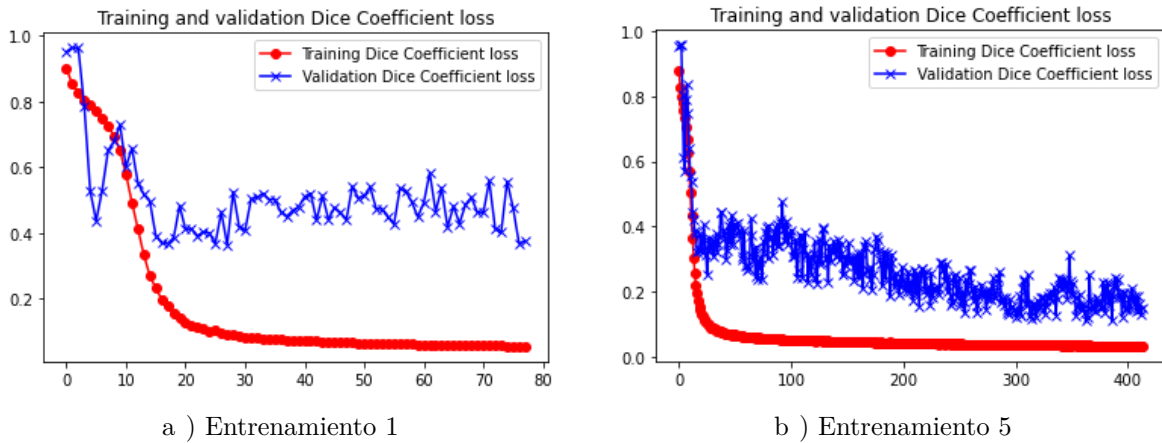


Figura 4.7: Comportamiento de la función de costo “Dice Loss” en dos entrenamientos con la técnica *Batch Normalization + Dropout (20%)*, esto en el proceso de optimización del W-Net supervisado: Dice Loss.

Continuando con la optimización, el siguiente hiperparámetro en la lista es el learning rate. Dentro de las opciones de optimización, la que obtuvo los mejores resultados fue el learning rate con valor $\mathbf{LR} = \mathbf{0,0001}$, con un valor promedio de coeficiente Dice igual a 0,959 (**Anexos, Tabla 7.6**). De igual manera que ocurrió con la optimización del W-Net supervisado mediante la función IoU Loss, los entrenamientos con $\mathbf{LR} = \mathbf{0,00001}$ y $\mathbf{LR} = \mathbf{0,000001}$ no alcanzaron a converger antes de la época 1000.

Por último, se escogió la cantidad de canales óptima. Nuevamente la opción con mejores

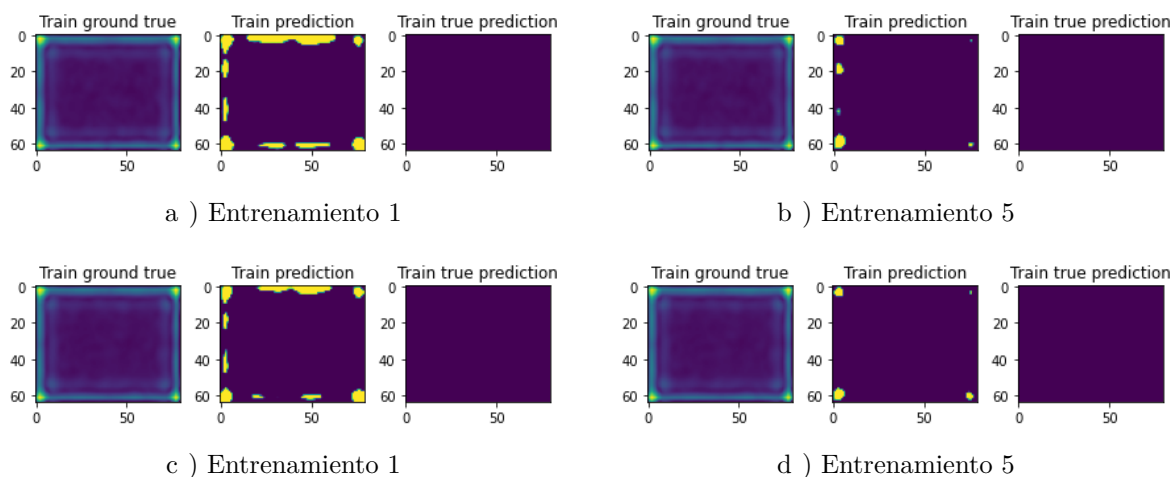


Figura 4.8: Predicciones realizadas por los entrenamientos 1 y 5 con la técnica *Batch Normalization + Dropout (20 %)*, esto en el proceso de optimización del W-Net supervisado: Dice Loss.

resultados fue **32-64-128-256-512** canales, alcanzando el valor promedio de coeficiente Dice de 0,963 (**Anexos, Tabla 7.7**).

De esta manera, la configuración óptima del modelo W-Net supervisado: Dice Loss, es la que se presenta en la **Tabla 4.6**, y con ella se evaluarán los dataset numérico de testing y dataset experimental para obtener sus respectivos resultados.

Tabla 4.6: Configuración óptima del modelo W-Net supervisado: Dice Loss.

Optimizador	Adam
Learning rate	0.0001
Estrategia de regularización	Sin regularización
Función de costo	Dice Loss
Cantidad de canales	32-64-128-256-512
Número de convoluciones por bloque	2

4.3.2. Dataset numérico

Siguiendo la metodología, antes de sacar resultados fue necesario seleccionar el modelo entrenado a utilizar para este fin. Se evaluó el dataset numérico de testing con los 5 entrenamientos de la configuración óptima y se obtuvo un promedio de coeficiente Dice de 0,9589 (**Anexos, Tabla 7.8**). El quinto entrenamiento fue escogido para la obtención de los demás resultados, ya que su coeficiente Dice fue prácticamente igual al promedio (Dice = 0,9588). La **Figura 4.9** ilustra gráficamente el comportamiento de la función de costo del modelo seleccionado a lo largo de la fase de entrenamiento.

Ya que el modelo fue seleccionado, se obtuvieron los resultados del dataset numérico de testing por intervalos. En la **Figura 4.10** se muestran algunas de las predicciones hechas por el modelo (correspondientes a las mismas imágenes expuestas en secciones anteriores).

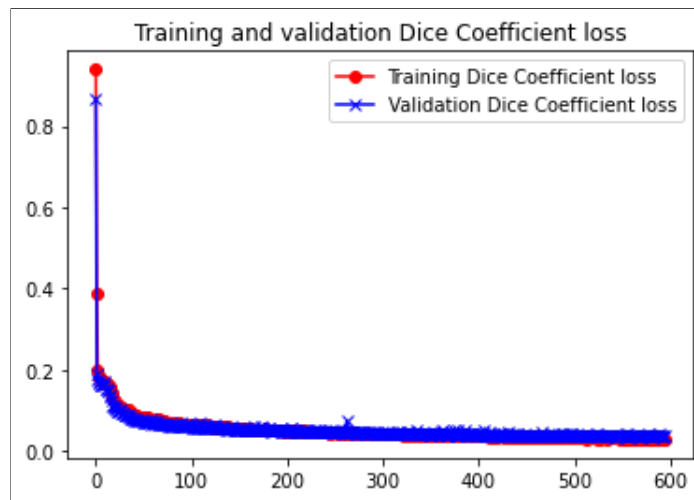


Figura 4.9: Función de costo a lo largo de la fase de entrenamiento del modelo óptimo seleccionado.

Además, la **Tabla 4.7** muestra los valores numéricos de métricas obtenidos. Más predicciones se visualizan en el capítulo **Anexos 7.5**.

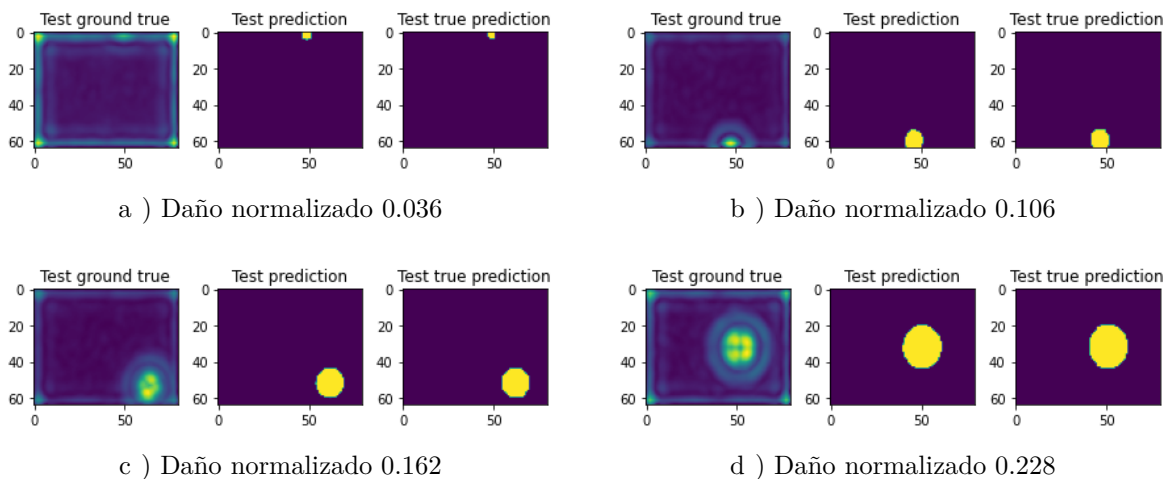


Figura 4.10: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. En cada imagen, la gráfica de la izquierda representa la imagen original introducida al algoritmo, la gráfica intermedia representa la predicción hecha por el algoritmo, y la gráfica de la derecha representa el daño real por delaminación.

4.3.3. Dataset experimental

Los cuatro casos, al ser evaluados con el modelo óptimo seleccionado, presentan las predicciones ilustradas en la **Figura 4.11**. Junto con esto, los valores de cada métrica que obtuvo el modelo se presentan a continuación en la **Tabla 4.8**.

Tabla 4.7: Resultados del dataset numérico de testing con el modelo W-Net supervisado: Dice Loss.

Intervalo	Índice de daño	W-Net (S): Dice Loss			
		IoU	Dice	PRI	VI
Intervalo 1	0 - 0.025	0,0107	0,0107	0,8395	0,7992
Intervalo 2	0.025 - 0.05	0,1236	0,2196	0,8787	0,8337
Intervalo 3	0.05 - 0.1	0,7911	0,8813	0,853	0,8109
Intervalo 4	0.1 - 0.15	0,9126	0,954	0,8862	0,8456
Intervalo 5	0.15 - 0.2	0,9486	0,9736	0,8254	0,7878
Intervalo 6	0.2 - 0.25	0,9518	0,9753	0,7859	0,7486
Dataset Completo	0 - 0.25	0,9195	0,958	0,8452	0,8049

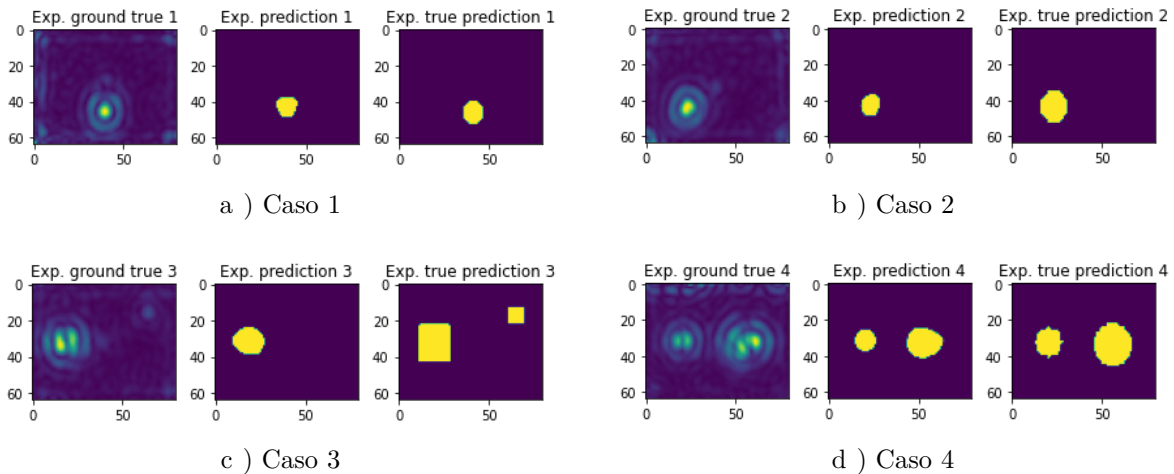


Figura 4.11: Daño experimental identificado utilizando el modelo W-Net supervisado: Dice Loss. Tamaño de daño normalizado: a) 0.09, b) 0.12, c) 0.14 y 0.07 y d) 0.11 y 0.17.

Tabla 4.8: Resultados del dataset experimental con el modelo W-Net supervisado: Dice Loss.

Casos	Daño normalizado	Figura	W-Net (S): Dice Loss			
			IoU	Dice	PRI	VI
Caso 1	0.09	Circular	0,4486	0,6193	0,6002	0,4063
Caso 2	0.12	Circular	0,4680	0,6376	0,6079	0,4946
Caso 3	0.14 y 0.07	Cuadrada	0,4482	0,6189	0,5621	0,4362
Caso 4	0.11 y 0.17	Circular	0,5848	0,7380	0,6690	0,5139

4.4. Resumen

Como se mencionó anteriormente, en esta sección se presentan los resultados de todo el capítulo, pero agrupados con un enfoque más comparativo. Solo se agruparán los resultados finales que, de no ser agrupados, perjudiquen la comprensión de su análisis. Bajo el mismo argumento, las imágenes no fueron agrupadas, para evitar generar confusión.

4.4.1. Dataset numérico

Los resultados del dataset numérico de testing se agruparon según métrica, ya que no tiene sentido comparar valores obtenidos por métricas diferentes. Para cada métrica se generó un gráfico que ilustra los resultados numéricos obtenidos por cada modelo. En éstos gráficos, no se agregan los resultados del primer intervalo (imágenes con índice de daño normalizado menor a 0,025), ya que las predicciones no detectan el daño por delaminación y no tiene sentido hacer una comparación entre resultados incoherentes.

Tabla 4.9: Comparación de resultados del dataset numérico - IoU Coefficient.

Intervalo	Índice de daño	IoU Coefficient		
		U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
Intervalo 1	0 - 0.025	0,0858	0,00075	0,0107
Intervalo 2	0.025 - 0.05	0,5233	0,1086	0,1236
Intervalo 3	0.05 - 0.1	0,858	0,7976	0,7911
Intervalo 4	0.1 - 0.15	0,9494	0,9249	0,9126
Intervalo 5	0.15 - 0.2	0,9617	0,9452	0,9486
Intervalo 6	0.2 - 0.25	0,967	0,9516	0,9518
Dataset completo	0 - 0.25	0,9456	0,9202	0,9195

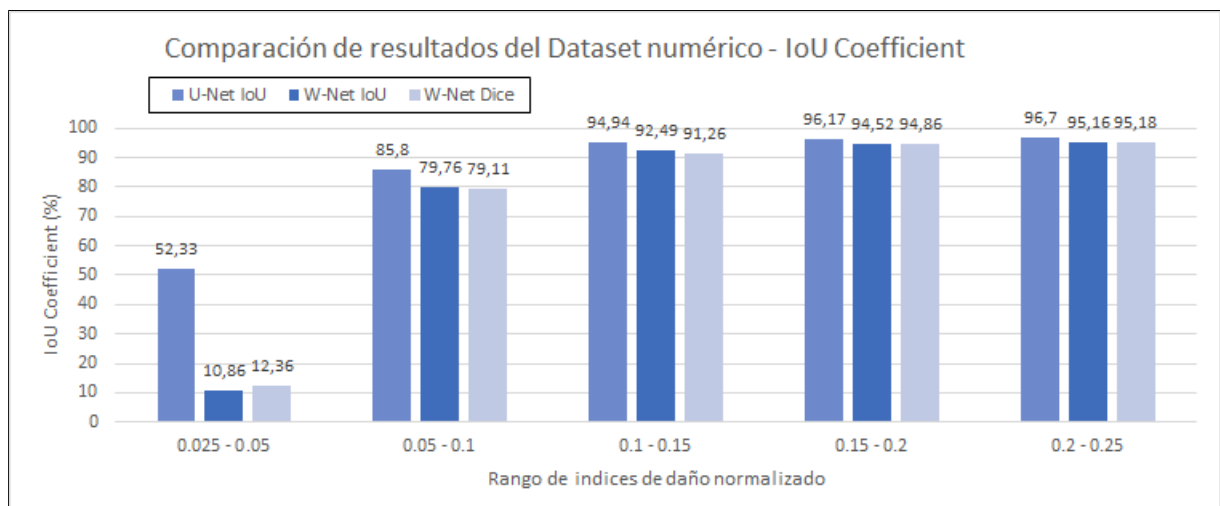


Figura 4.12: Comparación de resultados del dataset numérico - IoU Coefficient.

Tabla 4.10: Comparación de resultados del dataset numérico - Dice Coefficient.

Intervalo	Índice de daño	Dice Coefficient		
		U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
Intervalo 1	0 - 0.025	0,1277	0,00075	0,0107
Intervalo 2	0.025 - 0.05	0,665	0,1932	0,2196
Intervalo 3	0.05 - 0.1	0,9234	0,8863	0,8813
Intervalo 4	0.1 - 0.15	0,974	0,9609	0,954
Intervalo 5	0.15 - 0.2	0,9805	0,9718	0,9736
Intervalo 6	0.2 - 0.25	0,9832	0,9752	0,9753
Dataset completo	0 - 0.25	0,972	0,9584	0,958

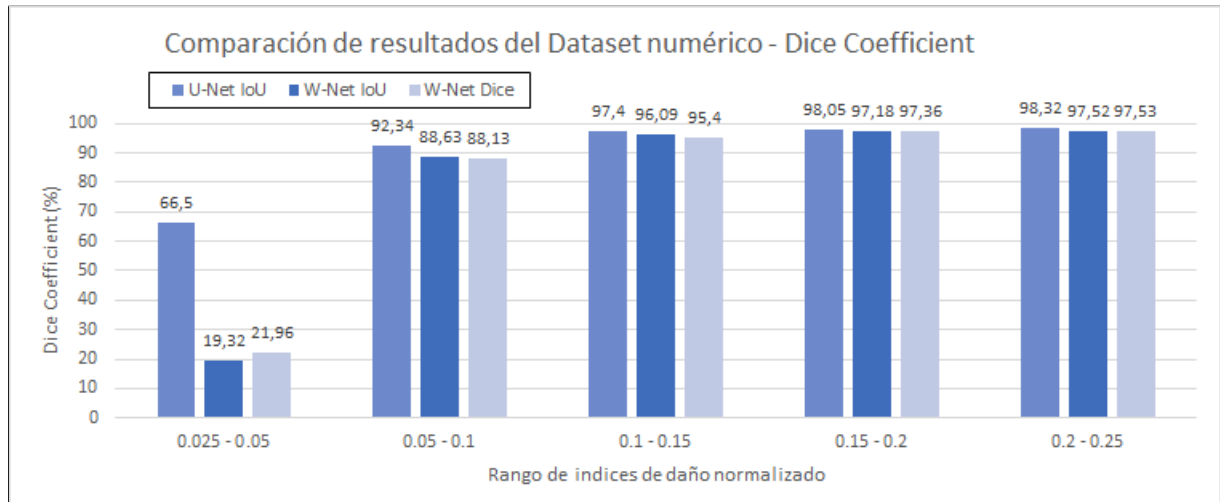


Figura 4.13: Comparación de resultados del dataset numérico - Dice Coefficient.

Tabla 4.11: Comparación de resultados del dataset numérico - Adjusted Rand score.

Intervalo	Índice de daño	Adjusted Rand score		
		U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
Intervalo 1	0 - 0.025	0,9656	0,8518	0,8395
Intervalo 2	0.025 - 0.05	0,4171	0,8737	0,8787
Intervalo 3	0.05 - 0.1	0,9223	0,8569	0,853
Intervalo 4	0.1 - 0.15	0,9657	0,8832	0,8862
Intervalo 5	0.15 - 0.2	0,9775	0,8354	0,8254
Intervalo 6	0.2 - 0.25	0,9788	0,7898	0,7859
Dataset completo	0 - 0.25	0,9053	0,8495	0,8452

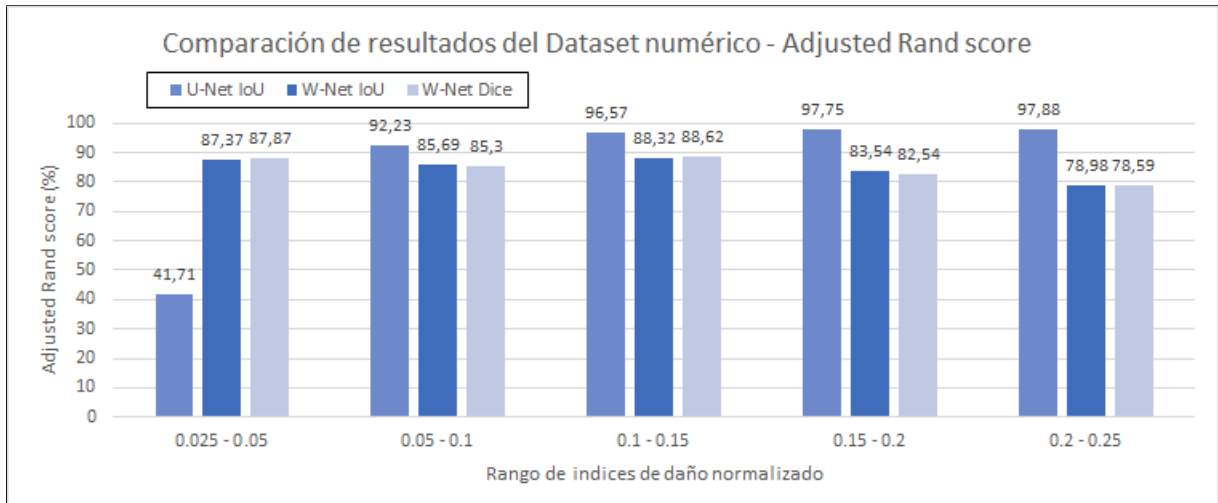


Figura 4.14: Comparación de resultados del dataset numérico - Adjusted Rand score.

Tabla 4.12: Comparación de resultados del dataset numérico - Adjusted Mutual Info score.

Intervalo	Índice de daño	Adjusted Mutual Info score		
		U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
Intervalo 1	0 - 0.025	0,9617	0,81	0,7992
Intervalo 2	0.025 - 0.05	0,3672	0,8294	0,8337
Intervalo 3	0.05 - 0.1	0,8611	0,8168	0,8109
Intervalo 4	0.1 - 0.15	0,925	0,8426	0,8456
Intervalo 5	0.15 - 0.2	0,9421	0,7939	0,7878
Intervalo 6	0.2 - 0.25	0,944	0,7484	0,7486
Dataset completo	0 - 0.25	0,8683	0,8082	0,8049

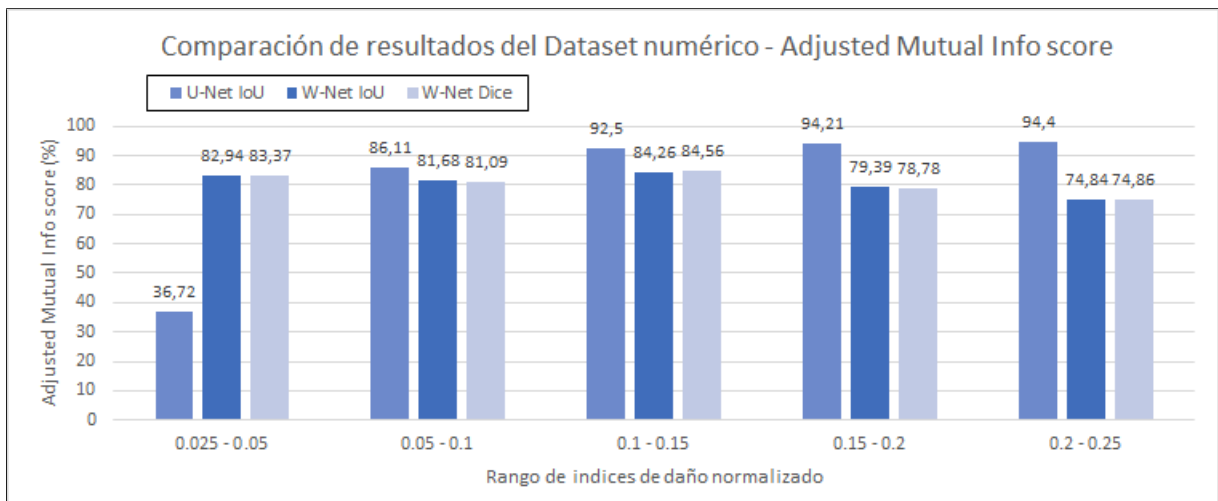


Figura 4.15: Comparación de resultados del dataset numérico - Adjusted Mutual Info score.

4.4.2. Dataset experimental

Siguiendo el formato de los resultados del dataset numérico, a continuación se presentan las tablas comparativas de los resultados del dataset experimental, con sus respectivos gráficos.

Tabla 4.13: Comparación de resultados del dataset experimental - Caso 1.

Métrica	Caso 1		
	U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
IoU	0,3331	0,4942	0,4486
Dice	0,4997	0,6615	0,6193
PRI	0,4809	0,6475	0,6002
VI	0,3204	0,4709	0,4163

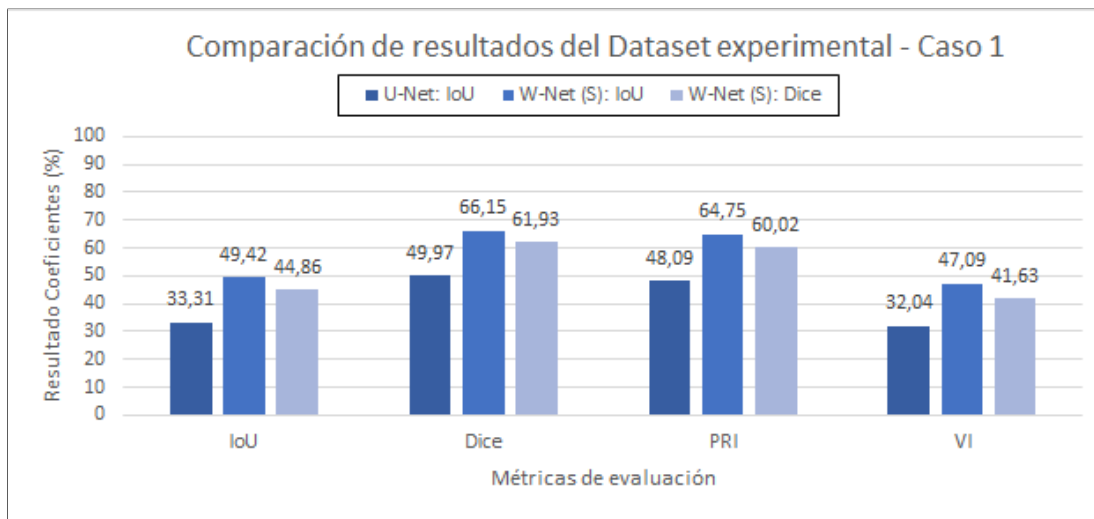


Figura 4.16: Comparación de resultados del dataset experimental - Caso 1.

Tabla 4.14: Comparación de resultados del dataset experimental - Caso 2.

Métrica	Caso 2		
	U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
IoU	0,3093	0,4953	0,468
Dice	0,4725	0,6625	0,6376
PRI	0,4412	0,6378	0,6079
VI	0,2785	0,5225	0,4946

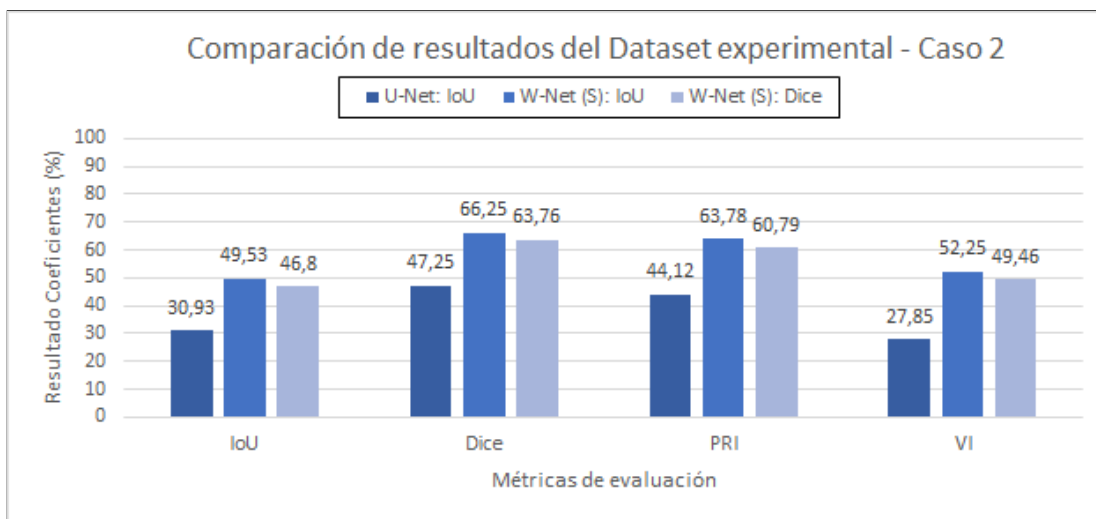


Figura 4.17: Comparación de resultados del dataset experimental - Caso 2.

Tabla 4.15: Comparación de resultados del dataset experimental - Caso 3.

Métrica	Caso 3		
	U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
IoU	0,4189	0,4146	0,4482
Dice	0,5904	0,5861	0,6189
PRI	0,5354	0,531	0,5621
VI	0,4148	0,4266	0,4362

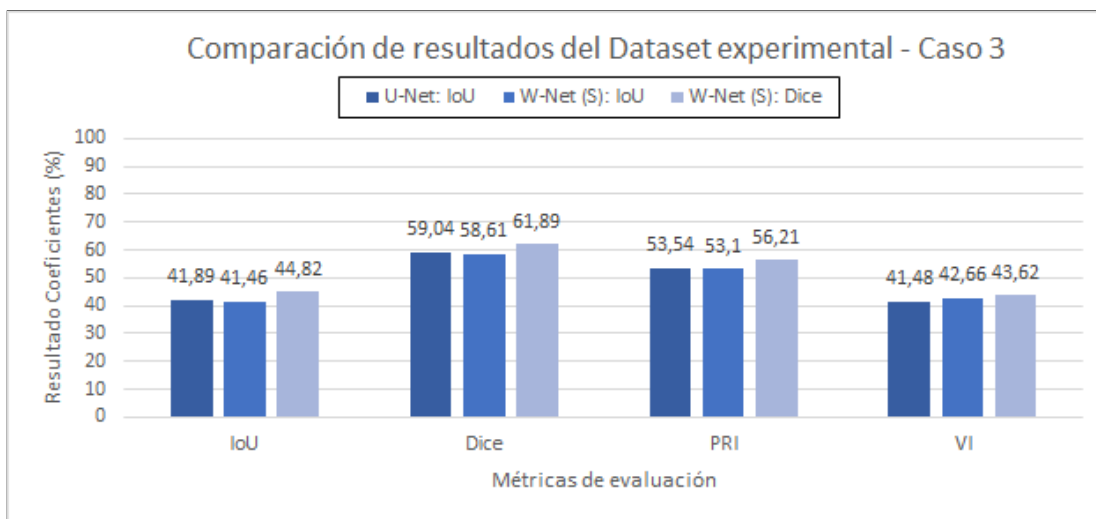


Figura 4.18: Comparación de resultados del dataset experimental - Caso 3.

Tabla 4.16: Comparación de resultados del dataset experimental - Caso 4.

Métrica	Caso 4		
	U-Net: IoU	W-Net (S): IoU	W-Net (S): Dice
IoU	0,4471	0,5347	0,5848
Dice	0,618	0,6968	0,738
PRI	0,5446	0,6232	0,669
VI	0,4308	0,44649	0,5139

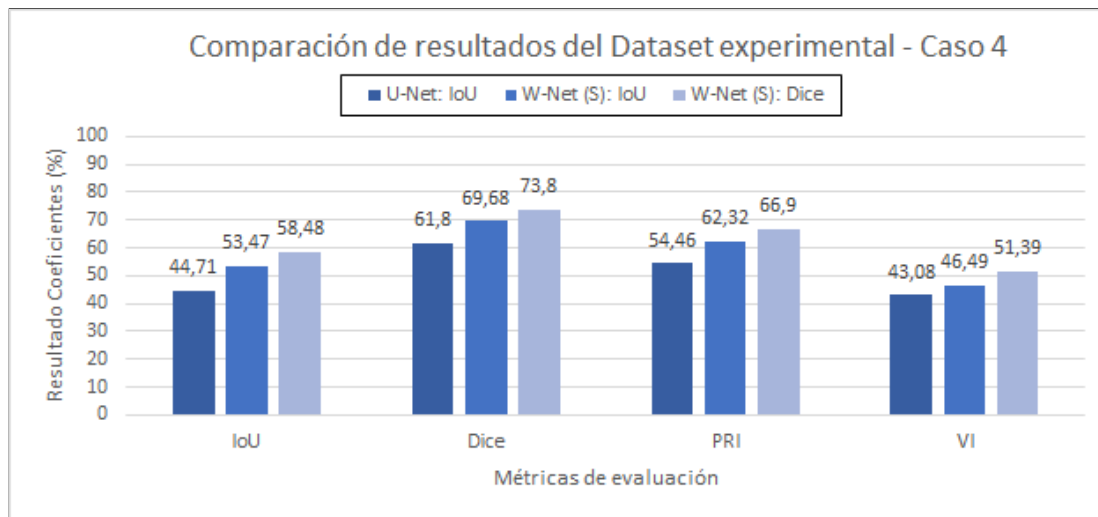


Figura 4.19: Comparación de resultados del dataset experimental - Caso 4.

Capítulo 5

Análisis de resultados

Antes de comenzar con las comparaciones y los análisis de efectividad de cada modelo, es importante mencionar y hacer notar que los resultados del trabajo anterior del U-Net no son idénticos a los obtenidos con el modelo entrenado disponible del U-Net en este trabajo. Ésto se debe a que el modelo fue reentrenado luego de dicho trabajo (no fue posible conseguir detalles del reentrenamiento) y, a grandes rasgos, se podría decir que los entrenamientos posteriores mejoraron su efectividad en la predicción del dataset numérico, realizando más predicciones correctas (aunque no perfectas) de zonas con bajo índice de daño (Intervalo 2) y conservando la efectividad para zonas con daño por delaminación medianas y grandes (Intervalos 3-6). Pero, en cuanto a resultados experimentales, la efectividad de predicción se redujo, más considerablemente en los casos 1 y 2. Por lo tanto, es necesario aclarar que las comparaciones se harán con los resultados rescatados en este trabajo y no con los expuestos en [1].

Partiendo con los procesos de optimización, se encontraron algunos comportamientos extraños que se pueden analizar. En primer lugar, las predicciones hechas con el hiperparámetro “Técnica de regularización” configurado en “Batch Normalization + Dropout (20%)” (expuestas en las **Figuras 4.3 y 4.8**) tienden a detectar los esfuerzos de sujeción en el borde del panel como daño por delaminación. Ésto sucede únicamente al predecir imágenes con bajo índice de daño normalizado y en mayor grado según disminuye el tramo de convergencia de la fase de entrenamiento, como se ve en la **Figura 4.7**. El fenómeno se produce porque el modelo no logra aprender lo suficiente de las etiquetas en pocas épocas, resultando en que el algoritmo no descarte el borde del panel como daño por delaminación.

En el apartado de learning rate, se pudo observar como las dos primeras opciones, $LR = 0.000001$ y $LR = 0.00001$, son valores muy pequeños para que el entrenamiento converja en ambos modelos del W-Net supervisado. Es posible que alcancen mejores resultados que la selección óptima de este trabajo, pero necesitan ignorar el factor *tiempo de entrenamiento*.

El último comportamiento anormal presente en el proceso de optimización de hiperparámetros, es el hecho de que el algoritmo “W-Net supervisado: IoU Loss” no pudo completar el entrenamiento con la cantidad de canales configurada en “64-128-256-512-1024” debido a falta de memoria RAM. Esto habla netamente del factor “recursos necesarios” para el entrenamiento, dejando en evidencia la relación existente entre la complejidad de la arquitectura y la capacidad de procesamiento necesaria (Hardware). Aún así, la optimización con la función

de costo “Dice Loss” no tuvo inconvenientes para realizar el entrenamiento mencionado, por lo que podemos asumir que la complejidad de la función de costo implementada en la fase de entrenamiento también es un factor a la hora de estimar el Hardware necesario para llevar a cabo un entrenamiento.

Ahora, pasando al análisis de los resultados numéricos, se observó que los tres modelos evaluados son capaces de detectar daño por delaminación desde valores de índice de daño normalizado 0.35. En los dos modelos de W-Net supervisado, la detección del daño a medida que aumenta el índice normalizado, es de carácter intermitente, es decir, algunas veces lo detecta y otras no. No así, el modelo del U-Net, que tiene un límite claro de detección, a partir del cual la mayor parte de imágenes obtiene una predicción de daño correcta. Ésto se puede ver reflejado en los gráficos de las **Figuras 4.12 y 4.13** y en las predicciones mostradas en el capítulo de **Anexos 7.1, 7.3 y 7.5**. El comportamiento de intermitencia en la detección se ve ya regularizado en el intervalo 3 (0.05 - 0.1 índice de daño normalizado). Los gráficos de comparación de las métricas “Adjusted Rand score” y “Adjusted Mutual Info score” (presentes en las **Figuras 4.14 y 4.15**) no representan el comportamiento de intermitencia debido a que su enfoque es la efectividad del clustering, mermando el hecho de que la predicción es errónea por la pequeña cantidad de píxeles que están mal agrupados.

También, es posible observar en las **Tablas 4.9, 4.10, 4.11 y 4.12** que el modelo U-Net consigue una mayor efectividad de predicción, independiente de la métrica de evaluación. Así mismo, se distingue la igualdad de efectividad de ambos modelos W-Net, lo que propone que, funciones de optimización similares no presentan cambios significativos en el desempeño de predicción.

Los resultados experimentales fueron analizados caso a caso. En el “Caso 1”, los mejores resultados se obtuvieron con el modelo “W-Net supervisado: IoU Loss” para todas las métricas de evaluación (**Figura 4.16 y Tabla 4.13**). De igual manera, las predicciones del modelo “W-Net supervisado: IoU Loss” alcanzaron el mayor valor de efectividad en el “Caso 2”, independiente de la métrica de evaluación que se visualice (**Figura 4.17 y Tabla 4.14**). No se puede afirmar lo mismo para los “Casos 3 y 4”, donde la superioridad de predicción es alcanzada por el modelo “W-Net supervisado: Dice Loss” en todas las métricas evaluadas (**Figuras 4.18, 4.19 y Tablas 4.15, 4.16**). El hecho de que la comparación de desempeño en predicción es similar, en cada caso y para todas las métricas, es argumento suficiente para afirmar que los resultados no dependen de la métrica utilizada, lo que se reafirma al comparar las imágenes de las **Figuras 4.2, 4.6 y 4.11**.

Luego del último punto analizado, es interesante buscar una explicación al hecho de que el modelo “U-Net: IoU Loss” obtenga mejores resultados que los otros modelos en cuanto a resultados numéricos, pero se ve superado por ellos al comparar los resultados experimentales. Esto puede ser debido a que el modelo “U-Net: IoU Loss” haya sufrido *overfitting* durante su proceso de reentrenamiento, causando que imágenes menos similares al dataset de entrenamiento, como es el caso del dataset experimental, obtengan predicciones de menor eficacia. Como no se pudieron recolectar detalles del proceso mencionado, no se puede afirmar esta aseveración.

Todos los análisis de comparación realizados entre los algoritmos U-Net y W-Net supervisado tienen en consideración la diferencia dimensional entre imágenes del dataset numérico introducidas a cada uno de los algoritmos (96x128 px. para U-Net y 64x80 px. para W-Net). Si bien éste factor puede sesgar la comparación a favor del U-Net, no es posible determinar las dimensiones del sesgo. Sin embargo, ésto, junto al hecho mencionado en la optimización del hiperparámetro “Cantidad de canales” en el algoritmo “W-Net supervisado: IoU Loss”, deja en evidencia que el algoritmo W-Net supervisado requiere una mayor capacidad de cómputo (Hardware) que el algoritmo U-Net para llevar a cabo entrenamientos idénticos.

Capítulo 6

Conclusiones

A continuación se presentan las conclusiones de este trabajo.

- Fue posible crear un algoritmo de aprendizaje profundo supervisado funcional a partir de la arquitectura de un algoritmo no supervisado. El algoritmo fue optimizado con dos funciones de costo diferentes y los resultados que se obtuvieron son coherentes. Gracias a esto, se pudo llevar a cabo una comparación fidedigna entre ambos algoritmos, U-Net y W-Net supervisado.
- Los resultados de este trabajo indican que el algoritmo “U-Net” es más eficaz que el algoritmo “W-net supervisado” al momento de predecir las imágenes del dataset numérico. Sin embargo, el U-Net se vió superado por ambos modelos W-Net supervisados cuando las predicciones se hicieron sobre las imágenes del dataset experimental.
- El comportamiento general de los resultados numéricos de los tres modelos es similar según los intervalos que se evalúan. También el comportamiento general de los resultados experimentales de todos los modelos evaluados es similar, en cada caso e independiente de la métrica que se observe.
- Bajo los parámetros que se escogieron en este trabajo (dimensión de imágenes, hiperparámetros de optimización, arquitecturas, entre otros), se confirmó que la arquitectura W-Net requiere un computador con mayor capacidad de procesamiento (Memoria RAM) que el algoritmo U-Net para llevar a cabo entrenamientos idénticos.
- Como primera aproximación a una comparación entre algoritmos supervisados y no supervisados, se obtuvieron conclusiones significativas que sustentan el prejuicio de que los algoritmos no supervisados tienen un alto requerimiento de hardware para ser entrenados. Sin embargo, las conclusiones no aportan hechos que apoyen significativamente el prejuicio de que un algoritmo no supervisado entregará mejores resultados.

Éste estudio comparativo, entre algoritmos supervisados y no supervisados, se puede profundizar en varios aspectos. A continuación dejo propuestos algunos de ellos:

- Realizar una comparación directa entre el algoritmo supervisado U-Net y el algoritmo no supervisado W-Net.
- Ampliar la aplicación del estudio. Se pueden hacer comparaciones en distintos set de imágenes provenientes de distintos campos de la industria.

- Aumentar complejidad de las imágenes de estudio, por ejemplo en la cantidad de canales de colores.
- Aumentar complejidad en la segmentación a más de dos, o dejando que el algoritmo escoja la cantidad de segmentos óptima.
- Explorar más funciones de costo para la optimización de los algoritmos.
- Agregar más hiperparámetros de optimización o métricas de evaluación.

Bibliografía

- [1] R. R. E. L. D. Viviana Meruane, Diego Aichele, “Damage assessment in composite sandwich structures based on full-field vibration measurements and deep learning.” Submitted to Structural Control and Health Monitoring, 2020.
- [2] V. Meruane, I. Fernandez, R. O. Ruiz, G. Petrone, and E. Lopez-Droguett, “Gapped gaussian smoothing technique for debonding assessment with automatic thresholding,” *Structural Control and Health Monitoring*, vol. 26, no. 8, p. e2371, 2019.
- [3] F. Seguel and V. Meruane, “Damage assessment in a sandwich panel based on full-field vibration measurements,” *Journal of Sound and Vibration*, vol. 417, pp. 1–18, 2018.
- [4] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [5] X. Xia and B. Kulis, “W-net: A deep model for fully unsupervised image segmentation,” *arXiv preprint arXiv:1711.08506*, 2017.
- [6] S. L. developers, “scikit-learn 0.24.1, api reference, metrics.” url: <https://scikit-learn.org/stable/modules/classes.html>, 2007-2020. Accedido 21-03-2021.

Capítulo 7

Anexos

7.1. Predicciones dataset numérico - U-Net: IoU Loss

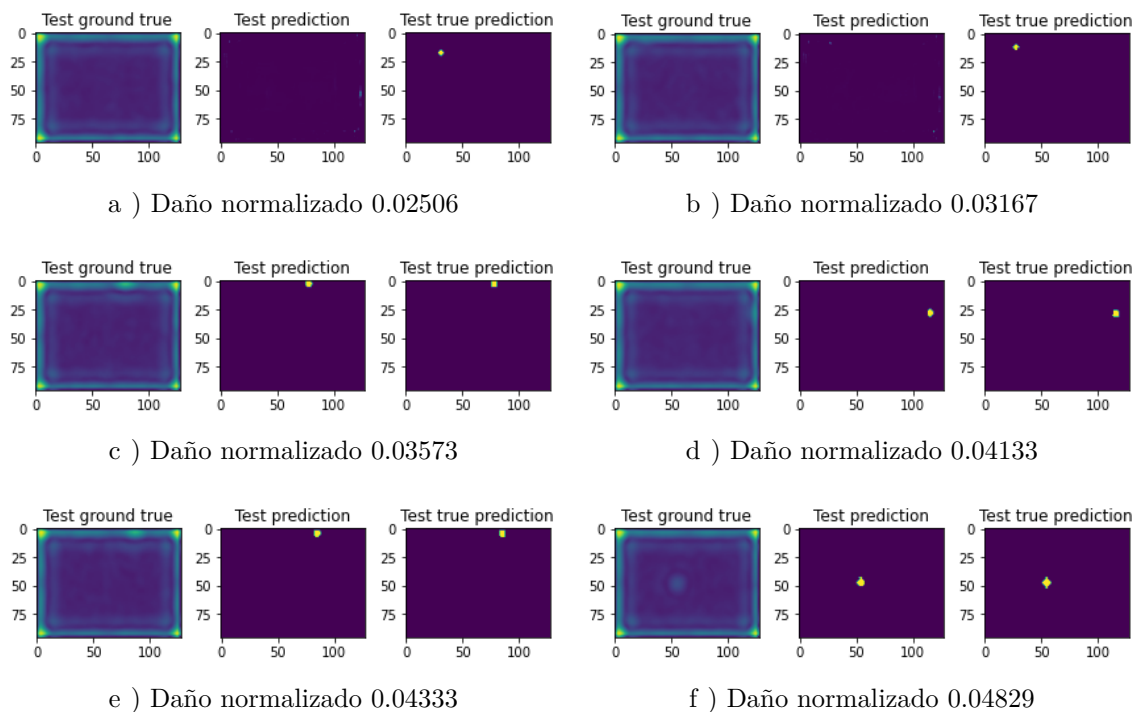


Figura 7.1: Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 2.

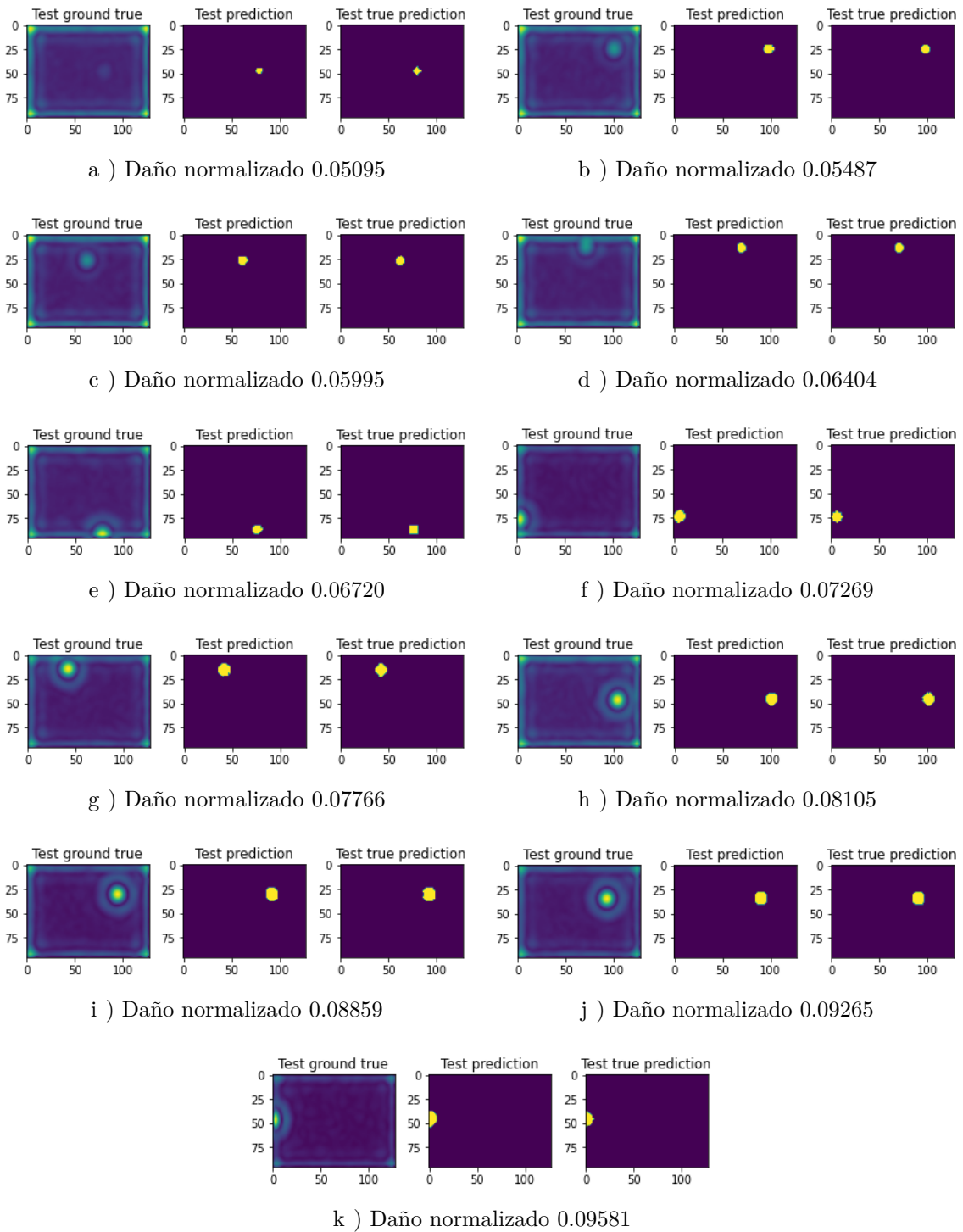


Figura 7.2: Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 3.

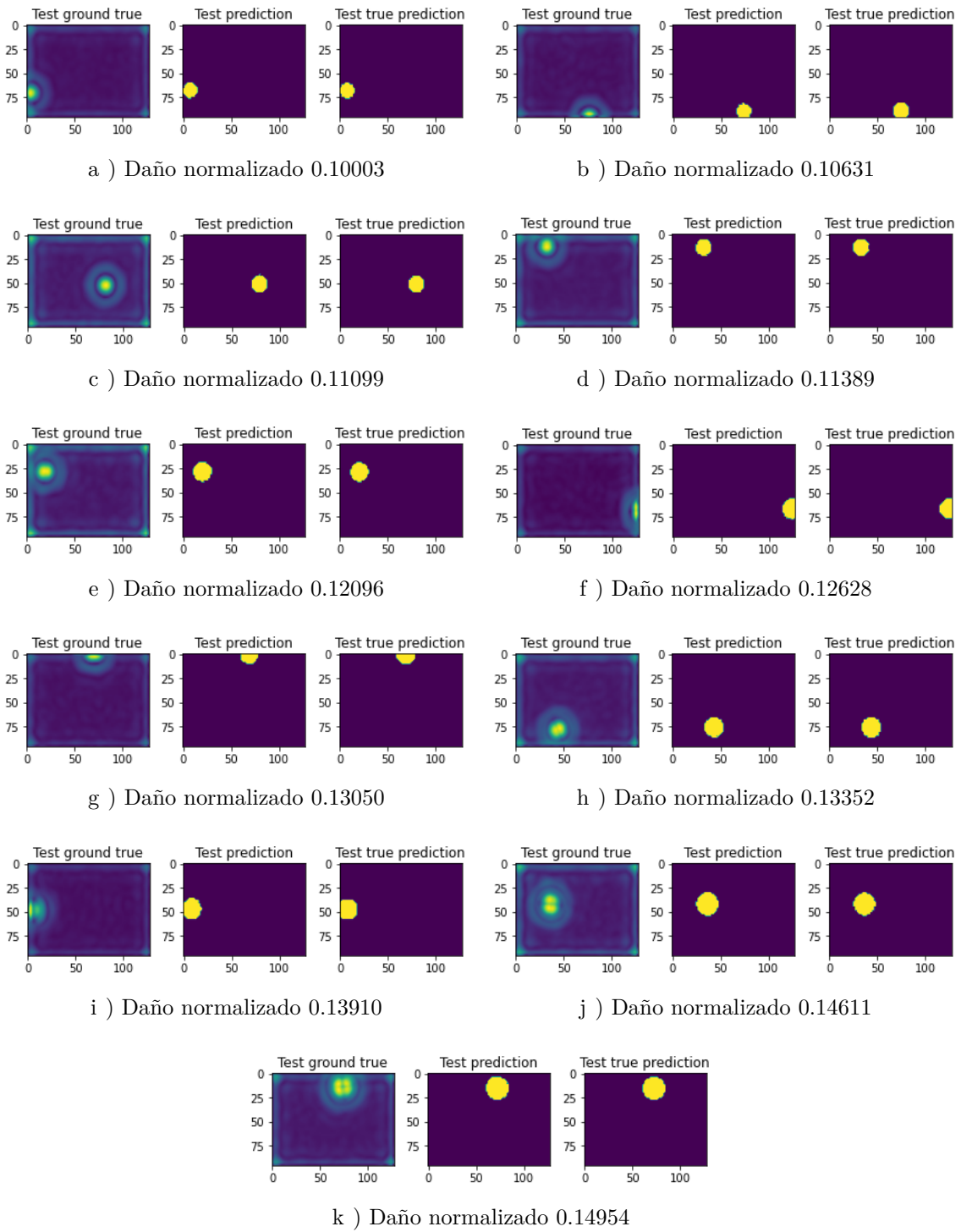


Figura 7.3: Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 4.

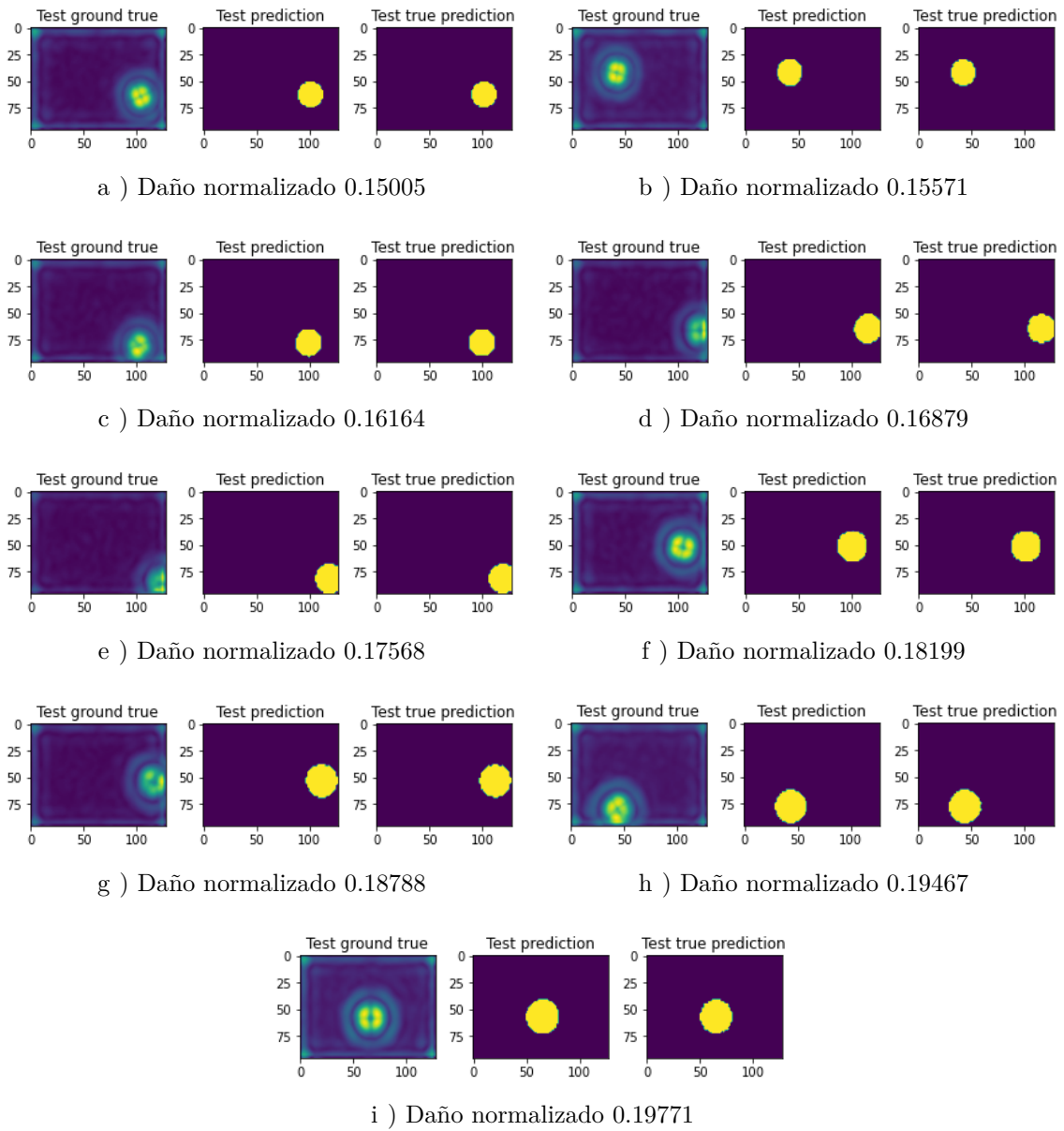


Figura 7.4: Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 5.

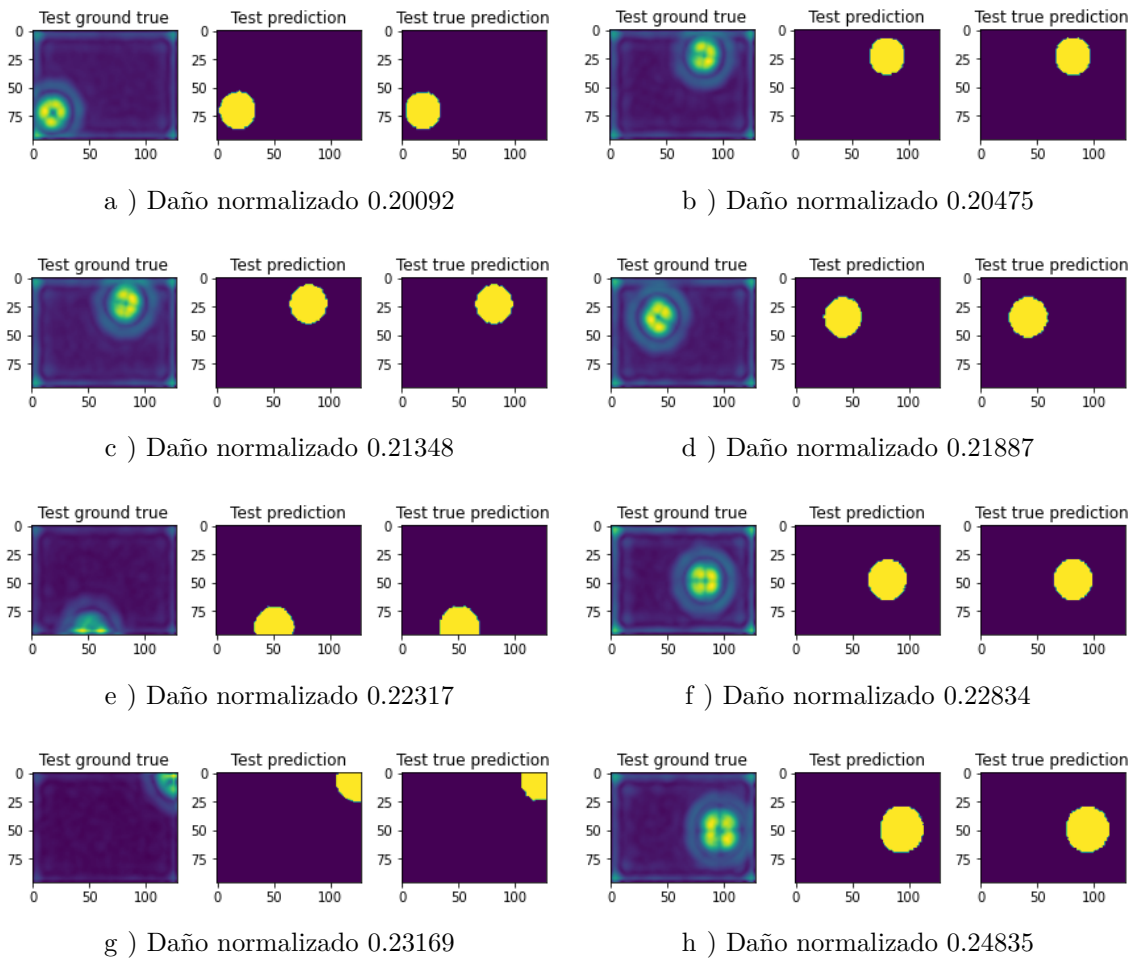


Figura 7.5: Predicciones realizadas por el modelo U-Net: IoU Loss. Dataset numérico de testing - Intervalo 6.

7.2. Proceso de optimización de hiperparámetros - W-Net (S): IoU Loss

Tabla 7.1: Resultados de optimización de la **técnica de regularización** para el algoritmo W-Net supervisado: IoU Loss.

Técnica de regularización	Dataset de validación		
	IoU Coefficient	Promedio	Desviación estándar
Sin regularización	0,9154 0,9192 0,9185 0,9171 0,91	0,9160	0,0037
Batch Normalization	0,9211 0,648 0,6013 0,9291 0,3532	0,6905	0,2417
Dropout (20 %)	0,8404 0,8492 0,8048 0,8374 0,7908	0,8245	0,0253
Batch Normalization + Dropout (20 %)	0,5549 0,6404 0,559 0,7461 0,5397	0,6080	0,0866

Tabla 7.2: Resultados de optimización de **learning rate** para el algoritmo W-Net supervisado: IoU Loss.

Learning rate	Dataset de validación		
	IoU Coefficient	Promedio	Desviación estándar
LR = 0.000001	0,7419 0,7358 0,7355 0,7337 0,7427	0,7379	0,0041
LR = 0.00001	0,8821 0,889 0,8865 0,8892 0,8933	0,8880	0,0041
LR = 0.0001	0,9163 0,9289 0,9185 0,9268 0,9304	0,9242	0,0064
LR = 0.001	0,9062 0,894 0,8892 0,8881 0,8998	0,8955	0,0076

Tabla 7.3: Resultados de optimización de **cantidad de canales** para el algoritmo W-Net supervisado: IoU Loss.

Cantidad de canales	Dataset de validación		
	IoU Coefficient	Promedio	Desviación estándar
16-64-128-256-512	0,9186 0,9102 0,9207 0,9232 0,9163	0,9178	0,005
32-64-128-256-512	0,9302 0,9294 0,9245 0,9314 0,9192	0,9269	0,0051
64-128-256-512-1024	- - - - -	-	-

Tabla 7.4: Selección del modelo para obtención de resultados con el algoritmo W-Net supervisado: IoU Loss.

Cantidad de canales	Dataset de testing		
	IoU Coefficient	Promedio	Desviación estándar
32-64-128-256-512	0,9236 0,9248 0,9202 0,9237 0,9096	0,9204	0,0063

7.3. Predicciones dataset numérico - W-Net (S): IoU Loss

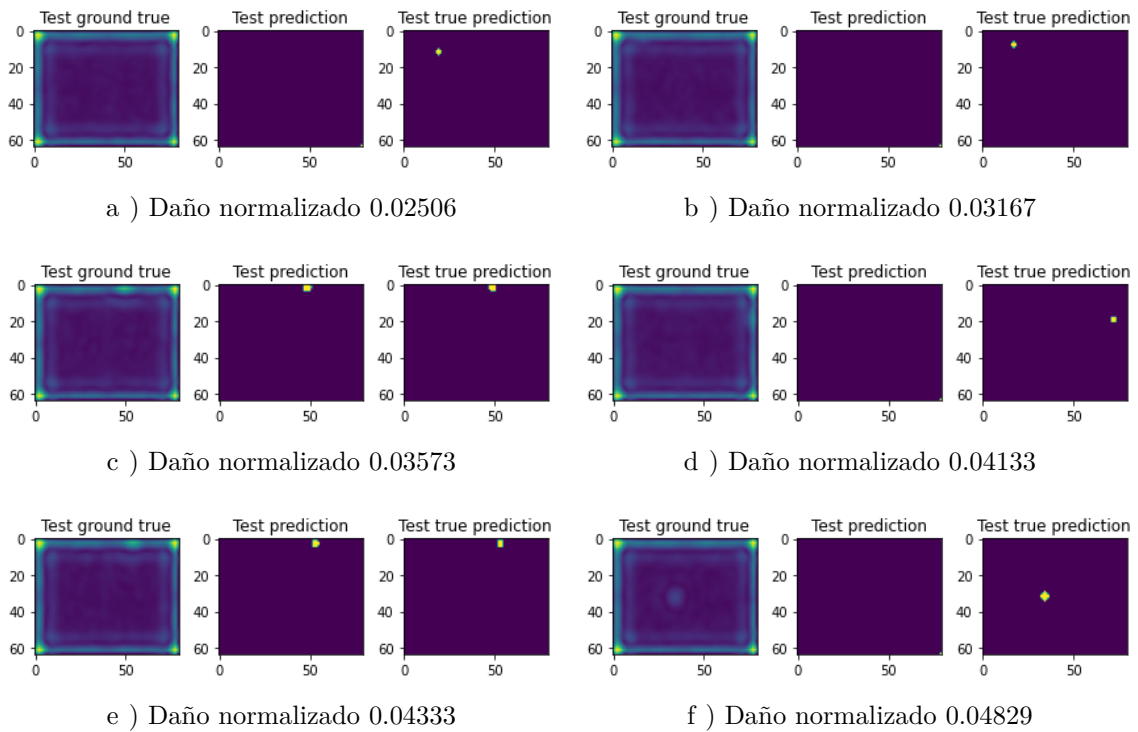


Figura 7.6: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 2.

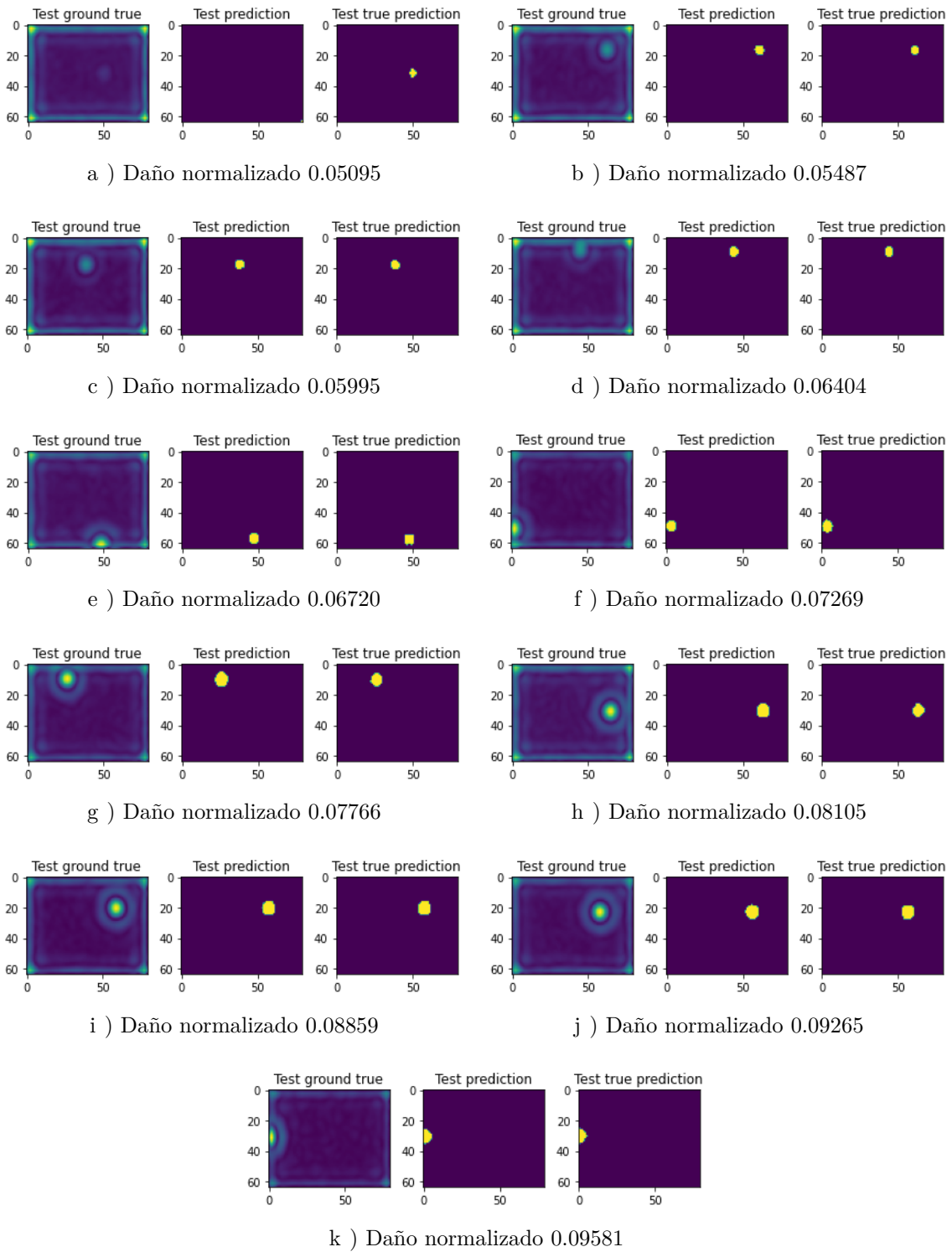


Figura 7.7: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 3.

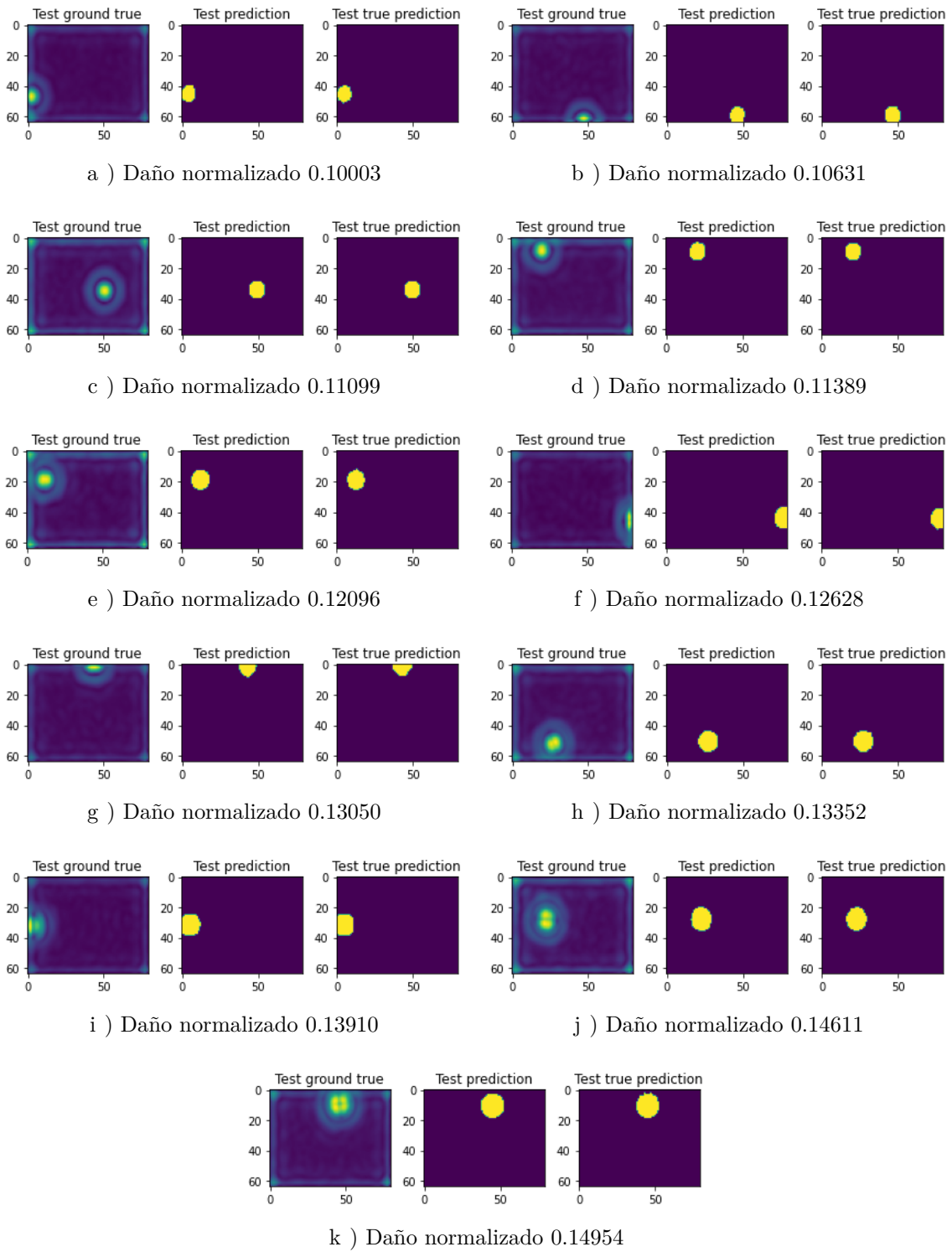


Figura 7.8: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 4.

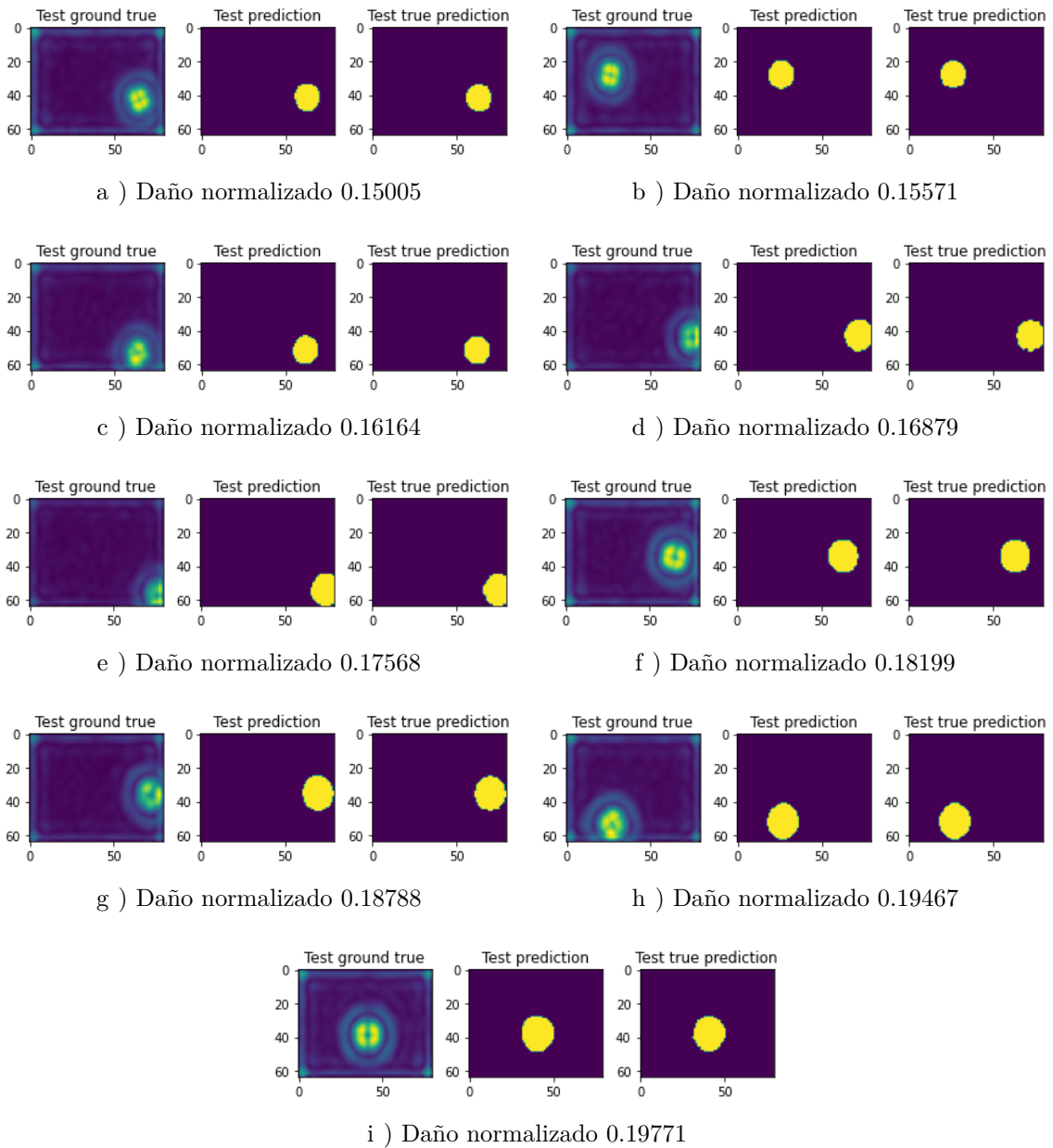


Figura 7.9: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 5.

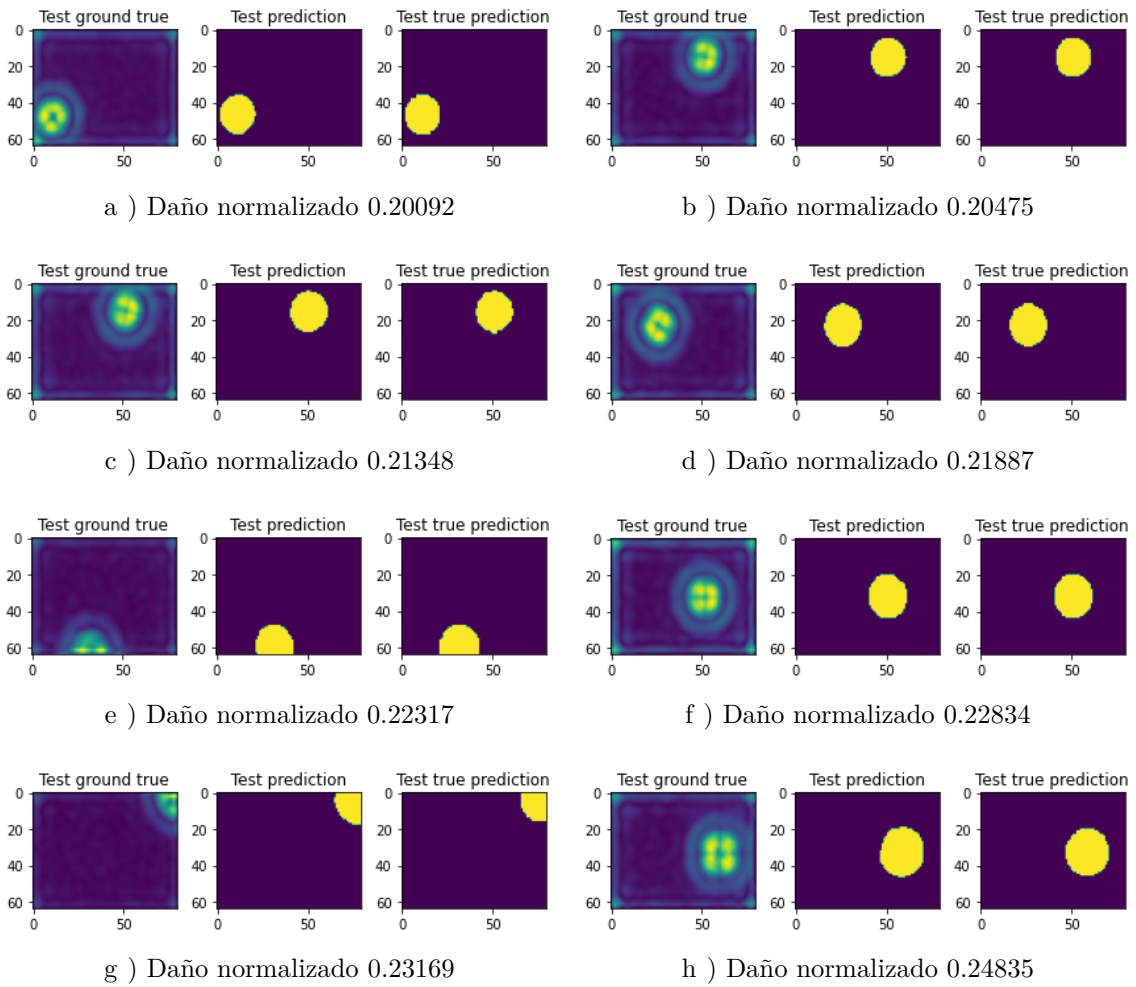


Figura 7.10: Predicciones realizadas por el modelo W-Net supervisado: IoU Loss. Dataset numérico de testing - Intervalo 6.

7.4. Proceso de optimización de hiperparámetros - W-Net (S): Dice Loss

Tabla 7.5: Resultados de optimización de la **técnica de regularización** para el algoritmo W-Net supervisado: Dice Loss.

Técnica de regularización	Dataset de validación		
	Dice Coefficient	Promedio	Desviación estándar
Sin regularización	0,9646 0,9607 0,9643 0,9686 0,9636	0,9644	0,0028
Batch Normalization	0,9666 0,9662 0,753 0,6148 0,9653	0,8532	0,1621
Dropout (20%)	0,9361 0,932 0,927 0,7136 0,9345	0,8886	0,0979
Batch Normalization + Dropout (20%)	0,6247 0,7192 0,644 0,7403 0,8516	0,716	0,0901

Tabla 7.6: Resultados de optimización de **learning rate** para el algoritmo W-Net supervisado: Dice Loss.

Learning rate	Dataset de validación		
	Dice Coefficient	Promedio	Desviación estándar
LR = 0.000001	0,8492 0,8545 0,8466 0,8489 0,8501	0,8499	0,0029
LR = 0.00001	0,9381 0,9409 0,9375 0,939 0,9445	0,94	0,0028
LR = 0.0001	0,9634 0,9592 0,9496 0,9584 0,9623	0,9586	0,0054
LR = 0.001	0,9328 0,9455 0,9427 0,9503 0,9496	0,9442	0,0071

Tabla 7.7: Resultados de optimización de **cantidad de canales** para el algoritmo W-Net supervisado: Dice Loss.

Cantidad de canales	Dataset de validación		
	Dice Coefficient	Promedio	Desviación estándar
16-64-128-256-512	0,9628 0,9577 0,9578 0,9556 0,9597	0,9587	0,0027
32-64-128-256-512	0,9597 0,9655 0,9657 0,9602 0,9635	0,9629	0,0028
64-128-256-512-1024	0,9642 0,962 0,9587 0,9598 0,9586	0,9607	0,0024

Tabla 7.8: Selección del modelo para obtención de resultados con el algoritmo W-Net supervisado: Dice Loss.

Cantidad de canales	Dataset de testing		
	Dice Coefficient	Promedio	Desviación estándar
32-64-128-256-512	0,9536 0,9632 0,9611 0,9577 0,9588	0,9589	0,0036

7.5. Predicciones dataset numérico - W-Net (S): Dice Loss

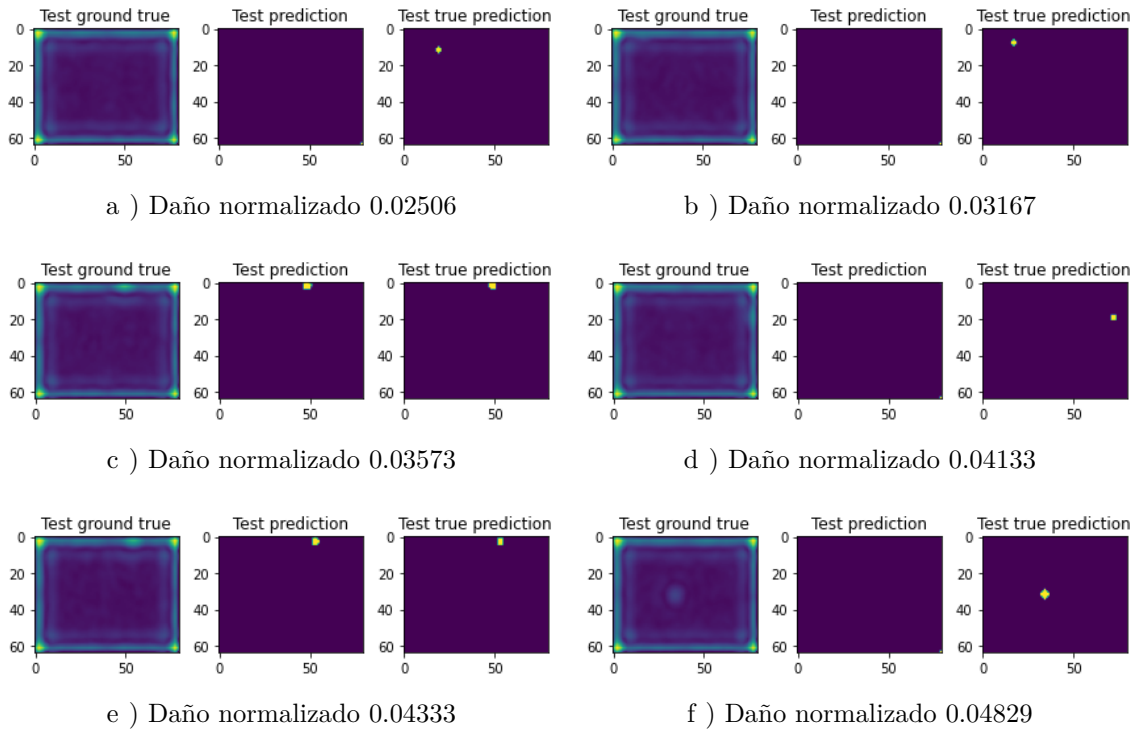


Figura 7.11: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 2.

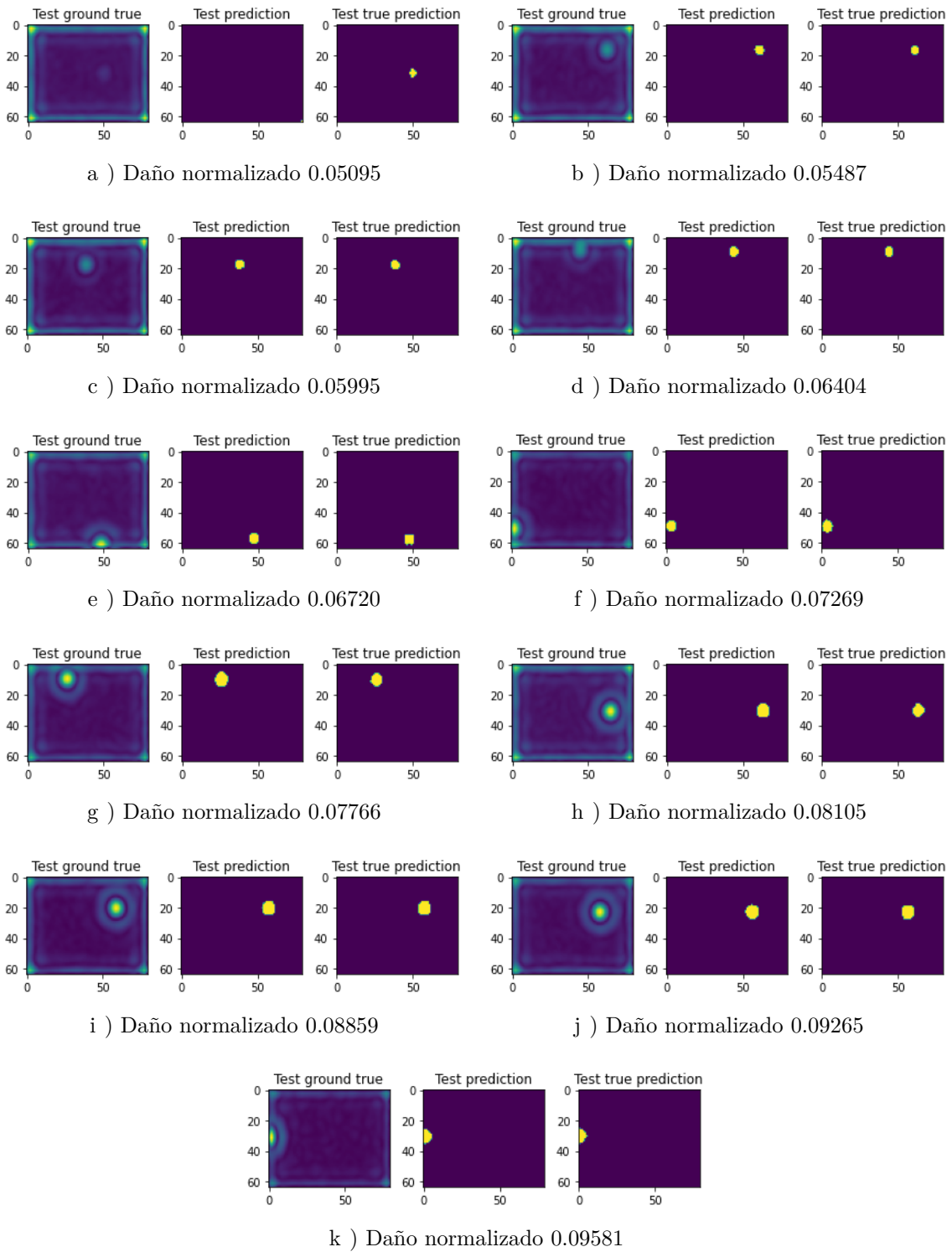


Figura 7.12: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 3.

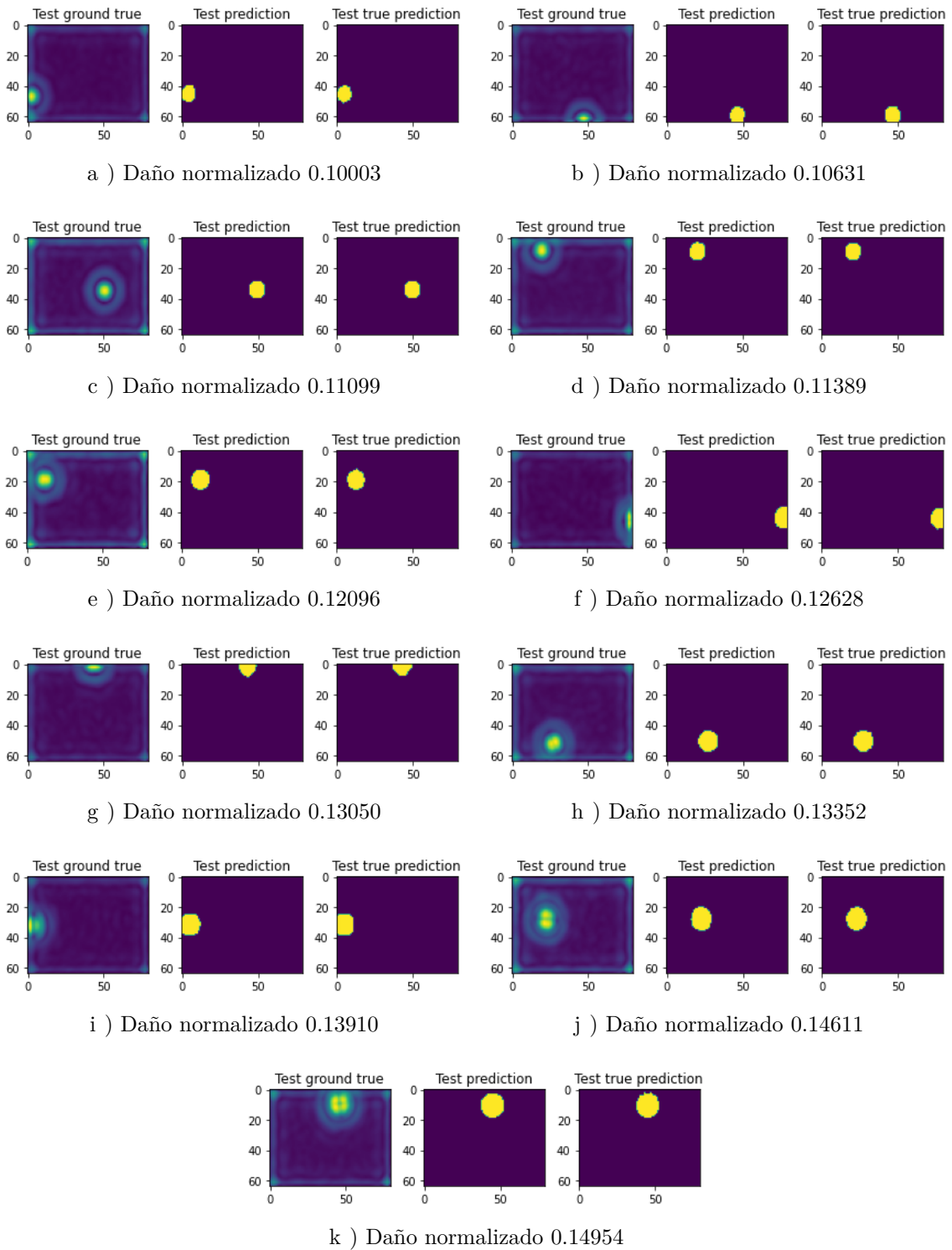


Figura 7.13: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 4.

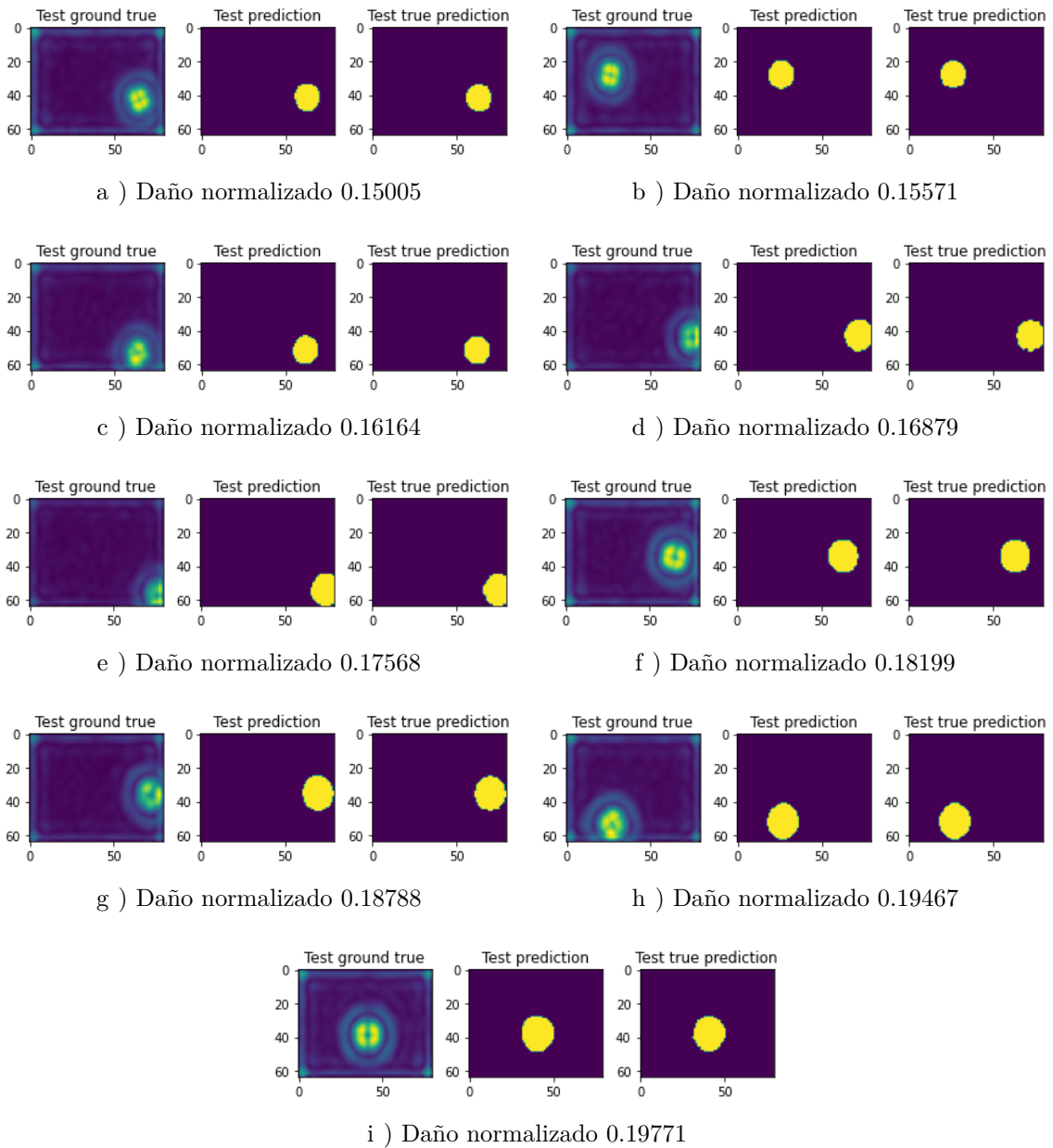


Figura 7.14: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 5.

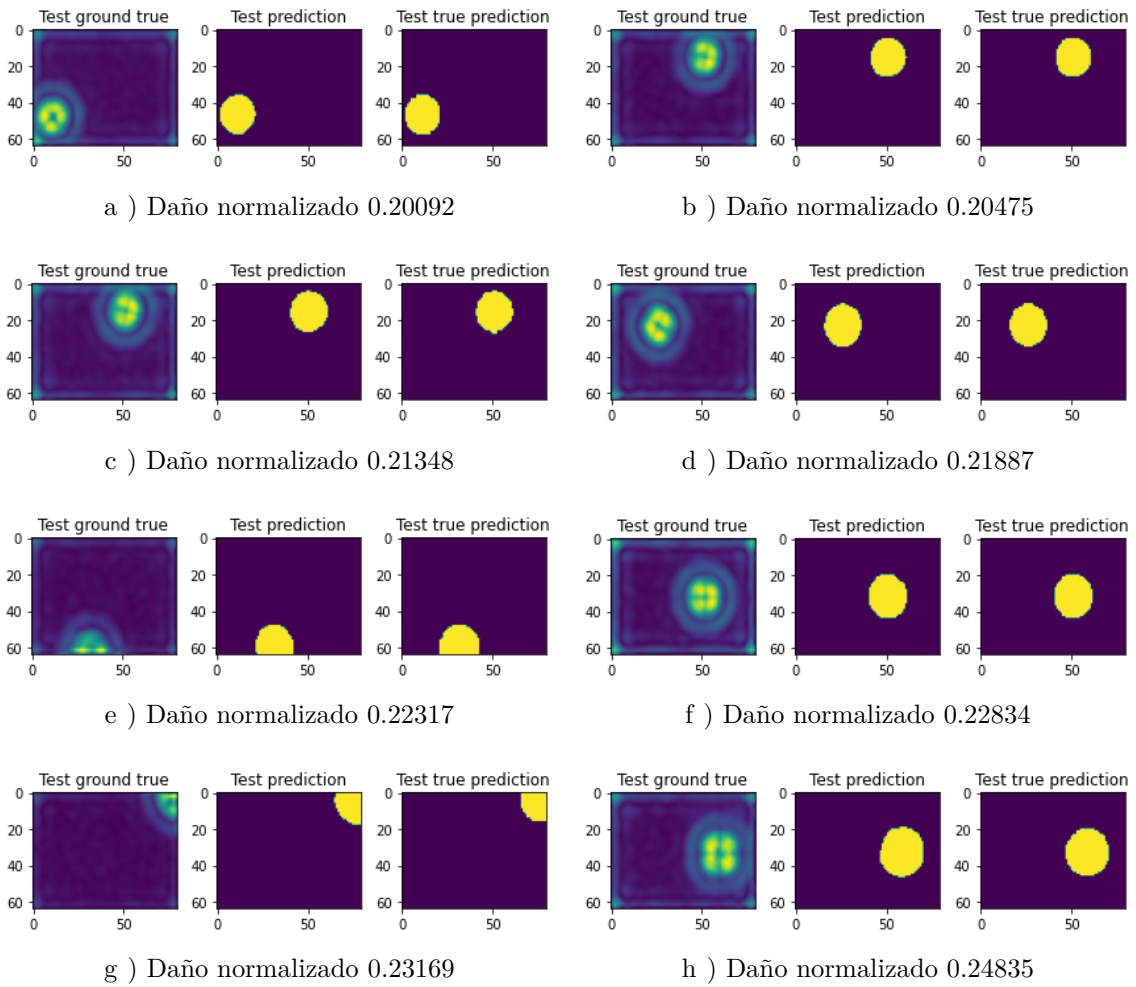


Figura 7.15: Predicciones realizadas por el modelo W-Net supervisado: Dice Loss. Dataset numérico de testing - Intervalo 6.