

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE OBRAS CIVILES

IDENTIFICACIÓN AUTOMÁTICA DE PATRONES DE ROTURA DEL  
OLEAJE EN LA ZONA ROMPIENTE MEDIANTE RED NEURONAL  
CONVOLUCIONAL DENOMINADA U-NET

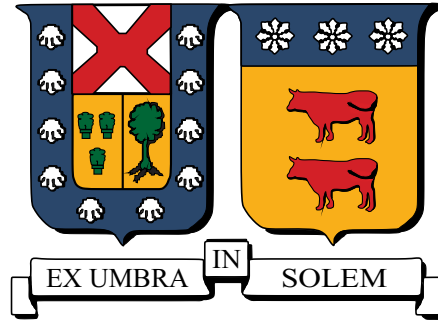
FRANCISCO JAVIER SÁEZ ROSALES

Ingeniero Civil

Magíster en Ciencias de la Ingeniería Civil

Abril de 2022  
Valparaíso, Chile





UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE OBRAS CIVILES

**IDENTIFICACIÓN AUTOMÁTICA DE PATRONES DE ROTURA DEL  
OLEAJE EN LA ZONA ROMPIENTE MEDIANTE RED NEURONAL  
CONVOLUCIONAL DENOMINADA U-NET**

Tesis de Grado y Memoria de Título presentada por  
**FRANCISCO JAVIER SÁEZ ROSALES**

como requisito parcial para optar al título de  
**Ingeniero Civil**

y al grado de  
**Magíster en Ciencias de la Ingeniería Civil**

Profesor Guía  
Dr. Patricio Andrés Catalán Mondaca

Abril de 2022  
Valparaíso, Chile



TÍTULO DE LA TESIS

**IDENTIFICACIÓN AUTOMÁTICA DE PATRONES DE ROTURA DEL OLEAJE EN LA ZONA ROMPIENTE MEDIANTE RED NEURONAL CONVOLUCIONAL DENOMINADA U-NET**

AUTOR

**FRANCISCO JAVIER SÁEZ ROSALES**

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el título de **Ingeniero Civil** y el grado de **Magíster en Ciencias de la Ingeniería Civil** de la Universidad Técnica Federico Santa María.

Dr. Patricio Catalán M. \_\_\_\_\_

Dr. Carlos Valle V. \_\_\_\_\_

Dr. Raúl Flores A. \_\_\_\_\_

Valparaíso, Chile, Abril de 2022.



*” Mis ideas son mías solo hasta que las programo  
Lo único que es mío son las ganas con las que lo hago  
Ojalá algún día pa vivir de esto se de la mano”*

pa’ mi abuelita



# Resumen

La rotura del oleaje es el principal fenómeno físico que ocurre en la interacción oleaje-costa y se caracteriza por ser altamente dinámico y caótico con variaciones espacio-temporales. Dicho esto, ha sido sumamente difícil contar con un algoritmo suficientemente robusto para realizar una identificación eficiente y precisa de estos patrones de rotura en imágenes de video y/o radar dentro de la comunidad costera. Sin embargo, el oleaje en rotura es un proceso fácilmente discernible para los ojos humanos que, gracias al eficiente funcionamiento de la corteza cerebral, permite identificar estos patrones bajo diferentes condiciones de oleaje y climáticas. En este trabajo se presenta un novedoso método basado en el uso de redes neuronales convolucionales, las cuales replican el funcionamiento de la corteza visual biológica, para identificar patrones bidimensionales de rotura de forma automática desde imágenes de video.

El presente método solo depende de mediciones en el rango electro-óptico, demostró ser suficientemente robusto bajo diversas condiciones climáticas y de oleaje y es capaz de hacer predicciones de forma computacionalmente eficiente. Esto se puede entender como una mejora a la metodología propuesta por Catalán et al. (2011), que hasta nuestro entendimiento, es la única metodología en la literatura que permite la identificación de patrones de rotura de forma bidimensional. A su vez, este avance permitiría mejorar las metodologías que dependen de los patrones de rotura del oleaje como entrada y así estimar, por ejemplo, cambios en el nivel medio del mar (Flores et al., 2016) o estimaciones de las corrientes medias o corrientes de retorno (Díaz et al., 2018) de forma más eficiente.

Por último, el modelo U-Net presente en este trabajo demostró un destacable rendimiento en la identificación de oleaje en rotura sobre imágenes completamente nuevas en una playa cerca de Las Cruces en Chile, imágenes que fueron obtenidas mediante el uso de dron permitiendo utilizar diferentes formas de adquisición para implementar la metodología propuesta. Esto abre la posibilidad de extender esta línea de investigación y tratar de generalizar este método presente para poder identificar patrones de rotura en diferentes playas en tiempo real. Por esta razón, los algoritmos y conjuntos de datos utilizados en este trabajo se colocan a disposición de la comunidad.

**Palabras Claves:** oleaje rompiente, mediciones electro-ópticas, red neuronal convolucional



# Abstract

Wave breaking is one of the most important mechanisms in nearshore hydrodynamics and morphodynamics, it's a dynamic and chaotic physic phenomena in space and time. Although this process is easily discernible and separable by humans, then an efficient identification of this mechanism through a computational algorithm has been a very difficult task. In this work, we present a novel Convolutional Neural Network based method able to identify 2-dimensional wave breaking patterns. This type of networks attempt to replicate human brain performance.

This method uses only electro-optic range measurements without the need to set pixel intensity threshold and computational efficient. Thus, we were able to solve problems in (Catalán et al., 2011), which to our knowledge it's the only methodology in literature that allows estimating wave breaking patterns. The accurate spatio-temporal identification of wave breaking has been shown to be relevant for estimating wave setup (Flores et al., 2016; Stephens et al., 2011), and mean currents (e.g. Díaz et al., 2018). In other words, if we can improve the wave breaking detection, existing methodologies will improve.

Through an exhaustive validation process, it was possible to reach a CNN model able to obtain waves breaking patterns over different environment and wave conditions, even the model was capable to detection of this process near to the shoreline and this could not be done by the original methodology. Furthermore, by a quantitative and qualitative pixel analysis, it is possible to demonstrate that U-net was able to learn a physical phenomena instead than just identifying high pixel intensity, obtaining a sufficiently robust model.

Finally, U-Net model demonstrated a very good performance on completely new images from a beach near to Las Cruces in Chile, those images was obtain by drone and not a fixed camera system. This opens the possibility to generalized the present method to identify breaking waves in different beach of interest.

**Keywords:** Nearshore wave breaking, optical imaging, convolutional neural networks .



# Índice general

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>Contenidos</b>	<b>VII</b>
<b>Lista de figuras</b>	<b>1</b>
<b>Lista de tablas</b>	<b>3</b>
<b>1. Introducción</b>	<b>5</b>
<b>2. Revisión de Literatura</b>	<b>9</b>
2.1. Proceso de rotura . . . . .	9
2.2. Cuantificación de la disipación de energía . . . . .	10
2.3. Metodología Catalán et al. (2011) . . . . .	11
2.3.1. PDFs y JPDF . . . . .	12
2.3.2. Determinación de umbrales de detección . . . . .	12
2.3.3. Resultados . . . . .	14
2.3.4. Problemas de la metodología . . . . .	14
2.4. Aplicaciones en ingeniería costera . . . . .	15
2.5. Redes Neuronales . . . . .	16
2.6. Resumen . . . . .	17
<b>3. Objetivos</b>	<b>19</b>
3.1. Objetivo General . . . . .	19
3.2. Objetivos Específicos . . . . .	19
3.3. Hipótesis . . . . .	19

<b>4. Metodología</b>	<b>21</b>
4.1. Datos . . . . .	22
4.2. Red Neuronal Convolutacional . . . . .	24
4.3. Conjuntos de entrenamiento, validación y prueba . . . . .	25
4.4. Arquitectura de Red Neuronal . . . . .	25
4.4.1. Convolución . . . . .	25
4.4.2. Función de Activación . . . . .	26
4.4.3. Subsampling con Max-Pooling . . . . .	26
4.5. Entrenamiento y Validación de la red . . . . .	27
<b>5. Resultados</b>	<b>29</b>
<b>6. Discusión</b>	<b>33</b>
6.1. Análisis de píxeles de máscaras de rotura . . . . .	33
6.1.1. Análisis cualitativo . . . . .	33
6.1.2. Análisis cuantitativo . . . . .	35
6.2. Aplicación en nuevo dominio . . . . .	37
<b>7. Conclusión</b>	<b>39</b>
<b>8. Trabajo Futuro</b>	<b>41</b>
<b>A. Implementación del modelo U-Net</b>	<b>43</b>
A.1. Entrenar el modelo . . . . .	43
A.1.1. Crear y cargar conjuntos de entrenamiento, validación y prueba. . . . .	43
A.1.2. Entrenamiento de la red . . . . .	44
A.2. Predicción sobre nuevas imágenes . . . . .	45
A.2.1. Creación de matrices de predicción . . . . .	45
A.2.2. Cargar modelo y hacer predicción . . . . .	46
A.3. Graficar los resultados . . . . .	46

# Índice de figuras

2.1. Ejemplo de modelo ola-roller propuesto por Duncan (1981) . . . . .	10
2.2. Posición y ejemplo de las plataformas de teledección utilizadas. a) Corresponde a la posición geográfica de los instrumentos, b) y c) son fotografías del radar y sistema de cámaras utilizado . . . . .	11
2.3. Ejemplo de resultados obtenidos por la metododología propuesta por Catalán et al. (2011), donde a) y b) representan las PDF del radar y del sistema de cámaras respectivamente, c) representa la JPDF y d) corresponde a un ejemplo de las mediciones obtenidas con radar y e) las obtenidas con el sistema de cámaras. Fuente: Díaz et al. (2018) . . . . .	12
2.4. Ejemplo de las regiones identificados mediante la metodología propuesta por Catalán et al. (2011) sobre el gráfico de JPDF. . . . .	13
2.5. Ejemplo de imágenes obtenidas con radar y video utilizadas como input en la metodología propuesta de Catalán et al. (2011) y su respectivo resultado. . . . .	14
4.1. Ejemplo de olas en rotura en diferentes configuraciones de imágenes e diferentes condiciones hidrodinámicas. . . . .	21
4.2. Muestra de las condiciones de oleaje presente durante los experimentos segun mediciones en el FRF y un ADCP. (a) Altura de ola significativa, (b) Espectro de frecuencia (fondo en escala de grises) y frecuencias peak (línea sólida), (c) Espectro direccional y (d) Elevación de la marea. Las líneas rojas y azules representan la muestras los conjuntos de imágenes seleccionadas. Por último, se muestra un ejemplo de cada conjunto de imágenes utilizadas. . . . .	23
4.3. Esquema de la arquitectura de la red neuronal convolucional U-Net. Fuente: elaboración propia . . . . .	24
4.4. Ejemplos de operaciones matemáticas utilizadas por la red neuronal convolucional U-Net. (a) representa el proceso de convolución, donde el hiperpaámetro número de filtros o <i>number of filter</i> es equivale al número de <i>features maps</i> y (b) corresponde a la operación Max-Pooling. . . . .	26
5.1. Evolución de la función de pérdida BCE y la métrica de precisión $F1 - Score$ . El desempeño de la combinación 3 durante el entrenamiento se muestra a través de la línea azul y durante la validación a través de la línea naranja. Las líneas grises corresponden a las demas combinaciones, donde con línea continua se representa el proceso de entrenamiento y en línea discontinua la validación. . . . .	30

---

6.1.	Ejemplo A resultados obtenidos con ambos métodos. Frame A corresponde a la imagen 348 del día 9 de Septiembre a las 11:00 (hora local). Los gráficos del medio corresponde a las máscaras de roturas obtenidas con la metodología propuesta por Catalán et al. (2011) (A.1) y por U-Net (A.2) y los gráficos de la derecha muestran la respectiva máscara de rotura sobre la imagen A. . . . .	34
6.2.	Ejemplo B resultados obtenidos con ambos métodos. Frame A corresponde a la imagen 348 del día 9 de Septiembre a las 15:00 (hora local). Los gráficos del medio corresponde a las máscaras de roturas obtenidas con ambas métodos y los gráficos de la derecha muestran la respectiva máscara de rotura sobre la imagen B. . . . .	35
6.3.	Estadística de la detección de oleaje rompiente en todas las imágenes del conjunto de datos. (a) Imagen de video de referencia; mapas de porcentajes de (b) pixeles verdaderos; (c) pixeles errados; (d) pixeles falsos; (e) Histograma que representa la evolución de la distribución de pixeles verdaderos (azul), errados (naranja) y falsos (naranja claro) sobre diferentes condiciones lumínicas y de oleaje. . . . .	36
6.4.	Ejemplo de aplicación del modelo U-Net sobre imágenes capturadas mediante Dron en las cercanías de Las Cruces, Chile. A.1 y B.1 corresponden a dos imágenes obtenidas mediante Dron, A.2 y B.2 son las máscaras de roturas obtenidas por U-Net. Finalmente, la tercera columna es la superposición de las máscaras con las imágenes. . . . .	37



# Índice de tablas

4.1. Espacio de hiperparámetros utilizados. Los valores en negrita representan la mejor combinación. . . . .	28
4.2. Clasificación del desempeño de las combinaciones de hiperparámetros. Se denota en negrita las 5 mejores configuraciones las que están clasificados su mejor valor de precisión y pérdida. . . . .	28
5.1. Cinco mejores combinaciones de hiperparámetros y sus respectivas métricas en los conjuntos de entrenamiento y validación. . . . .	29
5.2. Porcentajes de la cantidad de pixeles identificados por cada combinación de hiperparámetros candidatos. $\sigma$ denota la desviación estándar. . . . .	30



# Capítulo 1

## Introducción

El oleaje es un fenómeno altamente dinámico que tiene gran impacto para la sociedad dada su interacción con la costa. Esta importancia llama la atención desde diferentes ámbitos. Desde el mundo científico, gran parte de la energía del oleaje es disipada cuando llega a la costa, lo que genera diversas alteraciones en el comportamiento de la interacción oleaje-costa, como la generación de corrientes, transporte de sedimentos, entre otros. Desde una mirada como sociedad, numerosas actividades se llevan a cabo en la costa, desde actividades económicas hasta recreacionales, las que se pueden poner en riesgo producto del oleaje.

El principal fenómeno físico que ocurre en las cercanías de la costa es la disipación de energía producto de la rotura del oleaje. De hecho, Holman and Haller (2013) argumentaron que la medición de la disipación de energía del oleaje es análogo a medir los forzantes de los flujos medios en la zona de interés. Así podemos comprender que el estudio de este fenómeno es fundamental para entender de mejor manera los diferentes procesos costeros.

La disipación de energía del oleaje involucra diversos procesos físicos. A medida que una ola se acerca a la costa, producto de la disminución de la profundidad, la altura de la ola aumenta hasta llegar al asomeramiento. Producto de esto, la ola se ve sometida a un proceso de inestabilidad, lo que marca el inicio del fenómeno de rotura. Cuando la ola rompe se produce la separación del flujo, de lo cual se forman dos cuerpos de aguas distintos: la ola en sí misma y un segundo cuerpo turbulento compuesto por una mezcla de agua-aire denominado roller, lo que tiene como consecuencia la disipación de energía. Este cuerpo puede ser un factor dominante en el balance de masa de la zona rompiente y en la generación de corrientes perpendicular a la costa (Svendsen , 2006).

Durante el proceso de rotura, existe una transferencia de momentum entre la ola y el roller lo que provoca cambios del nivel medio del mar, patrones de circulación en la costa, corrientes longitudinales, perpendiculares a la propagación del oleaje y las corrientes de retorno. Este último tipo de corrientes son conocidas por significar una amenaza para la vida de los bañistas, dado que han generado la muerte de miles de personas en diversas partes del mundo (Brighton et al. (2019) y Brewster et al. (2019)). De lo anterior se desprende la importancia de cuantificar la disipación de energía del oleaje rompiente, la cual se ha modelado usando la geometría del roller (Duncan, 1981). Cuantificar esta energía presenta un gran desafío ya que el proceso de rotura del oleaje es un fenómeno altamente caótico que depende de la batimetría del lugar, condiciones del oleaje y ambientales, la marea, etc. Lo anterior provoca que la geometría del roller presente una alta variabilidad espacial y temporalmente.

Una de las técnicas de medición más utilizadas en el área de ingeniería costera son las mediciones remotas mediante cámaras de video y/o radar (Catalán et al., 2011, 2014). Colocando especial atención en las mediciones de video, estas se pueden asimilar a lo que ve una persona, es decir, bajo ciertas condiciones, las mediciones de video se pueden considerar como un retrato de la visión del ojo humano. Al ver una imagen de una ola rompiente, el ojo humano tiene la habilidad de concentrar su atención en los píxeles de gran intensidad que representan el roller. Además, apoyándose en el contraste que generan estos píxeles con el resto de la imagen, se pueden identificar los patrones de oleaje rompiente permitiendo a la persona detectar nuevos patrones al ver nuevas imágenes de este fenómeno. Hacer este mismo proceso de forma computacional implica una gran complejidad debido a que el algoritmo se suele confundir fácilmente, por ejemplo, entre la alta intensidad de píxel de olas rompientes y de la espuma remanente producto del mismo proceso.

Durante los últimos años, el uso de técnicas machine y Deep Learning han generado un significativo interés logrando diferentes aplicaciones en la comunidad costera, donde la identificación del oleaje rompiente y otros fenómenos ha sido el principal foco de interés. En esto, se han utilizado dos enfoques principalmente. El primero, trata de identificar características del oleaje que podrían ser usadas para discriminar entre los diferentes estados del proceso de rotura. Por ejemplo, Stringari and Power (2019) usa una red neuronal relativamente simple validada en Stringari et al. (2019) que permite identificar olas en rotura mediante el seguimiento de píxeles rompientes mediante imágenes en el rango electro-óptico. Dicha metodología entrega un conjunto de datos que fueron utilizados para calcular la fracción de oleaje rompiente. En el análisis de este trabajo, la atención fue puesta en identificar la ocurrencia de la rotura, pero sin cuantificar otras propiedades como su geometría.

Stringari et al. (2021) expandió el problema de clasificación para detectar el área de olas en roturas producto de límites de inclinación en aguas profundas. Para hacer esto usaron umbrales de intensidad logrando identificar manualmente las áreas correspondientes a rotura activa y discriminarla de la espuma remanente. Este proceso puede haber sido algo arbitrario dado que se realizó basándose en la información de la intensidad de los píxeles y no necesariamente sobre un proceso físico. Esta idea no es fácil de aplicar a mediciones en las cercanías de la costa dado que la rotura del oleaje en esta zona puede llegar a producir grandes cantidades de espuma remanente.

Otro enfoque común es el uso de características que pueden ser encontradas en las imágenes. Por ejemplo, Buscombe et al. (2020) correlacionó estas características con series de tiempo de la altura de la ola para lograr entrenar de forma exitosa una red neuronal que pueda predecir la altura de la ola y su período. Por otro lado, Buscombe and Carini (2019) usaron imágenes infrarrojas para entrenar y clasificar los diferentes tipo de rompientes presentes en las mediciones. En este enfoque, no es necesario colocar atención en otros aspectos de la rotura como su geometría y su posición, solo corresponde a un problema de clasificación multiclases de la imagen en su totalidad.

Como se puede ver, han habido diversas aplicaciones exitosas de machine y Deep Learning en el estudio de las olas en rotura, pero uno de los aspectos más relevantes de este fenómeno no ha sido complementamente desarrollada como lo es la cuantificación ola a ola, y no sólo su ocurrencia. Se ha demostrado que parámetros como su longitud transversal o el área del roller han sido fundamentales para estimar las tasas de disipación individual basado en el modelo de roller de Duncan (1981), con buenos resultados en laboratorio para medir su propagación (Haller and Catalán, 2009; Flores et al., 2016) y en su medición bidimensional (Díaz-Méndez et al., 2015; Carini et al., 2015; Díaz et al., 2018).

En esta tesis proponemos la implementación de una red neuronal convolucional que permite la identificación de la rotura del oleaje ola a ola y su desarrollo hacia la costa. Se utiliza el algoritmo propuesto por Catalán et al. (2011) para obtener las máscaras de roturas que se usan como datos

de referencia para el entrenamiento. De la misma forma que en de Silva et al. (2021), este proceso se puede considerar arbitrario dado que la metodología de Catalán et al. (2011) también es dependiente de la selección de un umbral de intensidad, pero a su vez depende de dos plataformas de teledecepciones que utilizan diferentes mecanismos físicos lo que nos permite considerar los datos lo suficientemente robustos para este objetivo. Dicho esto, la red neuronal convolucional tiene por objetivo extraer las características más relevantes de las imágenes correspondientes al roller y identificar su geometría y posición dentro de la imagen. Además, el método propuesto permite reducir los tiempos de cálculos en la segmentación de imágenes, siendo posible la predicción en tiempo real. Esto último tiene un gran impacto desde un punto de vista científico, mejorando la aplicación de todas las metodologías que dependen de estos patrones y desde una mirada social, esto permitiría salvar vida de bañistas por las razones ya expuestas.



## Capítulo 2

# Revisión de Literatura

### 2.1. Proceso de rotura

El comportamiento de las olas, en las cercanías a la costa, depende de la profundidad de ésta. Cuando se encuentran en aguas profundas o alejadas de la costa, la ola no se ve afectada por la profundidad. A medida que estas se propagan hacia la costa, la profundidad comienza a disminuir, lo que genera que la ola disminuya su celeridad y aumente su altura. En términos de energía, lo anterior corresponde a un balance de energía donde existe una disminución de la energía cinética y aumento de la energía potencial, respectivamente.

La altura de la ola aumenta hasta llegar a un cierto umbral, donde se ven sometidas a un proceso de inestabilidad. Este proceso se ha modelado utilizando dos cuerpos de aguas distintos, la ola propiamente tal y el roller, disipando energía en forma de ruido, calor y turbulencia entre agua y aire. El roller se define como un cuerpo turbulento de aire y agua, que es altamente caótico, que se desarrolla y se propaga con el frente de la ola a su misma velocidad. Svendsen (2006) plantea que el roller es un factor dominante en los procesos de costeros, debido a que altera directamente el balance de masa y de momentum.

Por ende, durante el proceso de rotura existe una transferencia de masa desde la ola hacia el roller, de lo cual se concluye que, si se transfiere masa entonces se transfiere momentum. Esta transferencia se lleva a cabo a través de una capa de corte en la interfaz entre la ola y el roller (Figura 2.1), donde este último extrae energía a la ola aumentando su masa y tratando de alcanzar un constante equilibrio a medida que se propaga con esta. Producto de lo anterior, se genera una alteración en el flujo de momentum, lo que implica un cambio en el gradiente del tensor de radiación. Esto se puede traducir como la generación de fuerzas de gran magnitud que actúan sobre el sistema dando origen a corrientes, cambio del nivel medio del mar y patrones de circulación en la costa (Svendsen , 2006).

Uno de los principales efectos de la transferencia de momentum son las corrientes, las que pueden ser en dirección longitudinal y/o perpendicular a la propagación del oleaje o corrientes rips (Svendsen , 2006). Estas corrientes pueden alcanzar velocidades de hasta uno o dos metros por segundo lo que explica, por ejemplo, el traslado de sedimentos de la costa en diferentes direcciones. Además, los rips son altamente peligrosos dado que arrastran a los bañistas mar adentro a gran velocidad y en muchos casos ha provocado la muerte de personas (Brighton et al. (2019) y Brewster et al. (2019)).

## 2.2. Cuantificación de la disipación de energía

Reconociendo la importancia de cuantificar la disipación de energía producto del oleaje rompiente y considerando la fuerte relación entre el roller y dicha disipación (Duncan, 1981; Haller and Catalán, 2009), a lo largo de la historia de la ingeniería costera se han hecho varios esfuerzos por cuantificar este fenómeno. Se han propuesto diversos enfoques para determinar cómo y cuándo se produce la rotura del oleaje. Dentro de los diferentes enfoques, se destaca la aplicación de métodos estadísticos para calcular la probabilidad de que una ola este rompiendo, donde toma relevancia la fracción de oleaje rompiente (Stringari and Power, 2019). Por otro lado, se encuentra un enfoque instantáneo fundado en un criterio de rotura basado en cada ola (Duncan, 1981; Díaz et al., 2018).

Tomando en cuenta el último enfoque, Duncan (1981) propuso un modelo de disipación de energía producto de las olas rompientes, que considera el ya mencionado balance entre el esfuerzo de corte en la interfaz entre roller y la ola y la componente tangencial del peso de este cuerpo de agua (ecuación 2.1). El modelo permite relacionar parámetros geométricos del roller de cada ola con la disipación producto del fenómeno de rotura. Si bien este método permite un gran avance hacia la cuantificación de la disipación de energía del oleaje, es relevante destacar que el proceso de rotura está estrechamente relacionado con la batimetría de la costa, por ende, las olas rompientes exhiben una significativa variabilidad espacial y temporal que es difícil de observar y de predecir con exactitud

$$\rho' g A \sin \theta = \tau L_r. \quad (2.1)$$

El método mencionado está representado en la ecuación 2.1, donde  $\rho'$  corresponde a la densidad del roller,  $A$  es el área del *roller*,  $\theta$  es el ángulo que describe la pendiente de la cara de la ola rompiente,  $\tau$  es el esfuerzo de corte y  $L_r$  es el largo de la superficie del roller. Esto se puede ver de forma gráfica en la Figura 2.1.

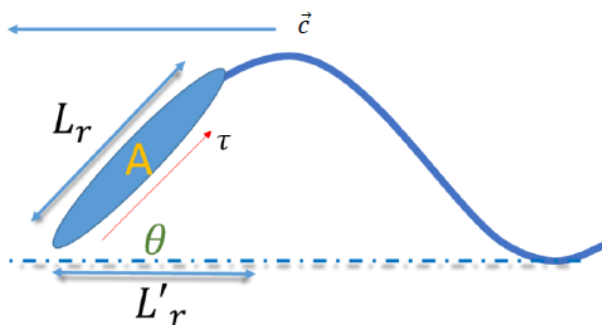


Figura 2.1: Ejemplo de modelo ola-roller propuesto por Duncan (1981)

Para validar los enfoques mencionados, se han implementado diversas técnicas de medición con el fin de cuantificar diferentes parámetros del oleaje, dentro de los cuales se destacan las mediciones in-situ y remotas. Este último tipo de mediciones corresponden a imágenes de video y/o radar que han posibilitado un gran avance en esta área en las últimas décadas, debido a que permite realizar mediciones por tiempo prolongado y a un muy bajo costo. Las ventajas de las mediciones remotas descritas permitieron a Haller and Catalán (2009) medir la longitud del roller de olas en un experimento de laboratorio, siendo este parámetro geométrico fundamental a la hora de calcular la disipación de energía mediante el método de Duncan. Otra aplicación fue propuesta por Flores

et al. (2016), quienes utilizaron mediciones de la longitud del roller y el método de Duncan para obtener un modelo de predicción de propiedades hidrodinámicas basa en los esfuerzos de radiación.

A pesar de los grandes avances obtenidos en los trabajos mencionados, las mediciones remotas presentan un problema en la identificación del proceso de rotura debido a las intensidades de retorno del video y radar y la forma de funcionar de cada sensor. El video tiene problemas en diferenciar las olas rompientes y la espuma residual que se genera producto del mismo proceso, ya que ambas poseen alta intensidad de retorno. Por su parte, el radar presenta problemas entre olas empinadas y olas rompientes, dado que generan intensidades similares.

### 2.3. Metodología Catalán et al. (2011)

Con el fin de abordar el problema de segmentación del proceso de rotura, Catalán et al. (2011) propone una metodología que combina ambas mediciones remotas. Para esto se crean funciones de densidad de probabilidad conjuntas que agrupan las intensidades de pixeles de las imágenes en el tiempo y así, se puede determinar cuándo un pixel corresponde a una ola en rotura. Ejemplo de los instrumentos utilizados en la metodología y su ubicación se puede ver en la Figura 2.2, particularmente, en la Figura 2.2.b se presenta el radar y en la Figura 2.2.c el sistema de cámaras de video. Esta metodología utiliza datos obtenidos en las instalaciones del cuerpo de ingenieros del ejército de los Estados Unidos, ubicado en Duck, en Carolina del Norte. El eje  $x$  corresponde al eje perpendicular a la costa y el eje  $y$  apunta  $18^\circ$  grados al oeste del norte geográfico como se puede ver en la Figura 2.2.a.



Figura 2.2: Posición y ejemplo de las plataformas de teledeceión utilizadas. a) Corresponde a la posición geográfica de los intrumentos, b) y c) son fotografías del radar y sistema de cámaras utilizado

Dado que el método depende de dos sensores remotos que toman muestras de forma independiente, es necesario hacer un preprocesamiento, calculando la correlación que existe entre ambas imágenes para verificar que corresponden al mismo instante. Específicamente, para calcular la correlación entre las series de tiempos de cada sensor, se utiliza la coherencia al cuadrado en cada píxel de las imágenes de cada sensor (Bendat y Piersol, 1986). Este valor de coherencia está en el rango entre 0 y 1, tomando el valor de 1 cuando existe una alta correlación. Mayor detalle sobre esta explicación se puede ver en el trabajo presentado por Catalán et al. (2011).

### 2.3.1. PDFs y JPDF

Como se mencionó, el método cuestión utiliza funciones de densidad de probabilidad para cada sensor por separado (PDFs) y funciones de densidad de probabilidad conjuntas (JPDF). La idea de trabajar con estas funciones es calcular la frecuencia de diferentes intensidades de pixeles y luego se suman para toda la imagen, ya sea para cada sensor por separado o en forma conjunta según corresponda. Las funciones de densidad resultantes se pueden ver en la Figura 2.3.

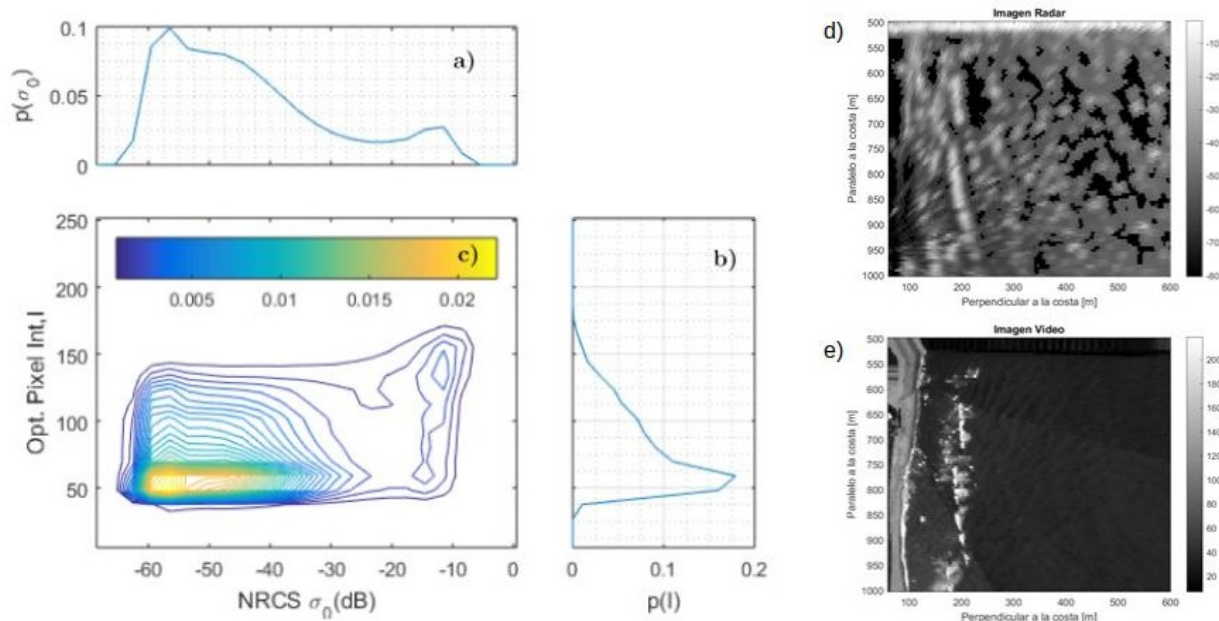


Figura 2.3: Ejemplo de resultados obtenidos por la metodolog a propuesta por Catal n et al. (2011), donde a) y b) representan las PDF del radar y del sistema de c maras respectivamente, c) representa la JPDF y d) corresponde a un ejemplo de las mediciones obtenidas con radar y e) las obtenidas con el sistema de c maras. Fuente: D az et al. (2018)

De la misma Figura, adem s se puede destacar que los pixeles que tienen mayor frecuencia corresponden a los de menor intensidad tanto para el video como para el radar. Esto se puede explicar revisando las im genes del dataset 2.3.d) y 2.3.e) dado que gran parte de la imagen corresponde a olas que no est n rompiendo. De la misma forma, los valores de intensidad m s altos en ambos sensores visualmente corresponden a las olas sometidas al proceso de rotura.

### 2.3.2. Determinaci n de umbrales de detecci n

Uno de los pasos claves de esta metodolog a corresponde a la determinaci n de umbrales de detecci n, es decir, encontrar el valor l mite de intensidad para considerar si el p xel corresponde a una ola en rotura o no. Por ejemplo, debido a que tanto las olas en rotura como la espuma residual est n compuestos por pixeles de alta intensidad, una correcta determinaci n de umbrales para ambos sensores remotos permiten diferenciar estos casos. La detecci n de umbrales se determina de forma autom tica utilizando diferencias discretas de primer y segundo orden para las funciones de densidad de probabilidad (PDF). Mayor detalle sobre el c lculo de los umbrales se puede encontrar en el trabajo original.

Una vez que se cuenta con los umbrales de intensidad para ambos sensores remotos se pueden diferenciar cuatro zonas correspondientes a las cuatro etapas a las que se somete una ola desde aguas profundas hasta que llega a la costa. Cabe destacar que  $I(x, y, t)$  corresponde al umbral seleccionado para las imágenes de video y  $\sigma(x, y, t)$  es el umbral para el radar.

Región de no rompimiento (1):

$$I(x, y, t) < I^t \wedge \sigma(x, y, t) < \sigma_0^t \quad (2.2)$$

Región de rompimiento (2):

$$I(x, y, t) \geq I^t \wedge \sigma(x, y, t) \geq \sigma_0^t \quad (2.3)$$

Región de espuma remanente (3):

$$I(x, y, t) \geq I^t \wedge \sigma(x, y, t) < \sigma_0^t \quad (2.4)$$

Región de olas empinadas (4):

$$I(x, y, t) < I^t \wedge \sigma(x, y, t) \geq \sigma_0^t \quad (2.5)$$

Estos resultados se pueden ver de forma gráfica en la Figura 2.4, donde se corrobora lo mencionado en la sección 2.3.1, la región (1) correspondiente a la región de no rompimiento es la que tiene mayor frecuencia. Por otro lado, colocando atención en la región (2) y (3), se aprecia que los valores de intensidad altos para video y para radar corresponde a las olas en rotura, mientras que la espuma remanente se caracteriza por presentar baja intensidad de píxel en el radar y alta intensidad para el video.

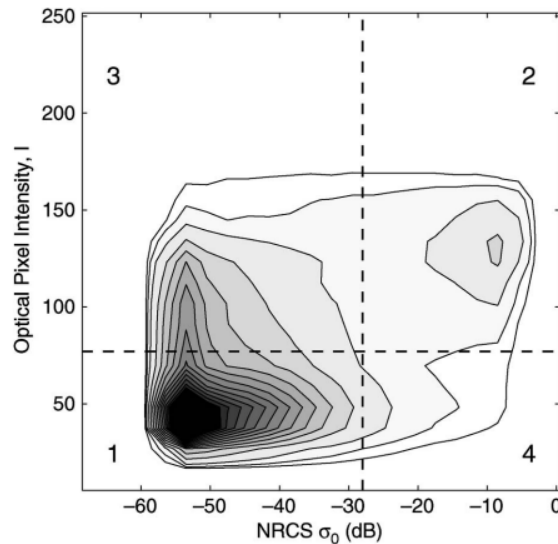


Figura 2.4: Ejemplo de las regiones identificadas mediante la metodología propuesta por Catalán et al. (2011) sobre el gráfico de JPDP.

### 2.3.3. Resultados

Ahora que se cuenta con umbrales de detección, se clasifican las imágenes píxel a píxel. Por cada imagen se obtiene una matriz binaria donde cada entrada de la matriz representa si el píxel corresponde a una ola en rotura o no. Si la intensidad del píxel en la imagen de video y de radar cumple la condición presentada en 2.2, el algoritmo retorna 1 y retorna 0 en si corresponde a otro caso. Es importante recalcar que este proceso de clasificación puede ser visto como un proceso de segmentación de patrones de rotura del oleaje.

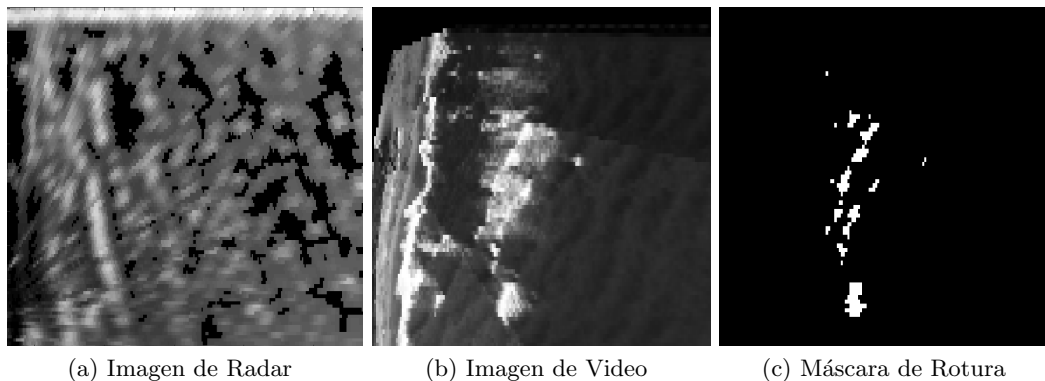


Figura 2.5: Ejemplo de imágenes obtenidas con radar y video utilizadas como input en la metodología propuesta de Catalán et al. (2011) y su respectivo resultado.

### 2.3.4. Problemas de la metodología

Esta metodología permite obtener resultados bastante robustos en la identificación de patrones de rotura, más aún, considerando lo difícil que es lograr la identificación de este fenómeno computacionalmente. Sin embargo, presenta problemas a la hora implementarla, al tratar de generalizar el método a otras playas y en su tiempo de ejecución.

Uno de los principales problemas que se presenta en Catalán et al. (2011) corresponde a la dependencia de dos sensores remotos lo que limita en gran medida el uso del método, ya que no es común la presencia de radares en las playas. Por otro lado, el umbral de detección presenta una alta sensibilidad al conjunto de imágenes de entrada, es decir, se deben ajustar los umbrales de intensidad para cada condición de oleaje o ambiental distinta. Si bien se cuenta con un algoritmo para poder calcular los umbrales de detección de forma automática, éste no siempre entrega resultados confiables, siendo necesario ajustar los umbrales de forma manual.

El método original depende del procesamiento de imágenes de dimensiones superior a los  $512 \times 512$  píxeles. Esto hace que la implementación del método conlleve un gran costo computacional al tener que cargar el conjunto de imágenes, calcular las PDFs y JPFD, determinar el umbral y finalmente obtener las mencionadas máscaras de rotura. Por lo que el tiempo que demora en hacer una predicción imposibilita una predicción en tiempo real.

Por último, se destaca que, si se acopla esta metodología con la propuesta por Duncan, se puede obtener un método para la cuantificación bidimensional de disipación de energía en la zona rompiente (Díaz et al., 2018). Para lograr esto se debe extraer propiedades geométricas del roller desde las máscaras de rotura e incorporarlos en el modelo de Duncan (1981), obteniendo campos

bidimensionales de disipación como se plantea en el trabajo de Díaz et al. (2018). Es importante mencionar que este último método generó una buena base teórica para la estimación de la disipación de energía utilizando un enfoque ola a ola.

## 2.4. Aplicaciones en ingeniería costera

Con el propósito de solucionar las limitaciones planteadas en la sección 2.3.4, es de interés buscar una nueva herramienta computacional que permita detectar de forma precisa y eficientemente los patrones de rotura del oleaje. Para esto, se pretende utilizar mediciones remotas por las razones que fueron expuestas y debido a la dificultad de contar con radares en los lugares de estudio, es de interés que el modelo de detección dependa de un solo sensor a diferencia de la metodología propuesta por Catalán et al. (2011). De lo anterior se concluye que, si se genera un modelo suficientemente robusto abrirá la opción a utilizar incluso mediciones obtenidas con dron, siendo mucho más accesible a la comunidad.

En este sentido, en las últimas décadas, la Inteligencia Artificial ha sido una herramienta utilizada en gran medida para optimizar y automatizar diversos procesos en diferentes áreas. Específicamente en ingeniería costera, se ha demostrado que incorporando las mediciones remotas o in situ como entrada de dichos modelos permite resolver problemas de clasificación (Buscombe and Carini, 2019) y/o regresión (Buscombe et al., 2020)

Dentro de los algoritmos de Inteligencia Artificial se destacan las redes neuronales, que han sido modelos computacionales altamente usados en el área de procesos costeros (Buscombe and Carini (2019), Stringari et al. (2019)). Este tipo de modelo, en un principio, fue basado en replicar el funcionamiento de las redes neuronales del cerebro humano, donde cada neurona tiene una tarea sencilla y específica, pero están conectadas entre sí con las demás neuronas con el fin de realizar tareas más complejas. Es decir, dado ciertos parámetros de entradas, como píxel de imágenes o cantidades estadísticas, el modelo encuentra la forma de combinarlos con tal de predecir una determinada salida o resultado. Por esta razón, es de interés indagar entre las diferentes arquitecturas de redes neuronales para encontrar una solución a las limitaciones planteadas en la sección 2.3.4.

Una aplicación concreta en ingeniería costera es el uso de una red neuronal capaz de predecir la rotura de la ola y cuantificar la fracción de olas rompientes teniendo como entrada cantidades estadísticas como la altura de la ola, periodo de ola, número de olas, etc. Stringari et al. (2019). El parámetro de fracción de olas rompientes es relevante en el estudio de la disipación de energía si se utiliza el enfoque estadístico.

Dentro del área de la Inteligencia Artificial se encuentra un subcampo denominado Visión Artificial. Este subcampo busca automatizar tareas que imitan las capacidades visuales humanas utilizando las ya mencionadas redes neuronales. En otras palabras, la tarea es extraer toda la información contenida en imágenes. Por ejemplo, gracias a esta área es posible predecir el tipo de ola (no rompiente, spilling, plunging) a partir de mediciones remotas Buscombe and Carini (2019).

Dependiendo de la tarea que se desea realizar, se debe elegir una determinada arquitectura de la red neuronal que permita obtener de forma más eficiente la información contenida en los datos de entrada. Por ejemplo, considerando la abundante cantidad de imágenes que se obtiene mediante mediciones remotas, es posible predecir diferentes parámetros del oleaje como la altura y periodo de las olas (Buscombe et al., 2020). O también, se puede utilizar una red neuronal para detectar olas rompientes y realizar un seguimiento a estas utilizando imágenes promediadas en el tiempo

(Stringari et al., 2019). En esta última investigación, el dataset utilizando no permite detectar la rotura del oleaje en forma bidimensional por lo que no se considera como una solución al problema propuesto en esta tesis.

Los trabajos que se han expuesto corresponden a problemas de clasificación y regresión, siendo estos los principales tipos de problemas que se busca resolver mediante técnicas de Inteligencia Artificial. Sin embargo, en los últimos años, en el campo de Visión Artificial se ha masificado el uso de redes neuronales para resolver problemas de segmentación de imágenes incluso en ingeniería costera (den Bieman et al., 2020). Por segmentación de imágenes se entiende el proceso de agrupar píxeles que cumplan con una determinada característica mejorando la representación de la imagen a una más fácil de analizar. Más precisamente, la segmentación se utiliza tanto para localizar objetos dentro de una imagen como para encontrar los límites de estos objetos dentro de la misma. También se puede ver como un problema de clasificación de píxeles de una imagen conservando la localización de cada píxel.

## 2.5. Redes Neuronales

En primer lugar, es esencial entender bien el problema que se desea resolver. Como se mencionó en las secciones anteriores y basándose en la metodología propuesta por Catalán et al. (2011), el problema consiste en identificar patrones de rotura en forma bidimensional. Específicamente, a partir de imágenes obtenidas con cámaras de video se desea segmentar cada imagen identificando cuándo y dónde ocurre la rotura del oleaje. Dicho esto, se puede considerar que el método de Catalán se traduce en un problema de clasificación considerando localización.

En el mundo de la Inteligencia Artificial se ha trabajado el problema de clasificación a partir de imágenes de mediante diferentes enfoques, siendo el uso de redes neuronales convolucionales (CNN) el más utilizado. Por definición, una red neuronal convolucional es un tipo de red neuronal artificial que procesa sus capas imitando la corteza visual del cerebro humano para identificar las principales características de las imágenes Lecun et al. (1998). Las primeras redes neuronales convolucionales aparecieron en 1998 con el trabajo hecho por Lecun et al. (1998) en un problema de clasificación de imágenes donde la etiqueta de cada imagen correspondía a una determinada clase. Este tipo de red, dado el año en que fueron desarrolladas, presentaban problemas en su implementación por la gran cantidad de imágenes que se necesitaban al entrenar el modelo y límites computacionales. A partir del 2006, se desarrollaron varios métodos gracias a los avances en el estado del arte y al poder de computación, permitiendo avanzar en el desarrollo de estas técnicas. En este contexto, Alexnet es una red neuronal convolucional clásica propuesta por Krizhevsky et al. (2012) que mejoraba considerablemente el desempeño en la tarea de clasificación de imágenes respecto a los métodos anteriores. Debido al éxito de Alexnet, se propusieron varias arquitecturas de CNN que mejoraban el desempeño, dentro de las cuales se destacan VGGNet (Simonyan and Zisserman, 2015) o GoogleNet (Szegedy et al., 2017).

Los métodos mencionados permiten hacer una clasificación de imágenes, pero categorizando la imagen completa como una sola clase. Avances posteriores y la necesidad de contar con redes neuronales más particulares, permitieron hacer clasificaciones más específicas de las imágenes donde se pueda identificar los objetos dentro de una imagen e incluso segmentar los contornos de estos objetos. Considerando que el problema que se quiere resolver es un problema de segmentación y, tomando en cuenta los avances en las diferentes arquitecturas de CNN, aparece una red neuronal convolucional denominada U-Net (Ronneberger et al., 2015). U-Net corresponde a una red neu-

ronal que inicialmente fue implementada para resolver problemas de segmentación de imágenes biomédicas, teniendo un gran éxito gracias a su alta eficacia en este tipo de problemas. Dentro de las propiedades de esta red se destaca que, por su especial arquitectura, es eficiente computacionalmente bajo ciertas condiciones y puede identificar especialmente los bordes del fenómeno de interés que se desea segmentar dentro de la imagen.

Es necesario hacer hincapié en que la implementación de técnicas de IA se basa en tener un buen conjunto de datos para entrenar el modelo y un correcto procesamiento de este para así obtener buenos resultados. En este caso, para las mediciones remotas, los datos está fuertemente relacionada a las condiciones ambientales de la zona de estudio y la calidad de los instrumentos para realizar las mediciones. Esto se ve reflejado en las máscaras que propone Catalán et al. (2011) dado que no todas las olas rompientes son detectas y se espera que la red pueda solucionar este problema aumentando la eficiencia de la detección.

Por último, es importante recalcar que este avance permitiría mejorar en gran medida la identificación de patrones de rotura del oleaje de forma eficiente e incluso, se podría llegar a realizar dicha detección en tiempo real bajo ciertas condiciones computacional y ambientales. Una aplicación de lo anterior corresponde a, dada la relación entre la rotura del oleaje y las corrientes de retorno, poder implementar la detección en tiempo real de la rotura para ver la existencia de dichas corrientes en una determinada playa. Esto permitiría tomar ciertas medidas, por ejemplo, el cierre temporal de cierto sector de la playa con el fin de salvar la vidas de los bañistas.

## 2.6. Resumen

En los últimos 15 años el uso de nuevas herramientas como modelos de inteligencia artificial ha crecido de manera exponencial y el éxito de estos modelos fueron presentados en esta sección. Se tratan temas como la importancia de utilizar mediciones remotas para enfrentar problemas en el área costera, por ejemplo, identificar patrones de rotura de forma bidimensional gracias a que estas técnicas de mediciones cuentan con un bajo costo en mediciones de larga duración.

Se mencionan las técnicas o metodologías existentes en la comunidad costera para abordar el problema de la rotura, donde se destaca el cálculo de la fracción de oleaje rompiente (Stringari and Power, 2019) o la predicción de la rotura del oleaje y su posterior seguimiento permitiendo estimar propiedades como el período del oleaje (Stringari et al., 2019). Si bien todas estos métodos son robustos y permiten predecir propiedades importantes, no permiten tener una estimación bidimensional de los patrones de rotura debido a los tipos de redes neuronales que han utilizado.

Considerando las problemáticas que presenta la metodología de Catalán et al. (2011) y la importancia que tiene la identificación y cuantificación de los patrones de rotura, se expone una oportunidad de investigación utilizando herramientas como las redes neuronales convoluciones para resolver este problema de segmentación. Dicho esto y considerando las herramientas de IA existentes, se propone utilizar una red neuronal convolucional denominada U-Net para resolver el problema en cuestión. Esto permitiría la posibilidad de realizar una predicción mucho más robusta, de forma automática, con un bajo costo computacional e incluso realizar predicciones en tiempo real bajo ciertas condiciones.



## Capítulo 3

# Objetivos

Si bien la metodología propuesta por Catalán et al. (2011) tiene resultados bastante robustos, permitiendo a Flores et al. (2016) estimar cambios en el nivel medio del mar y a Díaz et al. (2018) estimar corrientes medias, esta cuenta con los problemas mencionados anteriormente limitando su aplicación. De lograr una detección eficiente y precisa de la rotura, permitiría mejorar nuestro conocimiento acerca de este fenómeno y sus consecuencias.

### 3.1. Objetivo General

- Identificar patrones de rotura del oleaje mediante el uso de una red neuronal convolucional e imágenes remotas de video.

### 3.2. Objetivos Específicos

- Generar un conjunto de datos de calidad que incluya las diferentes condiciones ambientales presentes en la playa de estudio, es decir, diferentes condiciones de oleaje, mareales, climáticas y de luminosidad.
- Entrenar y validar la red neuronal convolucional utilizando el conjunto de datos creado anteriormente para encontrar los parámetros e hiperparámetros óptimos para el modelo.
- Medir el desempeño de la red en imágenes completamente nuevas para ver la capacidad de generalización de la red bajo distintas condiciones.

### 3.3. Hipótesis

- Se puede identificar patrones de rotura del oleaje de forma bidimensional solamente mediante imágenes remotas de video, sin seleccionar un umbral de intensidad y de forma eficiente computacionalmente.



## Capítulo 4

# Metodología

El proceso de rotura de oleaje es un fenómeno altamente dinámico y caótico que depende de diversos factores. Esto implica que, en playas con diferentes características, exista una alta variabilidad de este proceso. Dentro de los principales factores que alteran el proceso de rotura se encuentra la batimetría del lugar, condiciones ambientales, el efecto de la marea, etc.

Desde el punto de vista de la ingeniería costera, esto presenta un gran problema a la hora de definir cuando una ola está rompiendo y tratar de segmentar el roller del resto de la ola. Para lograr esto, se han utilizado una de las técnicas de mediciones más utilizadas en esta área como lo son las mediciones remotas, ya sea mediante radar y/o video. Enfocándose solamente en las mediciones de video, éstas de cierta forma se pueden considerar como un retrato de la visión del ojo humano.



(a) Configuración de rotura 1



(b) Configuración de rotura 2

Figura 4.1: Ejemplo de olas en rotura en diferentes configuraciones de imágenes e diferentes condiciones hidrodinámicas.

Dicho esto, en el caso hipotético que una persona se encuentre en una playa y pueda observar un determinado instante como el representado en la Figura 4.1a, la persona de forma natural podrá identificar la presencia de tres olas en rotura y diferenciarlas de las olas que no están rompiendo sin importar mucho los diferentes factores que pueden afectar el proceso de rotura. Luego, si la misma persona se encuentra en otra playa con diferentes condiciones hidrodinámicas y puede ver el instante representado en la Figura 4.1b, la persona de igual forma podrá identificar que en dicho instante una ola está rompiendo. Este sencillo ejemplo, que puede ser muy natural para el ser humano, es sumamente difícil de lograr mediante un algoritmo computacional capaz de detectar

las olas que están rompiendo en diferentes playas a partir de imágenes.

Al ver las imágenes de la Figura 4.1, se puede notar que una ola en rotura se destaca por estar formadas por pixeles de gran intensidad (color blanco) que representan el cuerpo de agua turbulento y caótico denominado roller. La corteza visual humana tiene la habilidad de poder concentrar su atención en estos pixeles y el contraste que generan respecto al resto de la imagen, para identificar ese patrón como una ola rompiente. Repetir este proceso mediante la programación de un computador implica una gran complejidad debido a que el algoritmo se puede confundir entre la intensidad de pixeles y su ubicación. Por ejemplo, una ola en rotura y la espuma remanente que deja este proceso se representan con alta intensidad de píxel en las imágenes, pero tienen distinta ubicación.

## 4.1. Datos

Los datos fueron recolectados en un experimento en la U.S Army Corp of Engineers Field Research Facility (FRF), Duck, NC durante septiembre del 2010. Se utilizó el sistema de coordenadas existente en FRF, donde la coordenada en la dirección perpendicular a la línea de la costa es denotada por  $x$ , mientras que la coordenada  $y$  se encuentra a  $18^\circ$  hacia el oeste respecto al norte (ver Figura 2.2). El experimento cuenta con dos sensores remotos: un radar marino de polarización simple (HH) (Si-Tex RADARpc-25.9) operando en un rango de microonda de 9.45 GHz y un sistema de tres cámaras de la estación de observación ARGUS III operando en el rango electro-óptico (Holman and Stanley, 2007).

Para la adquisición de datos se utilizó una matriz de pixeles en una grilla cartesiana, donde  $x = 60 - 600$ [m] y  $y = 500 - 1000$ [m] con una resolución espacial de  $\Delta x = 2$ [m] y  $\Delta y = 5$ [m], se utilizan las cámaras 0, 3 y 1 del sistema mencionado, resultando una imagen rectificadas de  $101 \times 271$  con 24 bits de profundidad. La configuración para sincronizar las mediciones de ambos sensores remotos fue la misma que se utilizó en Catalán et al. (2011), Haller et al. (2014) y Díaz et al. (2018). Sin embargo fue necesario una interpolación adicional usando la técnica del vecino más cercano sobre una malla uniforme en ambas direcciones, donde finalmente la imagen final fue cortadas en una resolución de  $512 \times 512$  como requerimiento de la red neuronal a utilizar. Además de utilizar las imágenes en una configuración de escalas de grises con una profundidad de 8 bits.

Los datos fueron recolectados durante el Surf Zone Optics Experiments (SZO) entre el 9 y 14 de septiembre del año 2010, donde se tomaron mediciones en el comienzo de cada hora durante el día con una duración de aproximadamente 17 minutos por video. La Figura 4.2 muestra las condiciones del oleaje caracterizado por una altura de ola entre  $H_{m0} \approx 0,5 - 1,2$  [m], campos de onda bimodales con periodos cortos entre  $T_p \approx 5 - 6$  [s] y una componente menos energética en el rango de  $T \approx 10 - 12$ [s]. Una dirección predominante desde el noreste con un peak de  $\theta = 45^\circ$  (positivo en dirección anti-horario), pero con una energía considerable desde el sureste con dirección  $\theta = -45^\circ$ , resultando en condiciones de oleaje bimodales. Seis conjuntos de datos siguen la evolución de las olas durante la bajamar y el último conjunto se caracteriza por un estado bimodal, dominado por la fase de la marea cercana a cero.

En total se cuenta con un total de 25 pares video-radar, en que los videos están enfocados en la zona rompiente cerca de la costa donde las olas rompen sobre la barra de arena existente. Se seleccionaron 7 pares de dataset con 758 imágenes cada uno a 0.7 Hz. Las condiciones hidrodinámicas de cada uno de estos videos se pueden apreciar en la Figura 4.2. Por Las líneas rojas discontinuas denotan los datos utilizados durante el proceso de entrenamiento y las líneas azules representan

los datos utilizados para medir la capacidad de la red. Las imágenes instantáneas corresponden a imágenes obtenidas con el sistema ARGUS y representan el comienzo de cada conjunto de datos. El orden de las imágenes es secuencial en el tiempo y el nombre de éstas corresponden a la fecha y tiempo (Mes/Día - Hora - Minuto).

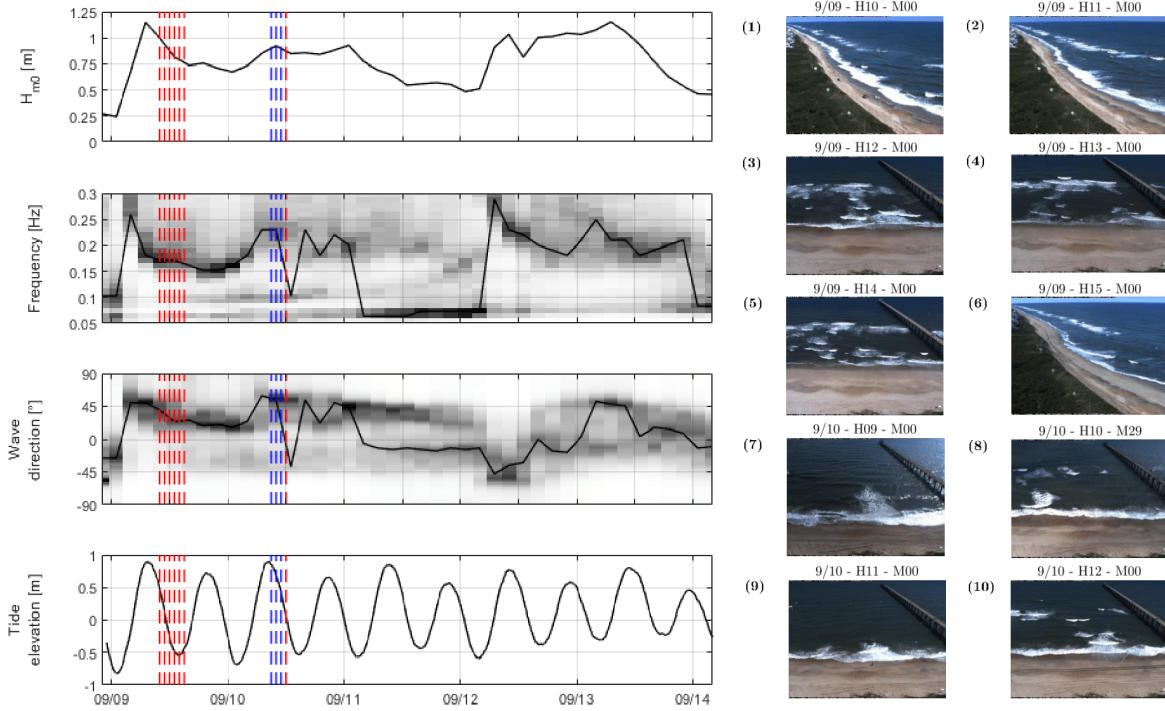


Figura 4.2: Muestra de las condiciones de oleaje presente durante los experimentos segun mediciones en el FRF y un ADCP. (a) Altura de ola significativa, (b) Espectro de frecuencia (fondo en escala de grises) y frecuencias peak (línea sólida), (c) Espectro direccional y (d) Elevación de la marea. Las líneas rojas y azules representan la muestras los conjuntos de imágenes seleccionadas. Por último, se muestra un ejemplo de cada conjunto de imágenes utilizadas.

Cada uno de estos dataset fueron pre-procesados siguiendo la metodología Catalán et al. (2011), donde los umbrales para los videos de radar y video fueron identificados mediante la función de densidad de probabilidad conjunta (JPDF) para separar las áreas de rotura activa de las demás etapas del proceso. En este trabajo, sólo utilizamos las áreas de rotura activa, las que se denominan como máscaras de roturas. El rango de valores de umbrales fueron  $I^t \sim 150 - 160$  y  $\sigma_0^t \sim -40 - -30$  dB, los cuales son consistente con los valores utilizados por Catalán et al. (2011) y Díaz et al. (2018).

El objetivo es entrenar un algoritmo de Deep Learning que pueda identificar olas en rotura individualmente basado solo en el uso de datos ópticos. Sin embargo, la mayoría de las imágenes consideran una larga región de olas no rompientes, entonces todas las imágenes fueron enfocadas en la rotura de las olas sobre la barra de arena existente, por lo que se posicionó una matriz de  $512 \times 512$  pixeles sobre la región de interés mencionada. Las máscaras de rotura son binarias donde el valor 1 corresponde a los pixeles identificados como rotura y 0 los demás casos. De esta forma, el conjunto de datos final consiste en 7 corridas con 758 imágenes por corrida, resultando en 5306 imágenes para el sistema de cámaras de video, el radar y sus respectivas máscaras de rotura como se muestra en la Figura 2.5. Estas máscaras se tratan como datos reales y no se considera una

relacional temporal entre las imágenes. Esto resulta beneficioso para el entrenamiento de la red, ya que es deseable que sean diferentes y no estén correlacionadas para maximizar las diferentes condiciones que alimentan la red. Esta decisión no afecta a la posible utilización de los resultados como series temporales, dado que se pueden concatenar los mapas de segmentación en el tiempo para, por ejemplo, obtener un seguimiento espacio-temporal de las olas individualmente (Díaz et al., 2018).

## 4.2. Red Neuronal Convolutiva

Por las razones ya expuestas se hace necesario utilizar nuevas herramientas, por ejemplo, el uso de herramientas de Inteligencia Artificial, específicamente en este trabajo se abordarán las redes neuronales convolucionales (CNN). Una CNN es un tipo de red neuronal artificial mediante aprendizaje supervisado, es decir, se le enseña a la red lo que debe aprender. El principal objetivo de este tipo de red es imitar la corteza visual del cerebro humano y así, poder identificar objetos dentro de las imágenes. Esta característica hace que se vuelva una gran herramienta para resolver este problema debido a que, como se mencionó al principio del capítulo 4, una vez que una persona aprende a identificar olas rompientes puede identificar estos mismos patrones en diferentes condiciones tanto ambientales como hidrodinámicas. Entonces, al imitar el funcionamiento de la corteza visual humana, la red puede aprender a identificar olas en rotura de forma similar a una persona.

Específicamente, la red a utilizar se denomina U-Net (Ronneberger et al., 2015) y permite la segmentación de objetos dentro de la imagen, es decir, identifica el objeto y su posición. A continuación, se presenta la arquitectura de la red neuronal.

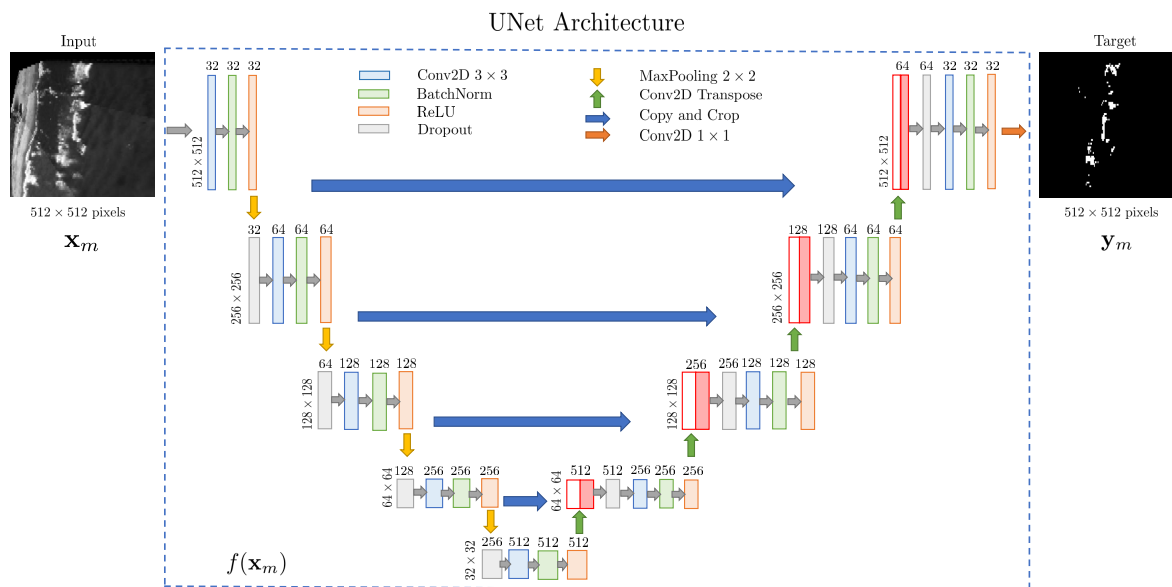


Figura 4.3: Esquema de la arquitectura de la red neuronal convolutiva U-Net. Fuente: elaboración propia

### 4.3. Conjuntos de entrenamiento, validación y prueba

Para minimizar un posible sesgo producto de seleccionar imágenes con condiciones lumínicas y de oleaje particulares, se utiliza un proceso automático en el cual se selecciona un número aleatorio uniforme entre 1 y 7 dado los siete conjuntos de imágenes con los que se cuenta y luego se selecciona un número aleatorio uniforme entre 1 y 758 para extraer una imagen del respectivo video seleccionado con su respectiva máscara de rotura. Se eligió este proceso en lugar de seleccionar un número aleatorio entre 1 y 5306 para minimizar la elección de una única condición de oleaje.

El proceso de selección de imágenes fue repetido 700 veces para definir el conjunto de entrenamiento, el cual está formado por 700 pares de imágenes video-máscara. Las imágenes seleccionadas fueron removidas del conjunto original y el proceso fue repetido 400 veces para definir el conjunto de prueba. Por su parte, el conjunto de validación es un subconjunto del conjunto de entrenamiento, el cual corresponde entre 20 % a 30 % de los datos de entrenamiento. Dicho esto, el algoritmo considera las imágenes ópticas como dato de entrada y las máscaras de rotura como dato de salida.

### 4.4. Arquitectura de Red Neuronal

La arquitectura de U-Net se caracteriza por estar formada por dos rutas (ver Figura 4.3). La primera corresponde a una ruta de contracción que tiene por objetivo la detección del fenómeno u objeto que se desea segmentar, mientras que la segunda ruta permite ubicar dicho fenómeno u objeto en la posición que se detectó. La ruta de contracción está formada por operaciones matriciales denominadas convoluciones, la aplicación de una función de activación y la reducción de neuronas mediante una técnica de subsampling. De la misma forma, la ruta de expansión se caracteriza por contar con operaciones de convoluciones traspuestas o inversa y una operación de copy and crop, que permite recuperar la información de dimensiones y borde que se pierden en la ruta de contracción.

#### 4.4.1. Convolución

El proceso de convolución consiste en tomar un grupo de pixeles cercanos de la imagen de entrada y aplicar un producto escalar entre el grupo de píxel seleccionado y una matriz de pesos denominada kernel. La matriz kernel recorre todos los pixeles de la imagen de entrada de derecha a izquierda y de arriba hacia abajo. Por último, se destaca que este proceso se repite todas las veces que se estime conveniente según los filtros que se utilicen, por ejemplo, si se utilizan dos filtros implica que se realiza la operación de convolución para dos kernel distintos. Esta operación tiene por objetivo extraer las características más importantes de las imágenes y así, como resultado, obtener mapas de características o capa convolucionada.

La Figura 4.4.a) representa a un esquema de una la operación convolución. Para este problema en particular, el filtro utilizado o matriz kernel está compuesto por una matriz de 3x3, donde cada una de las 9 entradas de la matriz es un peso de la red (o neurona) que va convergiendo al valor final a medida que pasa el entrenamiento. El filtro barre la imagen de izquierda a derecha y de arriba hacia abajo dando pasos de 1 o 2 pixeles según se estime conveniente. Luego, se realiza un producto escalar entre la matriz kernel y la matriz de 3x3 seleccionada de la imagen o de los mapas de características. De este producto se obtiene un escalar correspondiente a una entrada de la capa convolucionada, así se van obteniendo las diferentes entradas de los mapas de características hasta

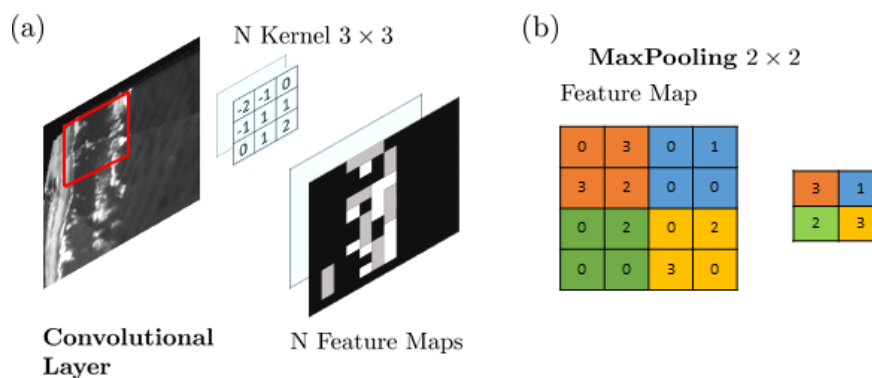


Figura 4.4: Ejemplos de operaciones matemáticas utilizadas por la red neuronal convolucional U-Net. (a) representa el proceso de convolución, donde el hiperparámetro número de filtros o *number of filter* es equivalente al número de *features maps* y (b) corresponde a la operación Max-Pooling.

completar el barrido en la matriz convolucionada.

#### 4.4.2. Función de Activación

Luego del proceso de convolución, es necesario aplicar una función de activación con el objetivo de activar y desactivar neuronas ocultas dado que no todas las neuronas transmiten información relevante para realizar buena predicción. Específicamente, se utiliza la función de activación ReLU, la cual retorna el máximo entre cero y cada una de las entradas del mapa de características, en terminos matemático se puede expresar como la ecuación 4.1. Además, contar con esta función permite que la red pueda lograr representaciones menos densas, lo que genera una mejor representación biológica considerando que el cerebro humano no utiliza todas las neuronas al realizar una acción (Glorot et al., 2011).

$$f(x) = \text{máx}(x, 0) = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (4.1)$$

#### 4.4.3. Subsampling con Max-Pooling

Es necesario aplicar una reducción de neuronas para así disminuir la complejidad del espacio de soluciones donde se busca la solución óptima de la red neuronal, es decir, a medida que aumenta la profundidad del modelo aumenta exponencialmente el número de parámetros de este y no todos son relevantes. Para esto se aplica la operación Max-Pooling, la cual retorna el máximo de un conjunto de pixeles como se puede ver en la 4.4.b). Por ejemplo, en el caso de la Figura mencionada, se aplica Max-Pooling con una matriz de 2x2 ejecutándose de forma similar a la matriz kernel en el proceso de convolución, pero barre el mapa de características saltando de 2 pixeles. Si bien, al aplicar esta operación se pierde información, permite concentrar la información de mayor relevancia obtenida con las operaciones anteriores.

## 4.5. Entrenamiento y Validación de la red

Para mejorar la generalización, reducir el sesgo y el sobreajuste (overfitting) se realiza un exhaustivo análisis de los parámetros e hiperparámetros de la red. Se entrenaron y probaron diferentes configuraciones de hiperparámetros de forma totalmente independiente. Para esto, todo el conjunto de entrenamiento tanto las imágenes como las máscaras son divididos en  $K$  subconjuntos, en el cual uno de estos es utilizado para validación y los restantes para entrenamiento. El término de entrenamiento se puede entender como el proceso que permite la obtención de los parámetros de la red, mientras que el proceso de validación se entiende como la determinación de los hiperparámetros.

Para cada uno de los  $K - 1$  conjuntos de entrenamiento, el algoritmo selecciona de forma aleatorio un número fijo de imágenes (el valor es definido por el hiperparámetro *batch size* = [2, 4, 8, 16]) y el entrenamiento se realiza una cierta cantidad de veces (determinado por el hiperparámetro *epochs* = 50). Durante el entrenamiento la red usa un algoritmo de optimización para identificar el mejor conjunto de parámetros (Kingma and Ba, 2014). Considerando que el método en cuestión es un problema de clasificación binaria, se utiliza la función sigmoïdal 4.2 para clasificar las neuronas,

$$g(z) = \frac{1}{1 + e^{-z}} \quad \begin{cases} \geq 0,5 \rightarrow \text{clase 1, (rotura)} \\ < 0,5 \rightarrow \text{clase 0, (no rotura)} \end{cases} \quad (4.2)$$

donde  $z$  representa cada neurona. Además, se utiliza la función *binary cross-entropy* 4.3 (BCE) para cuantificar el error de la red, donde penaliza los píxeles que son mal clasificados.

$$\ell(f(\mathbf{x}_m), y_m) = -y_m \ln f(\mathbf{x}_m) + (1 - y_m) \ln(1 - f(\mathbf{x}_m)). \quad (4.3)$$

El proceso de optimización es desarrollado internamente por la red y permite la identificación de los mejores parámetros o peso para cada neurona, aplicando una versión mejorada del gradiente descendente estocástico denominado ADAM (Kingma and Ba, 2014). El uso del algoritmo ADAM requiere la definición del hiperparámetro *learning rate* = [10<sup>-2</sup>, 10<sup>-3</sup>, 10<sup>-4</sup>], el cual se puede interpretar como la tasa de convergencia.

El proceso de validación se debe repetir  $K$  veces (usando el mismo conjunto de hiperparámetros) intercambiando cada uno de los conjuntos de entrenamiento y validación. Esto produce  $K$  estimaciones de precisión y pérdida del modelo, las que luego son promediadas para encontrar las estimaciones para cada conjunto de hiperparámetros. Esta técnica es conocida como *Cross-Validation* y específicamente en esta tesis se utiliza  $K = 3$ .

Finalmente, la inicialización de los parámetros es crucial para modelos de aprendizaje profundo como U-Net dado que al ser redes con grandes cantidades de capas, a medida que se realiza el entrenamiento, un pequeño cambio en el gradiente de convergencia en los parámetros utilizados en las capas superiores significa un gran cambio en las capas más profundas. Por esta razón, se utiliza la inicialización *He normal* (He et al., 2016). En esta implementación, el proceso de validación se realizó 10 veces con distinta semilla para cada combinación de hiperparámetros, donde el promedio de estas repeticiones es considerado como el promedio de la respectiva combinación de hiperparámetros.

Para evaluar la precisión de las diferentes configuraciones de hiperparámetros, se utiliza la métrica *F1 - Score* la cual es altamente utilizada en este tipo de problemas (Goutte and Gaussier,

2005). Esta función se define como

$$F_1 = \frac{t_p}{t_p + \frac{1}{2}(f_p + f_n)}, \quad (4.4)$$

donde  $t_p$  es el número de verdaderos positivos correctamente clasificados,  $f_n$  es el número de falsos negativos, los que corresponden a pixeles de rotura verdaderos que no fueron clasificados y  $f_p$  es el número de falsos positivos, es decir, un pixel que pertenece a una ola no rompiente verdadera y es clasificado como rompiente por la red. Esta métrica esta entre 0 y 1 y un alto valor de  $F_1$  corresponde un alto valor de precisión de la red.

Tabla 4.1: Espacio de hiperparámetros utilizados. Los valores en negrita representan la mejor combinación.

Batch size	<b>2</b> , 4, 8, 16
Dropout	0.5, <b>0.8</b>
Learning rate	<b>10</b> <sup>-2</sup> , 10 <sup>-3</sup> , 10 <sup>-4</sup>
Number of filters	4, 8, 16, <b>32</b>

El resumen de los hiperparámetros utilizados son mostrados en la Tabla 4.1, donde se puede ver en negrito la configuración óptima. Se realizaron un total de 96 experimentos, uno por cada combinación de hiperparámetros, que a su vez fueron corridos 10 veces para cada una de las semillas, resultando en un total de 960 corridas, pero solo 96 valores de la pérdida BCE y la métrica  $F_1$  (ecuaciones 4.3 y 4.4). De esta forma, se presentan las combinaciones de mejor desempeño en la Tabla 4.2. Para el desarrollo del prototipo del modelo se utilizó *Google Colab* y el proceso de validación fue ejecutado gracias al Laboratorio Nacional de Computación de Alto Rendimiento (NLHPC) de la Universidad de Chile ([www.nlhpc.cl](http://www.nlhpc.cl)), usando el entorno GPU con dos nucleos Intel Xeon Gold 6152 CPU @ 2.10 GHz y una GPU Nvidia Tesla V100 con 64 GB de memoria RAM.

Tabla 4.2: Clasificación del desempeño de las combinaciones de hiperparámetros. Se denota en negrita las 5 mejores configuraciones las que están clasificados su mejor valor de precisión y pérdida.

Clasificación	F1-Score Eq. 4.4	Número de combinación	Pérdida Eq. 4.3	Número de combinación
1	0,878	<b>11</b>	0,00849	2
2	0,869	<b>3</b>	0,00859	<b>27</b>
3	0,863	35	0,00859	<b>3</b>
4	0,858	<b>27</b>	0,00862	<b>15</b>
5	0,835	26	0,00871	50
6	0,821	59	0,00873	7
7	0,811	<b>15</b>	0,00878	<b>31</b>
8	0,801	<b>31</b>	0,00878	<b>11</b>
9	0,798	34	0,00905	10
10	0,794	25	0,00908	25

# Capítulo 5

## Resultados

Como resultado del proceso de validación se obtienen 96 vectores para todas las combinaciones de hiperparámetros, cada una con sus respectivos valores de pérdida BCE y de precisión  $F1 - Score$ . Los resultados de la pérdida BCE fueron clasificados en orden creciente y los de precisión en orden decreciente. La Tabla 4.2 muestra la clasificación de los 10 mejores modelos según las diferentes combinaciones de hiperparámetros. Es importante destacar que es muy poco probable que los mejores modelos clasificados según la pérdida BCE sean concordantes con los clasificados por la métrica de precisión, por esta razón los modelos fueron clasificados independientemente. Así, se obtuvieron un total de 15 combinaciones, de los cuales se consideran los 5 mejores según la BCE y  $F1 - Score$ . Dichos valores corresponden a las combinaciones 3, 11, 15, 27 y 31 y se denotan por negrita en la Tabla 4.2.

Estas cinco combinaciones candidatas fueron entrenadas nuevamente y los modelos resultantes fueron aplicados sobre el conjunto de validación. Los resultados obtenidos se muestran en la Tabla 5.1, los que fueron clasificados según su desempeño en el conjunto de imágenes mencionado. El modelo entrenado con la combinación 3 de hiperparámetros fue el que obtuvo el mejor desempeño en los conjuntos de entrenamiento y validación, por lo tanto se considera esta configuración de hiperparámetros como el modelo final y con esto se finaliza el proceso de validación.

Tabla 5.1: Cinco mejores combinaciones de hiperparámetros y sus respectivas métricas en los conjuntos de entrenamiento y validación.

Comb.	Hyper-Parameters					BCE Loss $\ell(f(\mathbf{x}_m), y_m)$		Accuracy, $F_1$	
	Batch Size	Epoch	Learn. Rate	Drop.	Num. Filt.	Train.	Valid.	Train.	Valid.
<b>3</b>	2	50	0.010	0.8	32	<b>0,001</b>	<b>0,007</b>	<b>0,957</b>	<b>0,895</b>
11	2	50	0.001	0.8	32	0,005	0,008	0,850	0,813
15	2	50	0.001	0.5	32	0,004	0,007	0,886	0,840
27	4	50	0.010	0.8	32	0,004	0,010	0,918	0,863
31	4	50	0.010	0.5	32	0,007	0,009	0,807	0,780

Además de encontrar la combinación de hiperparámetros óptima mediante técnicas de machine learning como *Cross Validation*, se realiza una cuantificación adicional para medir el desempeño de la red en base al análisis de píxeles según su correcta identificación. Para esto se consideran tres casos: los píxeles verdaderos son aquellos que son identificados como oleaje rompiente por ambas metodologías, los píxeles errados correspondiente a los píxeles que fueron clasificados como oleaje

no rompiente por U-Net, pero clasificados como rotura por Catalán et al. (2011) y finalmente se tienen los pixeles falsos que reflejan los pixeles correctamente clasificados por U-Net, pero no por la metodología original. La cantidad de pixeles verdaderos y los pixeles errados son complementarios, por lo tanto fueron normalizados según la cantidad total de pixeles verdaderos (suman 100%), mientras que los pixeles falsos fueron normalizados según la cantidad de pixeles perteneciente a rotura según U-Net.

El resumen de estos resultados se presentan en la Tabla 5.2, donde se puede ver que la combinación 27 es uno de los modelos que alcanza un gran porcentaje de pixeles verdaderos, pero esto éxito implica a su vez un alto porcentaje de pixeles falsos. El caso totalmente opuesto corresponde a la combinación 31, la cual tiene un pequeño porcentaje de pixeles falsos, pero disminuye su porcentaje de pixeles verdaderos considerablemente. Por otro lado, la combinación 3, la cual fue la que mostró mejor desempeño en el proceso de validación, muestra un bajo porcentaje en detecciones falsos y un porcentaje aceptable de detecciones verdaderas, por lo que se considera la mejor combinación nuevamente.

Tabla 5.2: Porcentajes de la cantidad de pixeles identificados por cada combinación de hiperparámetros candidatos.  $\sigma$  denota la desviación estándar.

Combination	True Pixels Mean $\pm \sigma$	Fail Pixels Mean $\pm \sigma$	False Pixels Mean $\pm \sigma$
<b>3</b>	<b>70,88 <math>\pm</math> 16,04</b>	<b>29,11 <math>\pm</math> 16,04</b>	<b>23,74 <math>\pm</math> 15,25</b>
11	68,27 $\pm$ 18,22	31,72 $\pm$ 18,22	24,93 $\pm$ 20,25
27	76,58 $\pm$ 14,22	23,41 $\pm$ 14,22	39,10 $\pm$ 21,69
31	59,54 $\pm$ 19,47	40,45 $\pm$ 19,47	16,20 $\pm$ 13,44
15	71,38 $\pm$ 15,04	28,61 $\pm$ 15,4	27,52 $\pm$ 14,85

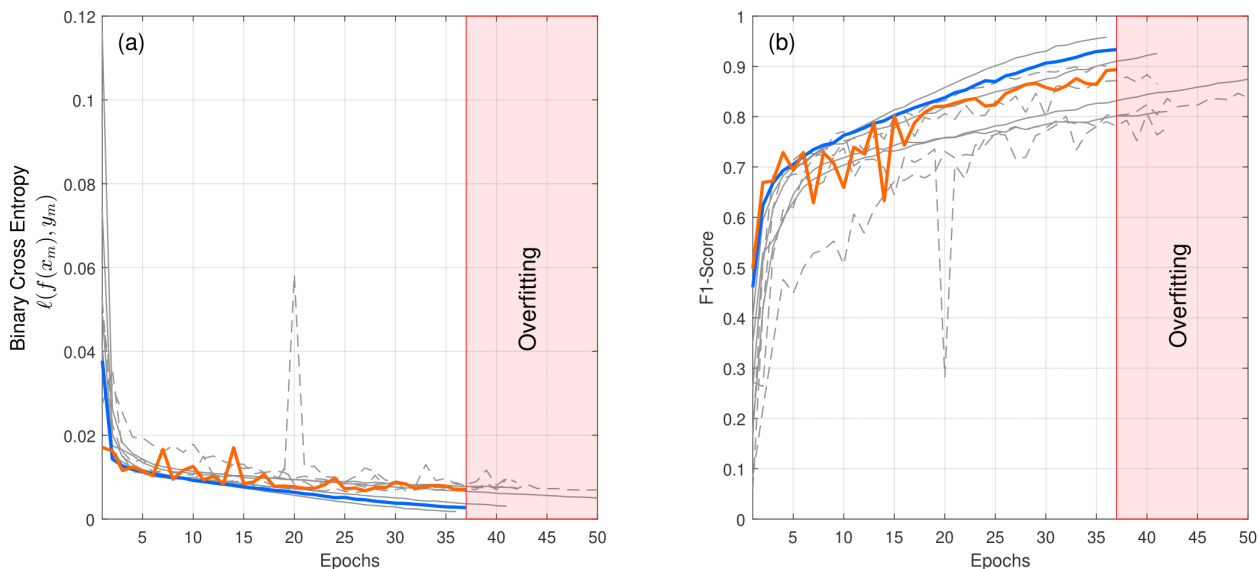


Figura 5.1: Evolución de la función de pérdida BCE y la métrica de precisión  $F1 - Score$ . El desempeño de la combinación 3 durante el entrenamiento se muestra a través de la línea azul y durante la validación a través de la línea naranja. Las líneas grises corresponden a las demás combinaciones, donde con línea continua se representa el proceso de entrenamiento y en línea discontinua la validación.

Como evaluación final se presenta la evolución de la función de pérdida BCE y la precisión  $F1$  durante el proceso de entrenamiento para evaluar la posibilidad de sobreajuste (overfitting). Este fenómeno es muy común en problemas de Deep Learning en donde el modelo memoriza las características presentes en el conjunto de entrenamiento, lo que disminuye el desempeño del modelo sobre datos completamente nuevos como el conjunto de prueba. El comportamiento de las métricas durante el proceso de entrenamiento se muestra en la Figura 5.1, donde la línea continua azul representa la evolución de la combinación durante el entrenamiento y con línea naranja se presenta la validación para la combinación 3 de hiperparámetros. Las otras cuatro combinaciones candidatas se muestran en el fondo del gráfico, con línea continua y discontinua se representa el entrenamiento y validación respectivamente.

De la Figura 5.1 permite concluir que los valores de la pérdida disminuyen y de la precisión aumentan rápidamente para las combinaciones estudiadas. Además, se puede ver que los valores de BCE para los diferentes modelos siguen la misma tendencia, mientras que respecto a la precisión, la combinación 3 tiene un comportamiento diferente a los demás modelos presentados teniendo un mejor desempeño durante este proceso. El sobreajuste puede ser detectado como una disminución en la tasa de cambio de la curva de pérdida de BCE y de  $F1$  a medida que aumenta la cantidad de *epochs*. Este fenómeno se detecta cerca de la *epoch* 36 para la combinación 3, gráficamente esto se aprecia donde comienza el rectángulo rojo en la figura mencionada.



# Capítulo 6

## Discusión

### 6.1. Análisis de píxeles de máscaras de rotura

#### 6.1.1. Análisis cualitativo

Uno de los principales desafíos de implementar técnicas de machine learning es tener un buen desempeño en los tres conjuntos de datos utilizados, el conjunto de entrenamiento, de validación y de prueba. Esto es especialmente desafiante para el caso de la clasificación de olas rompientes, dado que ha sido sumamente difícil encontrar un algoritmo robusto que permita identificar este fenómeno de forma automática. Para esta implementación se ha asumido que el método propuesto por Catalán et al. (2011) basado en la fusión de sensores es lo suficientemente robusto para obtener máscaras de rotura de forma precisa sin supervisión. Sin embargo, asumir que estos datos son completamente reales puede traer consigo algunos errores, lo que explica en parte la aparente baja tasa de detección de la red, promediando un 71 % de detección de píxeles verdaderos.

A continuación se presenta un ejemplo de lo mencionado anteriormente, en la Figura 6.1 y 6.2 se muestran dos imágenes correspondientes al conjunto de prueba y su máscara de rotura obtenida con la metodología original y la obtenida con U-Net. El cuadro A corresponde a la imagen 348 del video número 2 (Sep 9, 11:00, Fig.4.2) y el cuadro B corresponde a la imagen 719 de la corrida número seis (Sep 9, 15:00, Fig.4.2). De la Figura 6.1 se puede ver que el método de Catalán et al. (2011) clasifica dos olas en rotura que están en sucesión en el centro de la imagen como una sola ola en rotura con una alta producción de espuma remanente, la cual es altamente brillante. En consecuencia, el método de fusión de sensores falla en distinguir la cantidad de olas en rotura y las conecta como un solo evento. Analizando este proceso de forma física, es poco probable encontrar olas en rotura que presenten una geometría rectangular. Por su parte, U-Net puede identificar las dos olas en rotura por separado y con una forma físicamente posible. Este problema se ve recurrentemente durante los conjuntos de datos utilizados, dado que para solucionar esto se debe ajustar los umbrales de intensidad del método original permitiendo identificar de mejor forma este tipo de olas, pero se perdería la precisión sobre otras. A su vez, esto disminuye la cantidad de píxeles verdaderos identificados por U-Net y por ende explica el bajo porcentaje de detección.

La otra situación son los píxeles falsos, que son píxeles clasificados como rotura por U-Net, pero no por el método original. Esto se refleja en la Figura 6.2, en la cual corresponde a una condición de baja energía dentro del conjunto de datos. Ambas metodologías muestran dificultad para identificar las olas pequeñas en rotura, esto en parte se debe a que la cámara 1 cuenta con baja condición

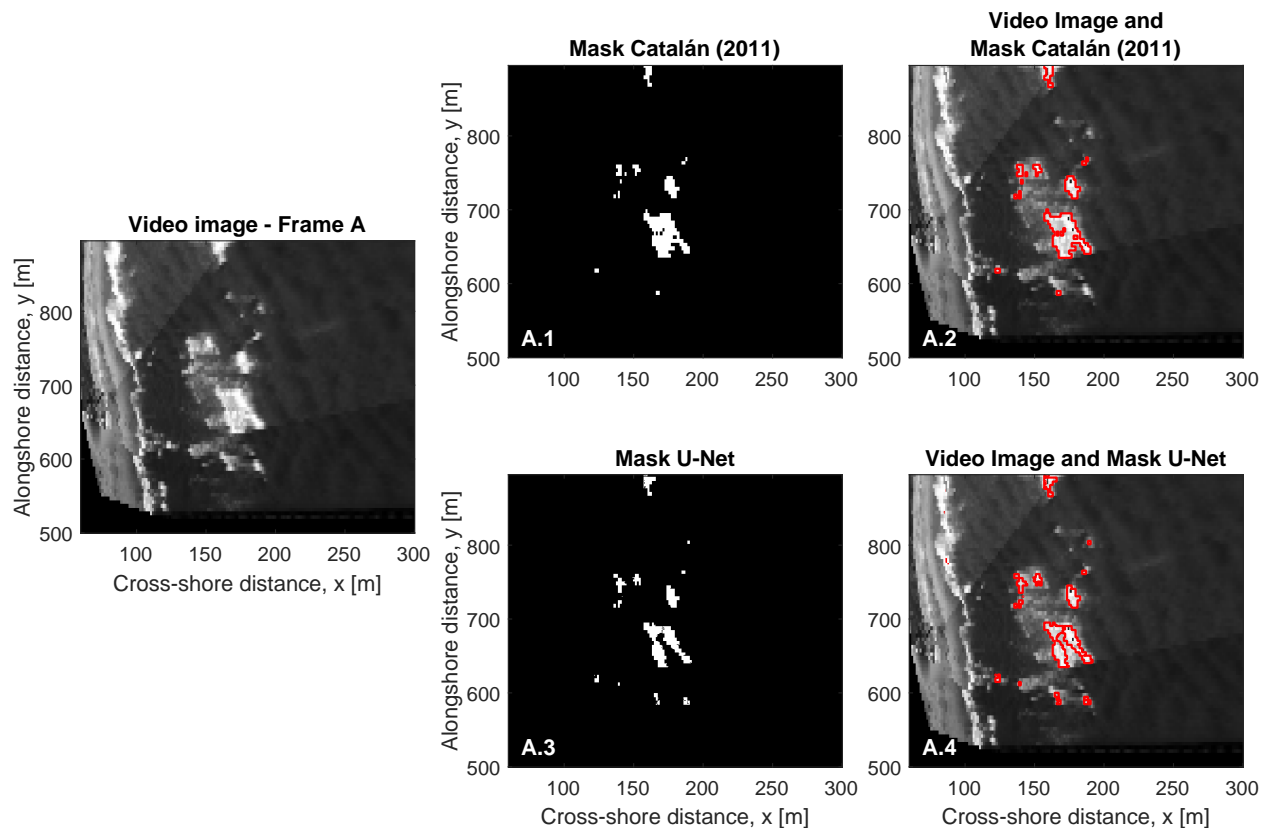


Figura 6.1: Ejemplo A resultados obtenidos con ambos métodos. Frame A corresponde a la imagen 348 del día 9 de Septiembre a las 11:00 (hora local). Los gráficos del medio corresponde a las máscaras de roturas obtenidas con la metodología propuesta por Catalán et al. (2011) (A.1) y por U-Net (A.2) y los gráficos de la derecha muestran la respectiva máscara de rotura sobre la imagen A.

lumínica. De hecho, el fenómeno contrario ocurre en la parte proveniente de la cámara 3, la que cuenta con un alto nivel de brillo permitiendo a U-Net identificar una gran ola rola rompiente en cerca de la posición  $(x, y) = (80, 800 - 900)$ [m] en las figuras 6.2-B.1 y 6.2-B.3. Se destaca que las olas mencionadas no fueron identificadas en su totalidad por la metodología original, aumentando el porcentaje de píxeles falsos. Además, analizando la parte de la imagen obtenida mediante la cámara 0, se aprecia que U-Net puede identificar mucho mejor la forma y el tamaño del roller, lo que en consecuencia aumenta la cantidad de píxeles falsos y píxeles errados. U-net obtiene un buen desempeño en identificación de eventos de rotura cerca de la línea de costa, más aún si consideramos que la rotura en esta parte de la costa no está presente en los datos de entrada. Recordando que el método utilizado corresponde a un método de aprendizaje supervisado, donde se le enseña mediante el objetivo  $y_m$  lo que la red debe aprender, da pie a argumentar que U-Net puede aprender el proceso físico de rotura en lugar de solo identificar donde existe alta intensidad de píxeles y segmentar estas áreas.

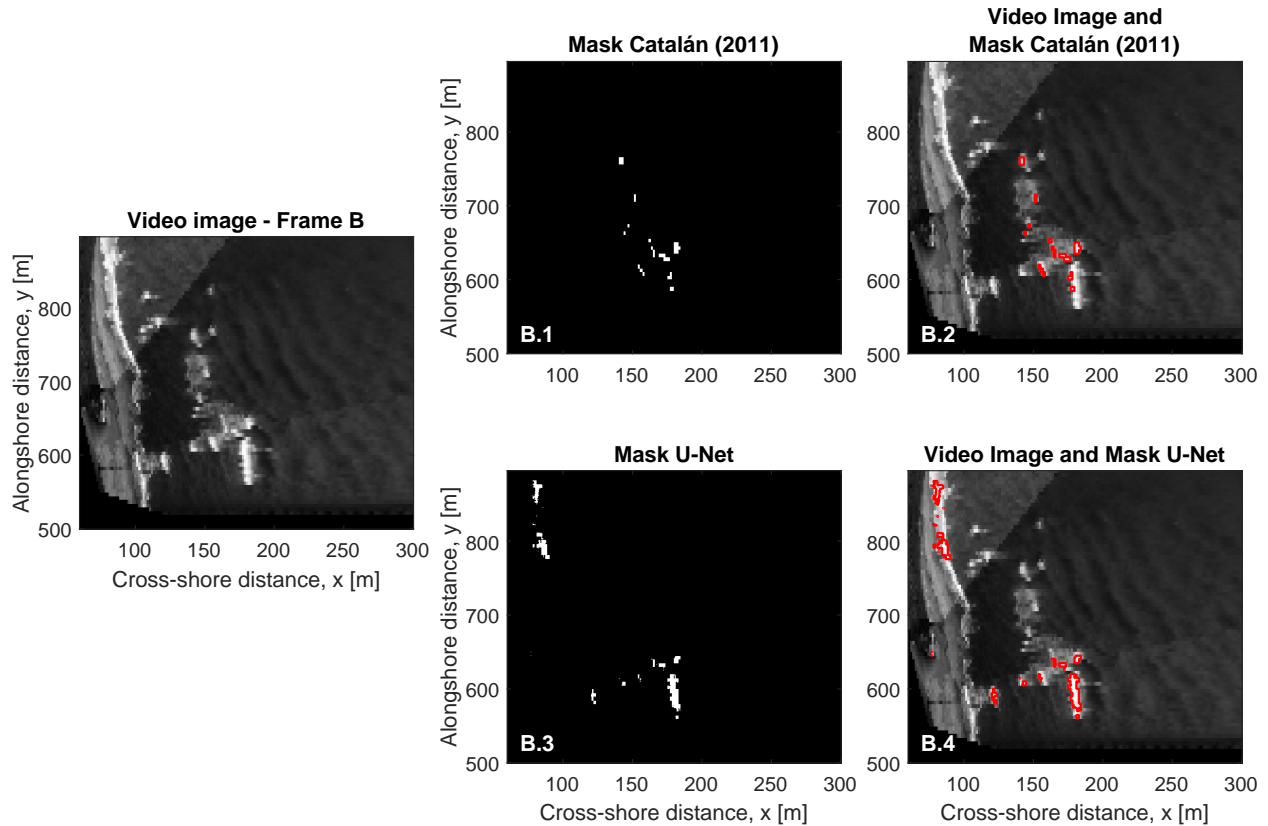


Figura 6.2: Ejemplo B resultados obtenidos con ambos métodos. Frame A corresponde a la imagen 348 del día 9 de Septiembre a las 15:00 (hora local). Los gráficos del medio corresponde a las máscaras de roturas obtenidas con ambos métodos y los gráficos de la derecha muestran la respectiva máscara de rotura sobre la imagen B.

### 6.1.2. Análisis cuantitativo

Con el fin de asegurar la aplicación del modelo U-Net entrenado y validado en este trabajo y para demostrar su capacidad de generalización, se realiza un último análisis sobre todo el conjunto de datos disponible (5308 imágenes). Este conjunto de datos está compuesto por pares de imágenes-máscaras utilizados en el proceso de entrenamiento, validación y prueba, pero también se incluyen tres nuevas corridas (758 pares cada uno) que no fueron utilizados en esta implementación. Estos nuevos datos se pueden apreciar en una línea discontinua azul en la Figura 4.2. Los resultados se reflejan en la Figura 6.3, en primer lugar se presenta una imagen de video de referencia. Las subfiguras 6.3.b - 6.3.d muestran el porcentaje de los pixeles verdaderos, errados y falsos sobre un total de 10 conjuntos de imágenes. Como se puede ver los pixeles verdaderos alcanzan un gran porcentaje de detección por el modelo U-Net en gran parte del dominio, con valores cerca al 80%. La detección de pixeles verdaderos varía dependiendo de la cámara que se analice, donde se alcanza un porcentaje cerca del 60% para la cámara 1. La detección de pixeles falsos se puede considerar generalmente baja, posiblemente afectada por el gran desempeño de U-Net en la predicción de oleaje en rotura en la línea de la costa.

Por otro lado, la subfigura 6.3.e muestra la variación de la detección sobre diferentes conjuntos de imágenes. Esta subfigura fue normalizada por la suma de todos los pixeles considerandos como

datos reales sumando además los pixeles falsos. Los pixeles verdaderos (Figura 6.3.d) está en el rango entre 65 % y 70 % con un conjunto de datos que alcanza un excelente desempeño (Sep 9, 10:00) y dos conjuntos con un pobre desempeño (Sep 9, 15:00 y Sep 10 9:00). De estos tres, dos formaron parte del conjunto de datos que se utilizó para la creación de los conjuntos de entrenamiento, validación y prueba, mientras el tercer conjunto nunca fue visto por la red neuronal. Por lo tanto, se concluye que la red entrenada en este trabajo puede ser aplicada con éxito en nuevos conjuntos de datos. Los bajos resultados ocurren para una condición de oleaje baja y relativamente tarde en el día (15:00 hrs), esto produce que cambie las configuraciones de la cámara a través de la imagen y se reduzca la intensidad de rotura para las olas pequeñas aumentando la detección de pixeles falsos, de lo cual se obtienen porcentajes no despreciables en estas corridas.

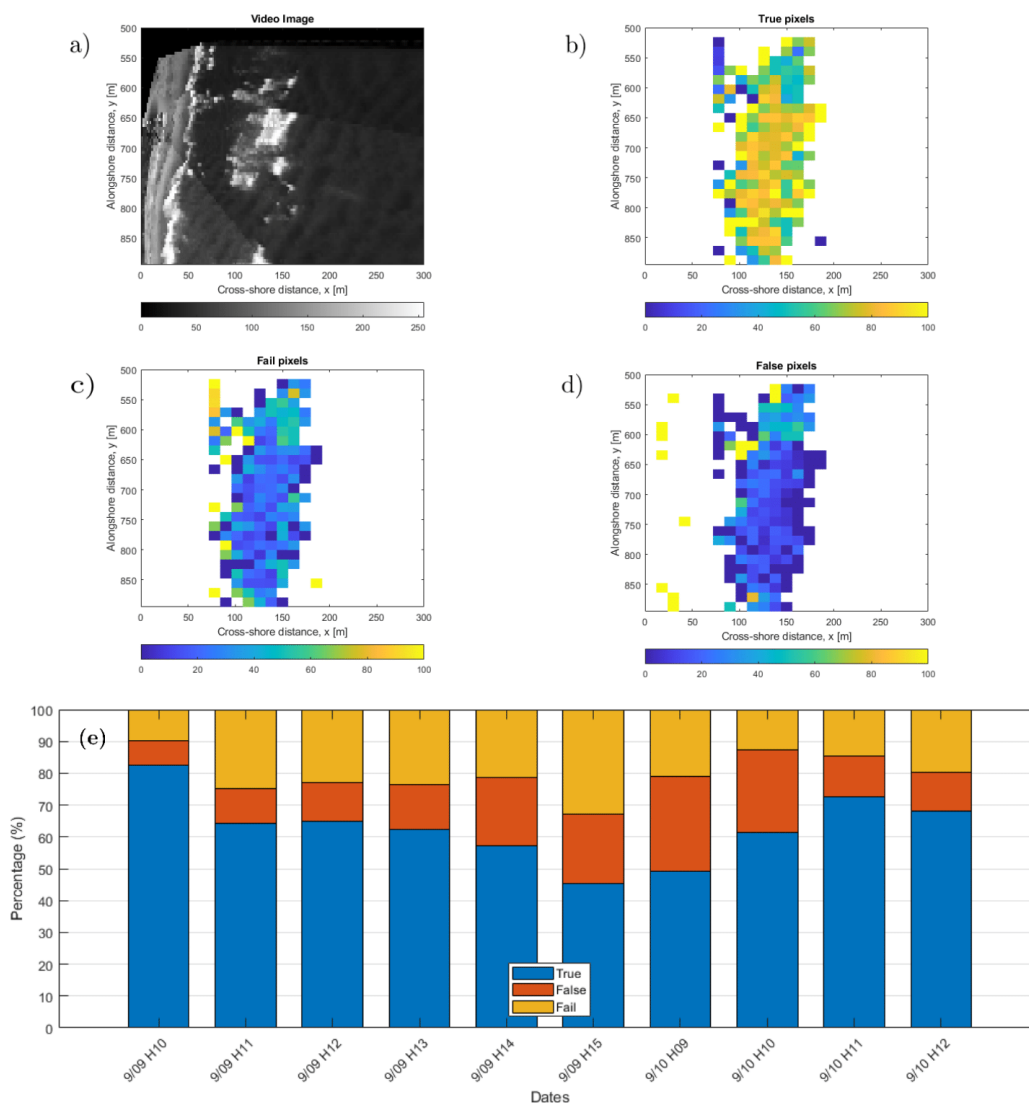


Figura 6.3: Estadística de la detección de oleaje rompiente en todas las imágenes del conjunto de datos. (a) Imagen de video de referencia; mapas de porcentajes de (b) pixeles verdaderos; (c) pixeles errados; (d) pixeles falsos; (e) Histograma que representa la evolución de la distribución de pixeles verdaderos (azul), errados (naranja) y falsos (naranja claro) sobre diferentes condiciones lumínicas y de oleaje.

## 6.2. Aplicación en nuevo dominio

Considerando la gran importancia que tiene la rotura en los procesos costeros y, por ende, la necesidad de identificarlo y cuantificarlo, este trabajo marca el comienzo de una excelente oportunidad de investigación para tratar de generalizar presente método logrando una eficiente y precisa detección de los patrones de rotura de forma bidimensional. La principal idea es transferir el conocimiento adquirido por la red entrenada en Duck, NC, USA. hacia nuevas playas de interés.

En la comunidad de machine o Deep Learning, esto no es una tarea sencilla de abordar dado que, como se puede ver en la Figura 4.3, la red neuronal puede ser vista como una función  $f(x_m)$ . Esta función recibe como entrada la matrix  $x_m$  que representa la imagen y se obtiene la respectiva salida  $y_m$  correspondiente a la máscara de rotura identificada a partir de la imagen de entrada. Los parámetros de la función son ajustados según los datos de entrenamientos y por lo tanto, la red tiene un buen desempeño en datos que responde a dicha distribución de intensidad de píxeles, no a datos con distinta distribución o, en otras palabras, con distintas condiciones de oleaje y ambientales. Esto se conoce como dominio fuente (datos del entrenamiento) y dominio objetivo (datos con distintas condiciones) en la comunidad de IA.

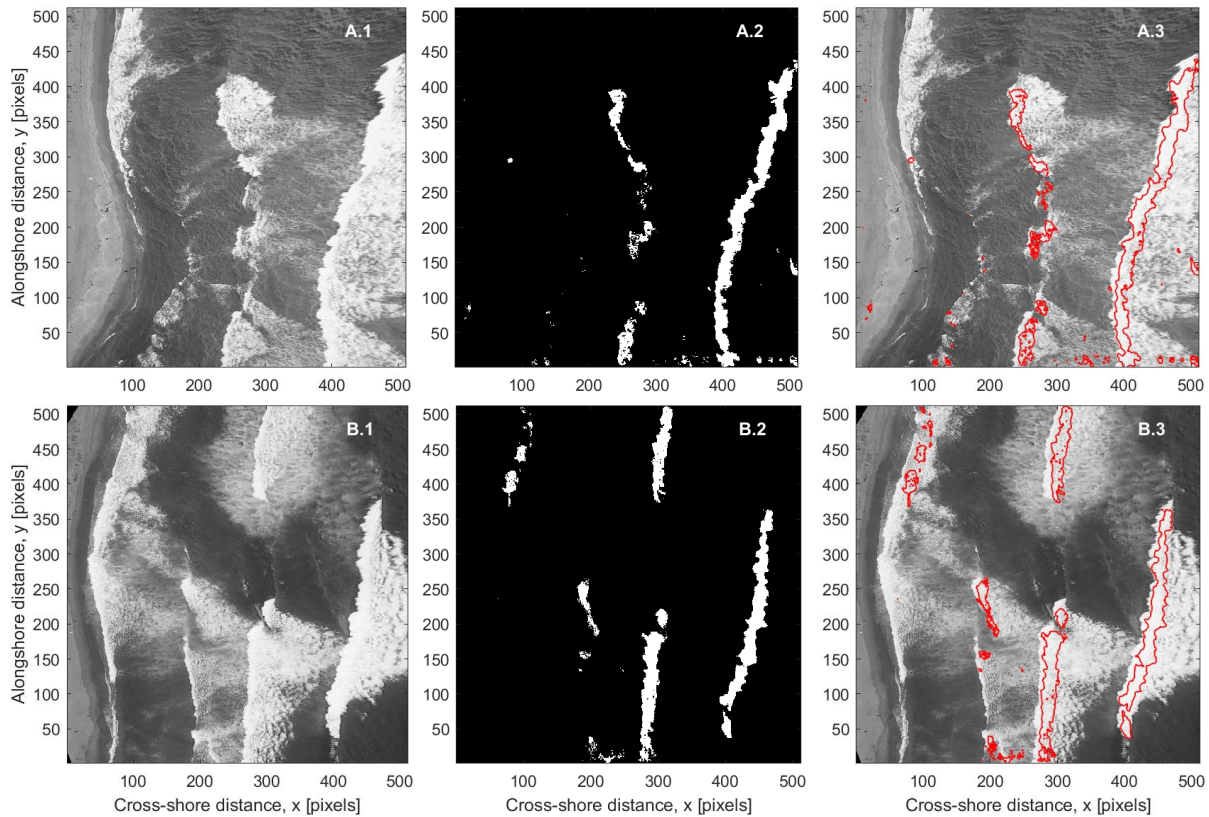


Figura 6.4: Ejemplo de aplicación del modelo U-Net sobre imágenes capturadas mediante Dron en las cercanías de Las Cruces, Chile. A.1 y B.1 corresponden a dos imágenes obtenidas mediante Dron, A.2 y B.2 son las máscaras de roturas obtenidas por U-Net. Finalmente, la tercera columna es la superposición de las máscaras con las imágenes.

Dicho esto, es interesante explorar el comportamiento del modelo U-Net entrenado y validado

en este trabajo en una playa con diferentes condiciones. Para esto se utilizan imágenes completamente nuevas obtenidas mediante el Dron DJI Mavic Pro, en una playa cerca de Las Cruces, Chile ( $33^{\circ}29'46''\text{S}, 71^{\circ}37'39''\text{W}$ ). Esta playa se caracteriza por presentar una condición de oleaje energético. En la Figura 6.4 se puede ver que debido a la intensidad que produce la espuma remanente de las grandes olas en rotura presentes en la imágenes, la red logra la identificación de los patrones de rotura con un destacable desempeño. Al igual que en el caso presentado en esta tesis, el mayor déficit es cerca de la línea de costa, lo que posiblemente pueda mejorar si se intenta capturar las imágenes desde una mejor posición. Los resultados presentes en la Figura 6.4 se obtuvieron sin necesidad de entrenar nuevamente la red, solo se aplicó directamente sobre estos nuevos datos. Eso es una ventaja respecto con el método propuesto por Catalán et al. (2011) ya que este depende del ajuste de umbrales de intensidad si se quisiera utilizar el modelo en un nuevo conjunto de imágenes.

En términos de Machine Learning, U-Net obtuvo un destacable desempeño en un dominio con diferentes condiciones, por ende, la red nunca había visto imágenes con estas características. Este sencillo ejemplo, permite corroborar el ejemplo mencionado en la introducción del capítulo 4, donde una persona (o U-Net) puede identificar el fenómeno de rotura en diferentes playas con diferentes condiciones gracias al funcionamiento de la corteza visual. Además, con esto se puede argumentar que la red neuronal pudo aprender el fenómeno físico de rotura y no simplemente a agrupar píxeles con intensidad alta.

Por otro lado, se destaca que el método fue implementado en los servidores de Google Colab otorgando la posibilidad que una persona no necesite un computador con gran poder computacional, si no solo una conexión estable a internet. Además, con el uso de esta plataforma y específicamente con el entorno GPU, se ha logrado un tiempo de computo excepcional al realizar la predicción de patrones de rotura en 700 imágenes alrededor de 15s. El excelente tiempo obtenido en hacer esta predicción permite concluir que incluso se puede hacer esta predicción en tiempo real, abriendo la posibilidad incluso a estimar la disipación de energía producto de la rotura (Díaz et al., 2018), mejorando nuestro entendimiento de los diferentes procesos costeros.

## Capítulo 7

# Conclusión

Se presentó un novedoso e innovador método basado en el uso de U-Net, una red neuronal convolucional que permitió la identificación automática de patrones bidimensional de rotura del oleaje en la zona rompiente. Para esto, el modelo solo fue entrenado con imágenes en el rango electro-óptico y, dado que corresponde un modelo de aprendizaje supervisado, las máscaras de rotura fueron obtenidas mediante el trabajo propuesto por Catalán et al. (2011). Este trabajo se puede considerar como una versión mejorada del método mencionado, donde a través del uso de U-Net se pueden obtener máscaras de roturas mediante el uso de imágenes de video, sin necesidad de ajustar umbrales de intensidad e incluso permitiendo el uso del modelo en tiempo real con un excepcional tiempo de cómputo (15s en predecir 700 imágenes). Esto permite argumentar que se cumplió el objetivo general de la presente tesis.

Si bien las métricas expuestas en este trabajo muestran un 71% de precisión respecto a la metodología original, se demostró mediante un análisis cuantitativo y cualitativo que el desempeño de U-Net está por sobre el método de Catalán et al. (2011). Esto debido a que en muchos casos las máscaras consideradas como datos verdaderos (obtenidas con el método original) sobreestimaban o subestimaban la forma y tamaño de los patrones de rotura, mientras que las máscaras obtenidas por U-Net muestran patrones con formas físicamente posibles y pudiendo identificar olas rompientes que no fueron detectadas por la metodología original. El proceso de validación y las métricas obtenidas son concordante con los objetivos específicos mencionados en el capítulo 3.

Los resultados presentados en este trabajo dan pie a continuar esta línea de investigación, mediante la generalización del método propuesto a otras playas de interés. U-Net demostró un destacable rendimiento en imágenes completamente nuevas de un lugar con diferentes condiciones ambientales y de oleaje, como fue el caso de la playa cerca de Las Cruces en Chile presentada en el capítulo 6. De ser posible esto, se podrá mejorar el entendimiento acerca de este fenómeno físico y de los demás procesos que surgen a partir de la rotura del oleaje. Por ejemplo, mejorarían las metodologías existentes para estimar el nivel medio del mar (Flores et al., 2016), estimar las corrientes medias o corrientes de retorno Díaz et al. (2018) colocando a salvo la vida de los bañistas.



## Capítulo 8

# Trabajo Futuro

Esta tesis cuenta con un abanico de posibles aplicaciones para el estudio de los procesos costeros mencionados. Sin embargo, la red neuronal presentada solo permite una detección de patrones de rotura eficiente bajo una determinada configuración. Por esta razón, el primer y principal trabajo futuro es poder aplicar nuevas técnicas de Deep Learning para generar nuevas redes neuronales que permitan generalizar el proceso de rotura del oleaje en playas con diferentes características de oleaje y ambientales.

Dentro de las técnicas de Deep Learning se encuentra la transferencia de aprendizaje mediante fine-tuning, la cual consiste en utilizar los una determinada cantidad de parámetros de una red ya entrenada bajo una configuración definida (playa de Duck) para entrenar una nueva red neuronal con un nuevo conjunto de datos. Es importante mencionar que esta técnica es útil siempre y cuando los conjuntos de datos utilizados tengan una cierta relación entre ellos, esto se cumple en la idea presentada dado que se quiere representar el mismo fenómeno físico.

Además, se pueden utilizar otras técnicas complementarias para mejorar el proceso de aprendizaje de la nueva red neuronal. Por ejemplo, es posible aplicar una técnica de self-training para que la red aprenda de conjuntos de datos etiquetados y no etiquetados. Esto aumentaría la facilidad de implementar el nuevo método propuesto al solo utilizar un poco de imágenes etiquetadas (difíciles de obtener) y una gran cantidad de imágenes no etiquetadas (fáciles de obtener).



# Apéndice A

## Implementación del modelo U-Net

Una de nuestra ideas como investigadores es compartir estos resultados con toda la comunidad costera, es decir, generar una metodología de libre acceso para que cualquier persona interesada en replicar esta implementación o utilizar el modelo entrenado en este trabajo para realizar una predicción sobre su propio conjunto de datos. Para lograr esto, se creó un repositorio en la plataforma GitHub <sup>1</sup> en donde podrán encontrar los datos utilizados en este trabajo y todos los códigos necesarios para su implementación.

Si se desea replicar los resultados expuesto en este trabajo se debe descargar los datos <sup>2</sup>. Luego, se debe subir estos datos a Google Drive, dado que es recomendable utilizar Google Colab para la ejecución de este método por el alto costo computacional que requiere el entrenamiento de una red neuronal, utilizando en la mayoría de los casos GPU para acelerar este proceso. En caso que solo se desee hacer una predicción, es recomendable de todas formas utilizar Google Colab dado que se creó un archivo `.ipynb` para facilitar todos los procesos.

### A.1. Entrenar el modelo

En primer lugar se debe ajustar los parámetros de entrada de la celda *Folder's path* según el nombre de la carpeta que se utilizará. Además de la ejecución de la celda que cuenta con todos los paquetes necesarios.

#### A.1.1. Crear y cargar conjuntos de entrenamiento, validación y prueba.

```
1 from google.colab import drive
2 drive.mount('/content/gdrive')
3 %cd 'gdrive'
4 %cd 'My Drive/Nombre_de_Carpeta'
```

Posteriormente, se debe transformar las imágenes en matrices, para lo es necesario utilizar las tres primeras celdas de la sección *Inputs*

```
1 # Parameters
2 im_width = 512
```

<sup>1</sup>[https://github.com/fj23eslaonda/Wave\\_by\\_Wave\\_Identification](https://github.com/fj23eslaonda/Wave_by_Wave_Identification)

<sup>2</sup><https://data.mendeley.com/v1/datasets/jprw8tg2hh/draft?a=1cf01733-85b4-4132-9515-07db2e8caf62>

```

3 im_height = 512
4 paths     = {"im_tr": "trainset_duck/image/",
5             "la_tr": "trainset_duck/label/",
6             "im_tst": "testset_duck/image/",
7             "la_tst": "testset_duck/label/"}

1 # List of names all images
2 ids_xtr   = next(os.walk(paths["im_tr"]))[2]
3 ids_ytr   = next(os.walk(paths["la_tr"]))[2]
4 ids_xtst  = next(os.walk(paths["im_tst"]))[2]
5 ids_ytst  = next(os.walk(paths["la_tst"]))[2]
6
7 print("N of images in train set = ", len(ids_xtr), "\n")
8 print("N of labels in train set = ", len(ids_ytr), "\n")
9 print("N of images in test set = ", len(ids_xtst), "\n")
10 print("N of labels in test set = ", len(ids_ytst), "\n")

1
2 # Creating of matrix for train and test set
3
4 x_tr, y_tr = create_inputs(ids_xtr, paths["im_tr"],
5                           paths["la_tr"], im_height, im_width) # Train set
6 x_tst, y_tst = create_inputs(ids_xtst, paths["im_tst"],
7                              paths["la_tst"], im_height, im_width) # Test set
8
9 # Save inputs
10 np.save('Nombre_Carpeta_Matrices/x_tr_duck.npy', x_tr)
11 np.save('Nombre_Carpeta_Matrices/y_tr_duck.npy', y_tr)
12 np.save('Nombre_Carpeta_Matrices/x_tst_duck.npy', x_tst)
13 np.save('Nombre_Carpeta_Matrices/y_tst_duck.npy', y_tst)
    
```

### A.1.2. Entrenamiento de la red

Para lograr esto se debe ejecutar las celdas de la sección *U-Net - Training*.

```

1 # Load sets
2 x_tr   = np.load('Nombre_Carpeta_Matrices/x_tr_duck.npy')
3 y_tr   = np.load('Nombre_Carpeta_Matrices/y_tr_duck.npy')
4
5 print('\n Matrix loaded!')
6 print('=====')
7
8 # Split train and validation set
9 x_tr, x_val, y_tr, y_val = train_test_split(x_tr, y_tr, test_size = 0.1,
10                                             random_state=42)
11
12 print(" ")
13 print("Train set shape = ", x_tr.shape, "\n")
14 print("Validation set shape = ", x_val.shape, "\n")

1 # Input shape
2 input_img = Input((512, 512, 1), name = 'img')
3
4 # Call function and compile
5 model = get_unet(input_img,
6                 n_filters = 32,
7                 dropout   = 0.2,
    
```

```

8         batchnorm = True,
9         seed       = 0)
10
11 model.compile(optimizer = Adam(lr=1e-2),
12              loss      = 'binary_crossentropy',
13              metrics   = [f1])
14
15 # Callbacks to avoid overfitting
16 callbacks = [
17     EarlyStopping(patience = 10,
18                  verbose   = True),
19     ReduceLRonPlateau(factor   = 0.1,
20                       patience = 10,
21                       min_lr  = 0.00001,
22                       verbose  = 1),
23     ModelCheckpoint('best_model_final.h5',
24                    verbose    = True,
25                    save_best_only = True,
26                    save_weights_only = True)
27 ]
28
29 # Training
30 unet = model.fit(x          = x_tr,
31                y          = y_tr,
32                batch_size = 2,
33                epochs     = 50,
34                verbose    = True,
35                callbacks   = callbacks,
36                validation_data = (x_val, y_val))

```

## A.2. Predicción sobre nuevas imágenes

Dado que no se volverá a entrenar el modelo, solo debe cargar las imágenes sobre las cuales se desea hacer la predicción, transformarlas en el formato de entrada de la red, cargar el modelo y hacer la predicción. Para esto se deben ejecutar la siguientes celdas:

### A.2.1. Creación de matrices de predicción

```

1 # Parameters
2 im_width  = 512
3 im_height = 512
4 paths    = {"im_tst": "testset_newimage/"}
5 ids_xtst = next(os.walk(paths["im_tst"]))[2]
6
7 print("N of images in train set = ", len(ids_xtst), "\n")
8
9 # Create matrix
10 x_tst = np.zeros((len(ids_xtst), im_height, im_width, 1), dtype = np.float32)
11
12 for i, index in tqdm_notebook(enumerate(ids_xtst), total = len(ids_xtst)):
13
14     # load image and transform image to array
15     x_img = img_to_array(load_img(paths["im_tst"]+index, grayscale = True))
16

```

```

17 # Save images
18 x_tst[i] = x_img /255.0
19
20 np.save('Nombre_Carpeta_Matrices/x_tst_new.npy',x_tst)

```

### A.2.2. Cargar modelo y hacer predicción

```

1 # Load JSON and Create model
2 json_file = open('model_final.json', 'r')
3 loaded_model_json = json_file.read()
4 json_file.close()
5 model = model_from_json(loaded_model_json, {"tf":tf})
6
7 # Load weight
8 model.load_weights("best_model_final.h5")
9 print("Model Loaded!")

```

```

1 # Prediction on Test set
2 preds_tst = model.predict(x_tst, verbose = True)
3
4 # Binary mask
5 preds_tst_t = (preds_tst > 0.5).astype(np.uint8)
6
7 print('\n Prediction is ready!')
8 print('=====')

```

### A.3. Graficar los resultados

Finalmente, para cualquier de los dos casos se puede utilizar la función *plot\_prediction* para ver los resultados de las predicciones en términos de gráficos.

```

1 # Test
2 plot_prediction(x_tst, # video matrix
3                y_tst, # target or mask matrix
4                preds_tst_t, # binary mask matrix
5                index=299) # index

```

# Referencias

- Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 2481–2495. doi:10.1109/TPAMI.2016.2644615.
- den Bieman, J.P., de Ridder, M.P., van Gent, M.R., 2020. Deep learning video analysis as measurement technique in physical models. *Coastal Engineering* 158, 103689. URL: <https://doi.org/10.1016/j.coastaleng.2020.103689>, doi:10.1016/j.coastaleng.2020.103689.
- Brewster, B. C., Gould, R. E., Brander, R. W., 2019. Estimations of rip current rescues and drowning in the United States. *Natural Hazards and Earth System Sciences* 19, 389–397. URL: <https://nhess.copernicus.org/articles/19/389/2019/>, doi:10.5194/nhess-19-389-2019.
- Brighton, B., Sherker, S., Brander, R., Thompson, M., Bradstreet, A., 2013. Rip current related drowning deaths and rescues in Australia 2004–2011. *Natural Hazards and Earth System Sciences* 13, 1069–1075. URL: <https://nhess.copernicus.org/articles/13/1069/2013/>, doi:10.5194/nhess-13-1069-2013.
- Buscombe, D., Carini, R.J., 2019. A data-driven approach to classifying wave breaking in infrared imagery. *Remote Sensing* 11, 1–10. doi:10.3390/RS11070859.
- Buscombe, D., Carini, R.J., Harrison, S.R., Chickadel, C.C., Warrick, J.A., 2020. Optical wave gauging using deep neural networks. *Coastal Engineering* 155, 103593. doi:10.1016/j.coastaleng.2019.103593.
- Carini, R.J., Chickadel, C.C., Jessup, A.T., Thomson, J., 2015. Estimating wave energy dissipation in the surf zone using thermal infrared imagery. *Journal of Geophysical Research: Oceans* 120, 3937–3957. doi:10.1002/2014JC010561.
- Catalán, P.A., Haller, M.C., Holman, R.A., Plant, W.J., 2011. Optical and microwave detection of surf zone breaking waves. *IEEE Transactions on Geoscience and Remote Sensing* 49, 1879–1893. doi:10.1109/TGRS.2010.2095864.
- Catalán, P.A., Haller, M.C., Plant, W.J., 2014. Microwave backscattering from surf zone waves. *Journal of Geophysical Research* 119, 3098–3120. doi:10.1002/2014JC009880.
- Chollet, F., 2017. Xception: Deep Learning with Depthwise Separable Convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 1800–1807. URL: <http://ieeexplore.ieee.org/document/8099678/>, doi:10.1109/CVPR.2017.195.
- Díaz-Méndez, Guillermo M., Haller, Merrick C., Raubenheimer, Britt, Elgar, Steve, Elgar, Steve, 2015. Radar Remote Sensing Estimates of Waves and Wave Forcing at a Tidal Inlet. *Journal of Atmospheric and Oceanic Technology* 32, 842–854. doi:10.1175/JTECH-D-14-00215.1.

- Díaz, H., Catalán, P.A., Wilson, G.W., 2018. Quantification of Two-Dimensional Wave Breaking Dissipation in the Surf Zone from Remote Sensing Data. *Remote Sensing* 1038, 38. doi:10.3390/rs10010038.
- Dumoulin, V., Visin, F., 2016. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 .
- Duncan, J., 1981. An experimental investigation of breaking waves produced by a towed hydrofoil. *Proc. R. Soc. London A* 377, 331–348.
- Flores, R.P., Catalán, P., Haller, M.C., 2016. Estimating surfzone wave transformation and wave setup from remote sensing data. *Coastal Engineering* 114, 244–252. doi:10.1016/j.coastaleng.2016.04.008.
- Glorot, X., Bordes, A., Bengio, Y., 2011. Deep Sparse Rectifier Neural Networks, in: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323.
- Goutte, C., Gaussier, E., 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation, in: *Proceedings of the 27th European Conference on Advances in Information Retrieval Research*, Springer-Verlag, Berlin, Heidelberg. p. 345–359. URL: [https://doi.org/10.1007/978-3-540-31865-1\\_25](https://doi.org/10.1007/978-3-540-31865-1_25), doi:10.1007/978-3-540-31865-1\_25.
- Haller, M.C., Catalán, P.A., 2009. Remote sensing of wave roller lengths in the laboratory. *Journal of Geophysical Research* 114, C07022. doi:10.1029/2008JC005185.
- Haller, M.C., Honegger, D., Catalán, P.A., 2014. Rip current observations via marine radar. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 140, 115–124. doi:10.1061/(ASCE)WW.1943-5460.0000229.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem*, 770–778. doi:10.1109/CVPR.2016.90.
- Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 .
- Holman, R., Haller, M.C., 2013. Remote sensing of the nearshore. *Annual Review of Marine Science* 5, 95–113. doi:10.1146/annurev-marine-121211-172408.
- Holman, R., Stanley, J., 2007. The history and technical capabilities of Argus. *Coastal Engineering* 54, 477–491. doi:10.1016/j.coastaleng.2007.01.003.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861 .
- Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning, ICML 2015* 1, 448–456.
- Kim, J., Kim, J., Kim, T., Huh, D., Caires, S., 2020. Wave-Tracking in the Surf Zone Using Coastal Video Imagery with Deep Neural Networks. *Atmosphere* 11, 304. URL: <https://www.mdpi.com/2073-4433/11/3/304>, doi:10.3390/atmos11030304.

- Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings , 1–15.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25, 1097–1105. URL: <https://dl.acm.org/doi/10.1145/3065386>, doi:10.1145/3065386.
- Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324. URL: <http://ieeexplore.ieee.org/document/726791/>, doi:10.1109/5.726791.
- Moris, J.P., Catalán, P.A., Cienfuegos, R., 2021. Incorporating wave-breaking data in the calibration of a Boussinesq-type wave model. *Coastal Engineering* 168, 103945. doi:10.1016/j.coastaleng.2021.103945.
- Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 1137–1149. URL: <http://ieeexplore.ieee.org/document/7485869/>, doi:10.1109/TPAMI.2016.2577031.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Springer International Publishing, Cham. pp. 234–241. doi:10.1007/978-3-319-24574-4\_28.
- de Silva, A., Mori, I., Dusek, G., Davis, J., Pang, A., 2021. Automated rip current detection with region based convolutional neural networks. *Coastal Engineering* , 103859. doi:10.1016/j.coastaleng.2021.103859.
- Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* , 1–14.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 1929—1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>, doi:10.1016/0370-2693(93)90272-J.
- Stephens, S.A., Coco, G., Bryan, K.R., 2011. Numerical simulations of wave setup over barred beach profiles: Implications for predictability. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 137, 175–181. doi:10.1061/(asce)ww.1943-5460.0000076.
- Stringari, C.E., Harris, D.L., Power, H.E., 2019. A novel machine learning algorithm for tracking remotely sensed waves in the surf zone. *Coastal Engineering* 147, 149–158. doi:10.1016/j.coastaleng.2019.02.002.
- Stringari, C.E., Power, H.E., 2019. The fraction of broken waves in natural surf zones. *Journal of Geophysical Research: Oceans* 124, 9114–9140. doi:10.1029/2019JC015213.
- Stringari, C.E., Veras Guimarães, P., Filipot, J.F., Leckler, F., Duarte, R., 2021. Deep neural networks for active wave breaking classification. *Scientific Reports* 11, 3604. doi:10.1038/s41598-021-83188-y.

- Svendsen, Ib A., 2006. Introduction to nearshore hydrodynamics. World Scientific. 24, Advanced Series on Ocean Engineering. doi:<https://doi.org/10.1142/5740>.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A., 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, in: Proceedings of the AAAI Conference on Artificial Intelligence. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/11231>.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem, 2818–2826. doi:10.1109/CVPR.2016.308.
- Tan, M., Le, Q.V., 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 36th International Conference on Machine Learning, ICML 2019 2019-June, 10691–10700.