



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Universitat Politècnica de València  
Departamento de Sistemas Informáticos y Computación

# Redes neuronales para la clasificación y segmentación de imágenes médicas

TRABAJO FIN DE MÁSTER

Máster en inteligencia artificial, reconocimiento de formas e imagen digital

*Autor:* Verdeguer Gómez, Jorge

*Tutor:* Casacuberta Nolla, Francisco  
Gómez Adrián, Jon Ander  
Peris Abril, Álvaro

Curso 2019-2020



# Resumen

En este trabajo se va a abordar el problema de clasificación y segmentación de imagen médica mediante el uso de redes neuronales. En concreto, las imágenes han sido tomadas para casos de cáncer de piel. Una vez explorado el problema, compararemos la precisión de nuestras soluciones con otras propuestas que han abordado el mismo problema.

**Palabras clave:** Clasificación multiclase, Segmentación, Deep Learning, Convolucionales, Imagen médica

---

# Resum

En este treball anem a abordar el problema de classificació i segmentació d'imatge mèdica amb l'ús de xarxes neuronals. En concret, les imatges s'han pres per a casos de càncer de pell. Una volta explorat el problema, compararem la precisió de les nostres solucions amb les d'altres propostes que han abordat el mateix problema.

**Paraules clau:** Clasificació multiclase, Segmentació, Deep Learning, Convolucionals, Imatge mèdica

---

# Abstract

We are going to work in classification and segmentation of medical image using neural networks. Specifically, the images have been taken from skin cancer cases. Once we have explored the problem, we will compare the accuracy of our solutions with the solutions of other proposals that have worked in the same problem.

**Key words:** Multiclass Classification, Segmentation, Deep Learning, Convolutional, Medical image

---



# Índice general

---

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>VII</b>
<hr/>	
<b>Listings</b>	<b>1</b>
<b>1 Introducción</b>	<b>3</b>
1.1 Motivación . . . . .	3
1.2 Objetivos . . . . .	4
1.3 Estructura del documento . . . . .	4
<b>2 Revisión trabajos relacionados</b>	<b>5</b>
2.1 Clasificación de imagen médica . . . . .	5
2.2 Residual networks . . . . .	5
2.3 VGG16 . . . . .	6
2.4 AlexNet . . . . .	6
2.5 Wide ResNet . . . . .	6
2.6 Segmentación médica . . . . .	6
<b>3 Teoría de redes neuronales</b>	<b>9</b>
3.1 Redes Neuronales . . . . .	9
3.1.1 Redes Neuronales Densas . . . . .	10
3.1.2 Redes Convolucionales . . . . .	11
3.2 Arquitecturas . . . . .	12
3.2.1 Redes residuales (Residual Nets) . . . . .	12
3.2.2 Redes residuales anchas . . . . .	13
3.2.3 UNet . . . . .	13
3.3 Clasificación . . . . .	14
3.4 Segmentación de imagen . . . . .	14
<b>4 Marco Experimental</b>	<b>17</b>
4.1 Sets de datos . . . . .	17
4.1.1 ISIC 2018 . . . . .	17
4.1.2 ISIC 2019 . . . . .	18
4.2 Medidas de evaluación . . . . .	19
4.2.1 Balanced multi-class accuracy . . . . .	19
4.2.2 Índice de Jaccard . . . . .	20
4.3 Software utilizado . . . . .	20
4.3.1 Pandas . . . . .	20
4.3.2 PyTorch . . . . .	20
4.3.3 TensorBoard . . . . .	20
4.4 Hardware . . . . .	21
<b>5 Desarrollo</b>	<b>23</b>

---

5.1	Análisis previo	23
5.2	Clasificación estándar	23
5.3	Clasificación equilibrada	24
5.4	Validación cruzada	25
5.5	Segmentación	26
<b>6</b>	<b>Resultados</b>	<b>27</b>
6.1	Clasificación estándar	27
6.2	Clasificación equilibrada	28
6.3	Validación cruzada	28
6.4	Segmentación	29
6.5	Soluciones del estado del arte	29
<b>7</b>	<b>Conclusiones y trabajos futuros</b>	<b>33</b>
7.1	Conclusiones	33
7.2	Trabajos futuros	33
	<b>Bibliografía</b>	<b>35</b>

## Índice de figuras

---

3.1	Ilustración de una red densa. Las flechas simbolizan los pesos y los círculos las neuronas con sus valores. . . . .	10
3.2	Ejemplo de convolución . . . . .	11
3.3	Atajo de una red ResNet . . . . .	12
3.4	Estructura y atajos de una red UNet . . . . .	13
4.1	A la izquierda, podemos ver la imagen original. A la derecha, podemos ver la máscara real de la imagen original. . . . .	18
4.2	Imagen del set de datos ISIC 2019 etiquetada como Melanocytic nevus . . . . .	19
5.1	Tabla del uso de las particiones en cada etapa de la validación cruzada. . . . .	25
6.1	Ejemplo de una muestra correctamente segmentada por nuestro sistema, con un índice de Jaccard superior a 0.8 sobre 1. La imagen de la izquierda representa la imagen original, la imagen central representa la máscara original y la imagen de la derecha representa la máscara predicha por nuestro modelo . . . . .	29
6.2	Ejemplo de una muestra incorrectamente segmentada por nuestro sistema, con un índice de Jaccard inferior a 0.4 sobre 1. La imagen de la izquierda representa la imagen original, la imagen central representa la máscara original y la imagen de la derecha representa la máscara predicha por nuestro modelo . . . . .	30

## Índice de tablas

---

5.1	Distribución de las muestras del set de datos ISIC 2019 . . . . .	23
6.1	Resultados de los modelos en la prueba de clasificación estándar . . . . .	27
6.2	Resultados de los modelos en la prueba de clasificación equilibrada . . . . .	28
6.3	Resultados de la precisión de nuestros modelos junto a sus intervalos de confianza al 95% . . . . .	28
6.4	Resultados de la validación cruzada para la métrica <b>balanced multi-label accuracy</b> . . . . .	29

6.5	Resultados de nuestros modelos con los resultados de las propuestas para el set de datos ISIC 2019 . . . . .	30
6.6	Resultados para la segmentación, incluyendo nuestro modelo y las soluciones propuestas . . . . .	31

# Listings

---



---

---

# CAPÍTULO 1

## Introducción

---

Cáncer es el nombre que se le otorga a una colección de enfermedades con relación entre sí. En todos los tipos de cáncer algunas células se empiezan a reproducir de forma descontrolada sin parar y se expanden a los tejidos adyacentes. Este tipo de enfermedad puede empezar en casi cualquier parte del cuerpo humano. Normalmente, las células se reproducen para reemplazar células dañadas o viejas. No obstante, cuando se produce el cáncer, las células que deberían reemplazarse no mueren, y las células nuevas son capaces de reproducirse sin parar, creando conjuntos de células llamados tumores.

Estos tumores se les identifica como malignos cuando son capaces de expandirse o invadir tejidos adyacentes. Además, pueden propagarse a través de la sangre para crear tumores en sitios distintos del cuerpo, lejos del tumor original. Cabe destacar que también existen los llamados tumores benignos, que no invaden los tejidos cercanos y, aunque pueden alcanzar tamaños considerables, no suelen reaparecer cuando se extirpan.

En la actualidad existen muchos casos de cáncer que se investigan, y entre las maneras que conocemos de combatirlo, es fundamental una detección preventiva de la enfermedad para poder actuar antes de que sea demasiado tarde. En este proyecto, nos vamos a centrar exclusivamente en el cáncer de piel, que es el tipo de cáncer más común de todos. Aunque el cáncer de piel puede darse de muchas formas diferentes, cabe destacar que el más letal de los cánceres de piel es el causado por el conocido como melanoma, las células encargadas de producir la melanina.

### 1.1 Motivación

---

Nuestra motivación es dar forma a una herramienta que, a partir de una imagen de la piel nos pueda otorgar información sobre los posibles problemas que pudiese tener dicha piel. Esta herramienta nos permitiría dar lugar a una detección preventiva que no dependa de un doctor para tener una primera opinión. En este tipo de casos, la detección temprana de dichos problemas es fundamental para garantizar la seguridad del paciente.

Este caso nos parece interesante debido a que si bien no es posible predecir con total exactitud el carácter de la posible enfermedad a partir de una imagen,

si que nos puede ayudar a saber cuándo debemos pedir una segunda opinión a un experto. Este tipo de herramienta también se podría incluir en un sistema automatizado que puede hacer la revisión de un paciente en poco tiempo, sin la exploración exhaustiva de un doctor.

Por lo tanto, apuntaremos a crear un sistema capaz de etiquetar las imágenes médicas dependiendo de lo que suceda en la imagen de entrada.

## 1.2 Objetivos

---

Nuestros objetivos principales son los siguientes:

- Comprobar que el problema es abordable, es decir, nuestros modelos son capaces de extraer información relevante de las imágenes.
- Crear un sistema capaz de clasificar las imágenes en distintas etiquetas, correspondientes al problema que sufra la piel en la imagen.
- Comparar la eficacia entre distintos modelos para este problema de clasificación.
- Comparar nuestra eficacia con los modelos que ya han intentado abordar este problema.
- Garantizar una medida de precisión justa con **validación cruzada**

La dificultad principal de este problema, más allá de la dificultad intrínseca que puedan tener las propias imágenes a la hora de detectar los problemas, es que los datos que se nos han proporcionado tienen un número de muestras para cada etiqueta muy diferentes entre sí. Esto provoca que algunas etiquetas son menos representadas que otras en el clasificador.

## 1.3 Estructura del documento

---

Primero haremos una introducción al estado del arte respecto a la descripción de imágenes en el capítulo 2. En segundo lugar, explicaremos de las herramientas que hemos utilizado, con especial atención en las **redes neuronales**, en la capítulo 3. En tercer lugar, explicaremos el marco experimental del proyecto en la capítulo 4, el cual se centra en el set de datos, las medidas de evaluación, las arquitecturas utilizadas y el software que hemos usado. Más tarde, describiremos el procedimiento que hemos seguido en nuestros experimentos y los modelos que hemos utilizado en el capítulo 5. Seguiremos con los resultados que hemos obtenido en cada uno de nuestros experimentos en el capítulo 6 y por último presentaremos las conclusiones de los resultados alcanzados, junto a los trabajos futuros en los que nos gustaría trabajar después de este proyecto en el capítulo 7.1.

---

# CAPÍTULO 2

## Revisión trabajos relacionados

---

En este capítulo haremos una recapitulación de que técnicas y modelos se usan para trabajar con clasificación y, mas concretamente, en la clasificación de imagen médica. También haremos una breve introducción a los modelos más eficaces actualmente, los cuales hemos usado para abordar nuestro problema.

### 2.1 Clasificación de imagen médica

---

La clasificación de imagen médica ya esta siendo abordada principalmente mediante redes neuronales convolucionales en la actualidad. En publicaciones como [1] ya se ha explorado este tipo de problema mediante el uso del deep learning.

Las pruebas de este modelo se hicieron principalmente sobre la tarea de clasificación del set de datos **ILD** [2]. Este set de datos consiste en una colección de casos con dolencias que afectan al pulmón. Posee una gran cantidad de imágenes de tomografías de alta resolución del pulmón. Estas imágenes vienen etiquetadas por regiones con los parámetros que los médicos han estimado convenientes, para proceder a su predicción.

### 2.2 Residual networks

---

Las residual networks, comunmente abreviadas como ResNet, son un tipo de red muy utilizado en la actualidad. Pese a hacer su aparición en 2015 [3], son un tipo de red que se sigue utilizando en algunos casos, debido a su eficacia en la actualidad. La potencia de estas redes reside en las conexiones residuales que utiliza, las cuales por su topologia permiten una retropropagación más eficaz del gradiente para así conseguir que dicho gradiente no se desvanezca en una red muy profunda.

Para la demostración de este modelo, se ha usado el set de datos de **ImageNet** [4], un set de datos muy utilizado para probar la eficacia de nuevos modelos. Este set de datos consiste en una tarea de clasificación de imágenes las cuales poseen una etiqueta entre 1000 clases distintas. Este set de datos tiene el propósito de crecer hasta los 50 millones de imágenes de alta resolución perfectamente eti-

quetadas. Aún así, lo común es no utilizar todo el set de datos, debido a su gran tamaño. Por ejemplo, la publicación en la que se presentan las ResNet hace uso de 140k imágenes para su entrenamiento, su validación y su test.

## 2.3 VGG16

---

La VGG16 es un tipo de red convolucional que se usó por primera vez para abordar el conjunto de ImageNet en la publicación [5]. Esta red intentaba evaluar la capacidad de análisis de las redes convolucionales usando solo filtros convolucionales de 3x3. Los resultados fueron muy satisfactorios para esta tarea, por lo que hemos elegido el modelo para abordar nuestro problema de clasificación.

## 2.4 AlexNet

---

En 2017, la AlexNet [6] consiguió mejorar los resultados obtenidos por el state-of-the-art en el problema de ImageNet. Esta red consiste en un conjunto de redes convolucionales, capas de max-pooling, capas de dropout y un conjunto de capas densas al final para realizar la clasificación pertinente. La razón para la creación de un modelo tan grande y con tanta generalización (otorgada principalmente por las capas de dropout) es el objetivo de poder generalizar lo suficiente la entrada como para abordar el problema de la variabilidad que presentan naturalmente las imágenes.

## 2.5 Wide ResNet

---

La red Wide ResNet es una red convolucional con una propuesta diferente a las anteriores [7]. Mientras que estas redes se basan en aumentar el número de capas convolucionales para tener una mayor cantidad de parámetros, los creadores de la Wide ResNet propusieron un progreso diferente. Para aumentar el número de parámetros hicieron que cada capa convolucional tuviera una cantidad de filtros mayor, para en cada nivel de la red convolucional tener mas parámetros que pueden aprender. Este tipo de enfoque permite evitar el problema del gradiente que se desvanece, ya que no son tan profundas como las otras propuestas, pero llegan a un número igual o mayor de parámetros. Los resultados que se obtuvieron con esta red en el problema de ImageNet han demostrado que esta red podría otorgar los resultados que buscamos en nuestro problema de clasificación.

## 2.6 Segmentación médica

---

Para el problema de segmentación hemos decidido usar el modelo UNet. Este modelo ya se presentó en su día para resolver un problema de segmentación de imagen médica en el artículo[8]. Su estructura ha permitido facilitar la segmentación para otorgar unos resultados adecuados para la tarea. Hemos elegido este

---

tipo de red debido a su gran éxito en los problemas de segmentación de imagen médica. Ejemplos de estos casos se pueden ver en artículos como [9] y [10].



---

---

# CAPÍTULO 3

## Teoría de redes neuronales

---

En este capítulo haremos una introducción a las redes neuronales y a sus diferentes tipos. También describiremos las arquitecturas más comunes en este ámbito que hemos usado.

### 3.1 Redes Neuronales

---

Una red neuronal consiste principalmente en muchas unidades de proceso muy simples llamadas neuronas. Las neuronas no se usan de forma individual, sino que se agrupan en capas, que pueden ser de diferentes tipos. Estas neuronas poseen un valor real al que llamaremos peso por cada conexión con la capa anterior, el cual se usa para procesar un valor de entrada. Dichas operaciones son diferentes según el tipo de capa se utilice. Dependiendo de las conexiones de cada capa, se pueden clasificar en tres tipos diferentes:

- Capas de entrada: Las capas de entrada obtienen la información directamente de algún tipo de datos, no de otra capa de la red. Son la primera capa por la que pasa la información en una red neuronal.
- Capas de salida: Las capas de salida son las que proporcionan la información de salida de la red neuronal. Para encontrar dicha salida de la red neuronal, se extraen los valores procesados en las neuronas de las capas de salida.
- Capas ocultas: Las capas ocultas son las capas que no son de entrada ni de salida, es decir, obtienen su información de entrada de otras capas y una vez la procesan, otras capas la extraen para procesarla.

Para conseguir que la red otorgue los resultados deseados, es necesario que las neuronas tengan los pesos adecuados para la tarea. Esto se consigue mediante el entrenamiento, el cual puede ser de dos tipos: supervisado y no supervisado.

El aprendizaje supervisado es el más común de los dos, se posee un set de datos que consiste en los datos que vamos a suministrar a la red y la salida que esperamos obtener de los mismos. Los datos, sean imágenes, secuencias o cualquier otro tipo de datos se suministran a la red, la cual procesará dichos datos de

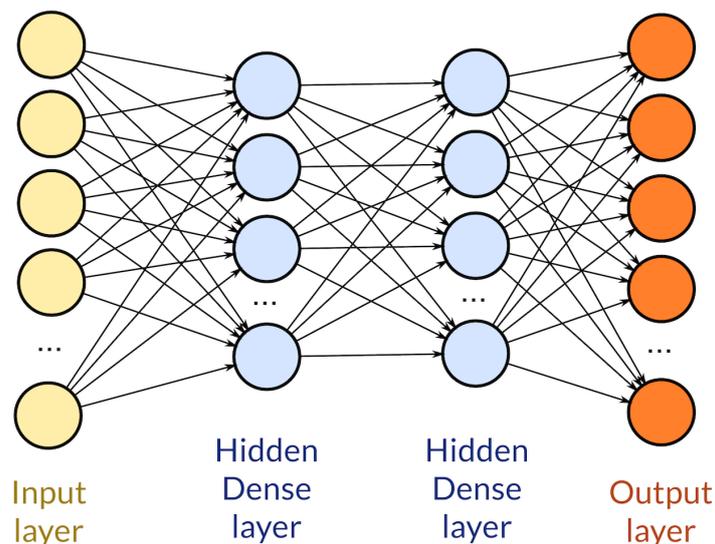
forma individual. Tras procesar los datos obtendremos una salida, la cual compararemos con la salida que esperábamos del modelo. Esta comparación se puede hacer de muchas formas diferentes, las cuales se denominan funciones de pérdida. Dependiendo de la tarea, será más beneficioso gastar una u otra. Una vez se hace dicha comparación, se obtiene la pérdida en las capas de salida, las cuáles se propagan por el modelo mediante derivaciones de las funciones de activación. Esta pérdida se usa para actualizar los pesos de la red, para que así el resultado de procesar la entrada se acerque cada vez más al objetivo. Si el sistema está bien construido y es capaz de resolver el problema, eventualmente convergerá, dando lugar a un modelo capaz de producir la salida esperada para la entrada del modelo.

El aprendizaje no supervisado no suele usarse para el aprendizaje, sino como una forma de inicializar los pesos para lograr la convergencia de un modelo que usa el aprendizaje supervisado de una forma más rápida. Algunos ejemplos incluyen la estructura del Auto Encoder, la cual comentaremos más adelante.

A continuación vamos a comentar los distintos tipos de capas

### 3.1.1. Redes Neuronales Densas

Las redes densas son las capas más simples de las redes neuronales. En estas capas, las neuronas están conectadas a cada neurona de la capa anterior. Para calcular el valor de salida, se multiplica cada valor de salida de la capa anterior por el peso de la conexión con cada neurona y se suman, junto con un bias. Estas capas operan normalmente sobre un vector de entrada unidimensional. Un ejemplo visual de esto se puede ver en la figura 3.1.



**Figura 3.1:** Ilustración de una red densa. Las flechas simbolizan los pesos y los círculos las neuronas con sus valores.

Actualmente, cuando se trata de trabajar con imágenes, las redes densas no suelen tomar la tarea de extraer las características de la misma. En cambio, cuando otro tipo de red se encarga de extraer las características de la imagen, las capas

densas se encargan de analizar dichas características, normalmente en forma de un vector unidimensional, para clasificar la imagen original en diferentes clases.

Formalmente, las redes densas están formadas por un conjunto de pesos  $W \in R^d$  con longitud  $n \times m$  formado por  $W = \{w_{1,1}, w_{2,1}, \dots, w_{n,1}, w_{1,2}, \dots, w_{n,m}\}$  y con un conjunto de neuronas de salida  $J$  de longitud  $m$  donde  $J = \{j_1, j_2, \dots, j_m\}$ . Considerando una entrada  $x \in R^d$  con una longitud  $n$  donde  $x = \{x_1, x_2, \dots, x_n\}$  la operación que ejecuta la capa densa sobre la entrada sería

$$s_j = \sum_i w_{i,j} \cdot x_i, 1 \leq j \leq m$$

### 3.1.2. Redes Convolucionales

Las redes convolucionales son un tipo de red que ha demostrado ser muy eficiente a la hora de procesar imágenes. Las capas convolucionales procesan la imagen de entrada mediante una ventana con tamaño fijo, que va desplazándose hasta extraer valores para toda la imagen. Para extraer los valores, se multiplica el valor de los píxeles de la imagen por los pesos de dicha ventana. Una visualización de este proceso se puede observar en la figura 3.2.

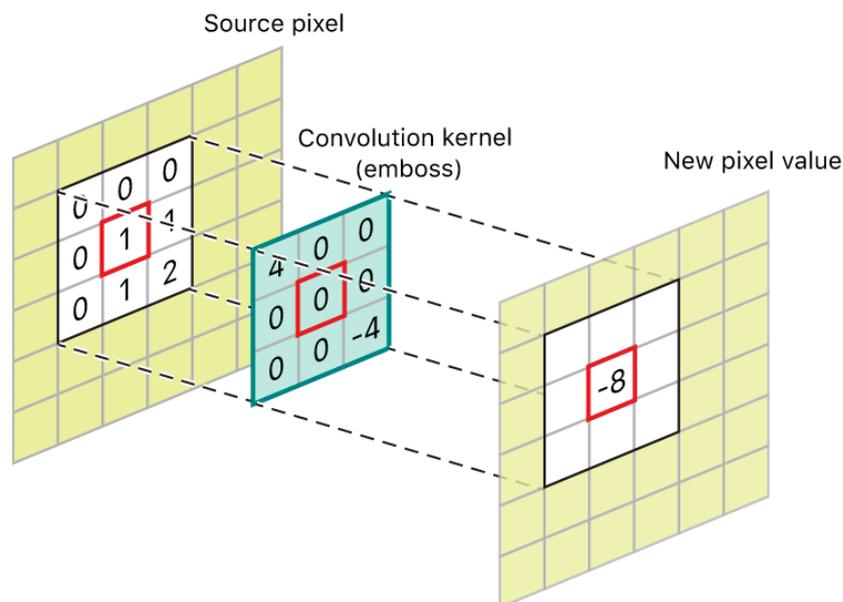


Figura 3.2: Ejemplo de convolución

Las capas convolucionales poseen un número de filtros de salida. Estos filtros son la cantidad de imágenes que estarán en el vector de salida tras procesar la imagen de entrada. La metodología para formar cada imagen consiste en que la capa convolucional tiene por cada imagen de salida un conjunto de pesos distinto para la ventana. Esto hace capaz a la capa convolucional de extraer varias imágenes diferentes, conocidos como filtros, de la imagen de entrada.

## 3.2 Arquitecturas

En esta sección comentaremos las diferentes estructuras que se han usado a lo largo de nuestros experimentos.

### 3.2.1. Redes residuales (Residual Nets)

Las redes residuales son en esencia redes convolucionales, con la principal diferencia de que hay atajos entre diferentes capas, como se puede visualizar en la figura 3.3. Estos atajos permiten combatir los problemas que provocan tener una red demasiado profunda. Cuando una red es demasiado profunda, el gradiente se va desvaneciendo capa a capa hasta que las capas cerca de la entrada no logran aprender nada. Estos atajos permiten que el gradiente se propague hasta las primeras capas, facilitando su conservación.

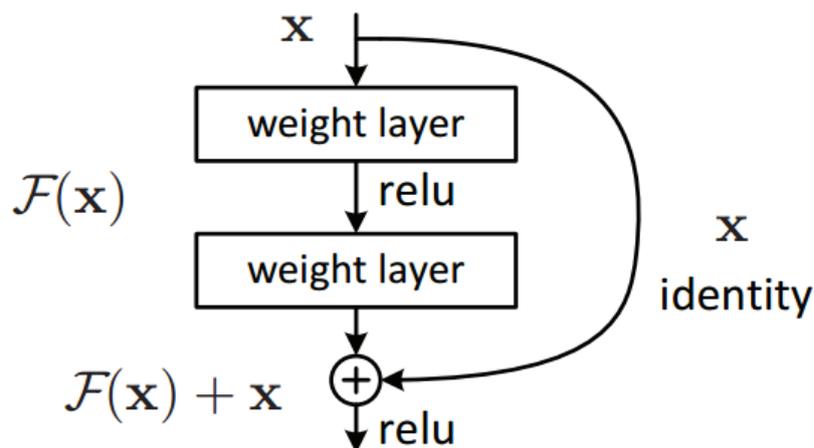


Figura 3.3: Atajo de una red ResNet

En nuestro proyecto vamos a usar dos tipos de residual nets. En primer lugar, hemos utilizado un modelo hecho por nosotros mismos. Este modelo consiste de un conjunto de bloques que consisten en capas convolucionales, capas de pooling y capas de dropout. Por último, este modelo llega a un conjunto de capas densas, las cuales se encargan de la clasificación de la imagen. Por otra parte, hemos decidido usar también la red ResNet101, la cual es una red residual que se ha preentrenado en el problema de ImageNet.

La razón de usar una red preentrenada es que aunque se haya entrenado para un problema diferente, una red preentrenada facilita la convergencia en un problema adyacente, al otorgar unos pesos iniciales más productivos que unos pesos aleatorios. Tras la creación de este modelo, los pesos se reajustan utilizando el problema en cuestión para que tengan mejores resultados respecto a este nuevo problema que estamos abordando.

### 3.2.2. Redes residuales anchas

Para abordar el problema de clasificación también hemos utilizado una wide resnet. Este tipo de red es similar a las residual net, pero en vez de explotar la profundidad de la red gracias a las capas residuales, lo que hacen es expandir las capas convolucionales horizontalmente (Es decir, aumentando el número de filtros de cada capa convolucional). Aunque la explicación completa se puede encontrar en [7]. Para abordar este problema hemos hecho uso de una Wide ResNet preentrenada, para facilitar la convergencia, como se ha explicado en la sección 3.2.1.

### 3.2.3. UNet

La red UNet es una red que posee una arquitectura peculiar dentro de la familia de las redes convolucionales. Esta red tiene una estructura encoder-decoder en el sentido de que esta formada por dos partes. En la primera parte, la imagen es reducida a través de los filtros convolucionales para extraer las características. En la segunda parte, este vector de características sigue el camino contrario, aumentando sus dimensiones hasta el tamaño de la imagen original, donde a la salida se encuentra la máscara creada. La diferencia con la estructura tradicional de encoder-decoder es que hay conexiones que copian los datos que se están procesando en el encoder al decoder para facilitar la generación de la máscara. Estas conexiones también ayudan a la propagación del gradiente, evitando que se desvanezca al pasar por toda la red para llegar al inicio. Para poder ver estas conexiones de forma clara, añadimos la figura 3.4

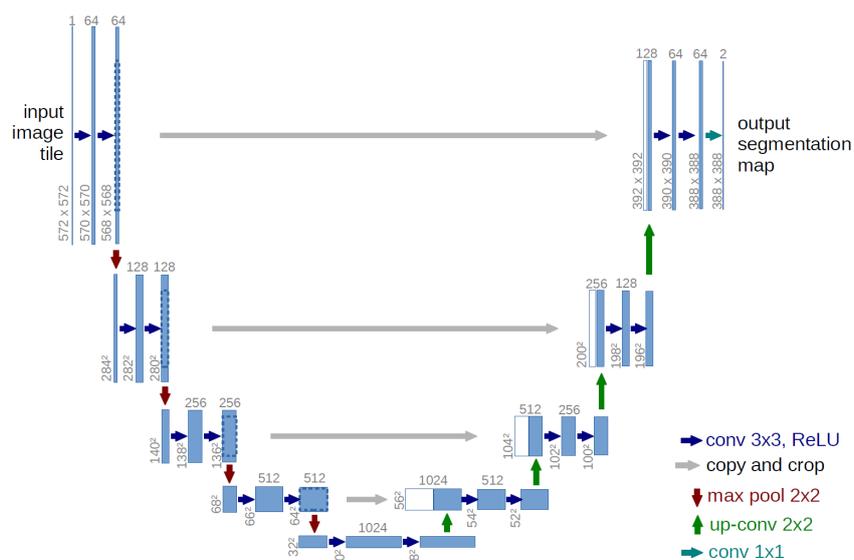


Figura 3.4: Estructura y atajos de una red UNet

### 3.3 Clasificación

---

El problema de clasificación consiste en recibir unos datos de entrada, en nuestro caso una imagen, y asignarle una de las clases que presenta el problema. Para el entrenamiento y la validación, dichas imágenes están emparejadas con su etiqueta correspondiente. Estas etiquetas representan la clase de las imágenes que hemos de procesar. La metodología habitual para otorgar la clasificación en una clase a partir de una imagen es la siguiente:

1. Las imágenes se preprocesan para que tengan un formato similar, en caso que sean diferentes en tamaño.
2. Estas imágenes son analizadas mediante una red convolucional, la cual extrae las características y forma un vector que representa la imagen
3. Dicho vector se pasa por un conjunto de capas densas, las cuales analizan todo este vector.
4. Tras pasar por las capas densas, los pesos resultantes llegarán a una capa Softmax, la cual actuará como salida
5. Para asignar la etiqueta correspondiente a la muestra tras haberlo procesado con el modelo, elegimos la etiqueta con mayor probabilidad, ya que la salida de la softmax tiene esta propiedad

En nuestro problema, tenemos un conjunto de imágenes  $X$  y un conjunto de clases posibles  $C$  donde  $C = \{0, 1, 2, 3, 4, 5, 6, 7\}$  debido a que tenemos ocho clases diferentes posibles. Formalmente, definimos nuestro problema de clasificación como un conjunto de imágenes donde para cada imagen  $x$  se cumple que esta imagen pertenece al set de datos  $X$ , es decir,  $x \in X$ . A cada imagen  $x$  le corresponde una clase  $c$ , que pertenece al conjunto de clases  $C$ , es decir,  $c \in C$ .

En nuestro caso, poseemos un clasificador  $G$ , el cual nos permite obtener las probabilidades de que la muestra  $x$  pertenezca a la clase  $c$  mediante la función  $g_c(x)$ . Por tanto, elegiremos la etiqueta escogida por el clasificador, denominada  $\hat{c}$ , por la fórmula 3.1

$$\hat{c} = G(x) \equiv \arg \max_{c \in C} g_c(x) \quad (3.3.1)$$

3.1: Fórmula para denominar la clase escogida por el clasificador  $G$  sobre la muestra  $X$

### 3.4 Segmentación de imagen

---

El problema de segmentación de imagen consiste en, a partir de una imagen, crear una o más mascarar sobre la misma imagen. Esto tiene muchas aplicaciones en imagen médica, ya que puede permitir etiquetar automáticamente órganos con una radiografía o delimitar la zona a la cual afecta un tumor, por ejemplo.

En nuestro caso, el objetivo es crear una máscara que incluya el tumor que se ha detectado. Para ello, se suele seguir este procedimiento:

1. Las imágenes se preprocesan para que tengan un formato similar, en caso que sean diferentes en tamaño.
2. Estas imágenes son analizadas mediante una red convolucional, la cual extrae las características y forma un vector que representa la imagen
3. Dicho vector se pasa por un conjunto de capas densas, las cuales analizan todo este vector.
4. Una vez analizado el vector, se empieza a reconstruir la imagen original
5. Cuando la reconstrucción llega al tamaño original se busca que en vez de llegar a los valores de la imagen original, se cree la máscara que buscamos como etiqueta. Para ello, tendremos un número de filtros de salida equivalente al número de posibles máscaras y el valor en cada píxel representa la probabilidad de que dicho píxel pertenezca a esa máscara.



---

---

# CAPÍTULO 4

## Marco Experimental

---

En este capítulo vamos a hacer un análisis de todas las herramientas y sistemas que hemos usado para poner en marcha nuestros experimentos, junto con un análisis de los conjuntos de datos que hemos usado.

### 4.1 Sets de datos

---

En esta sección, analizaremos los dos sets de datos que hemos utilizado en nuestros experimentos.

#### 4.1.1. ISIC 2018

Este set de datos [11] consta de 2594 imágenes con sus correspondientes 2594 máscaras reales. Las imágenes han sido adquiridas de una variedad de tipos de dermatoscopios, de todos los sitios anatómicos excluyendo mucosas y uñas, de una muestra histórica de pacientes presentados para el cribado de cáncer de piel. Cada imagen contiene exactamente una lesión primaria resaltada por la máscara, mientras que las lesiones secundarias o regiones pigmentadas pueden pasarse por alto. La distribución de estas muestras no representa el mismo caso que en la vida real, ya que en estas muestras hay una representación mayor de las lesiones malignas.

Las máscaras consisten en máscaras binarias en formato PNG. Estas máscaras indican la posición que ocupa la lesión primaria en cada imagen de entrada. Las máscaras se han codificado como imágenes de un solo canal, donde cada pixel tiene:

- Valor 255, en el caso de que represente la posición de la lesión principal
- Valor 0, en el caso de que represente el fondo de la imagen o una lesión secundaria

Como la lesión primaria en estas imágenes es una sola region contigua, las máscaras también deberían estar formadas por una única region contigua, sin ningún vacío interior o secciones desconectadas.



**Figura 4.1:** A la izquierda, podemos ver la imagen original. A la derecha, podemos ver la máscara real de la imagen original.

Las máscaras de cada imagen que se proporcionan como verdaderas han sido generadas a partir de varias técnicas, pero todas las máscaras han sido revisadas por expertos en dermatología, para asegurar su validez. Estas máscaras han sido generadas por cualquiera de estos métodos:

- Un algoritmo automatizado, revisada por un experto
- Un algoritmo de relleno semiautomatizado, con parámetros elegidos por un experto
- Trazado de polígonos manual por parte de un experto

Estas imágenes las hemos reescalado al tamaño 768x1024, tanto estas imágenes como sus máscaras. Esto es para darles a todas un tamaño idéntico y así poder procesarlas con la misma red. Podemos observar un ejemplo de una muestra con su máscara en la figura 4.1.

Para la evaluación de este sistema, se usará la métrica del índice de Jaccard 4.2.2.

#### 4.1.2. ISIC 2019

Este set de datos consiste en 25331 imágenes de cáncer de piel. Estas imágenes vienen etiquetadas con su correspondiente categoría. Las categorías posibles son las siguientes:

1. Melanoma
2. Melanocytic nevus
3. Basal cell carcinoma
4. Actinic keratosis
5. Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis)
6. Dermatofibroma
7. Vascular lesion

## 8. Squamous cell carcinoma

Cabe destacar que las imágenes de este set de datos no están equilibradas para todas las clases posibles, es decir, hay un número diferente de imágenes para cada una de las etiquetas. Para procesar estas imágenes hemos hecho un reescalado a 256x256 píxeles, para que todos los datos de entrada tengan las mismas dimensiones. Podemos ver un ejemplo de este tipo de imagen en la figura 4.2.

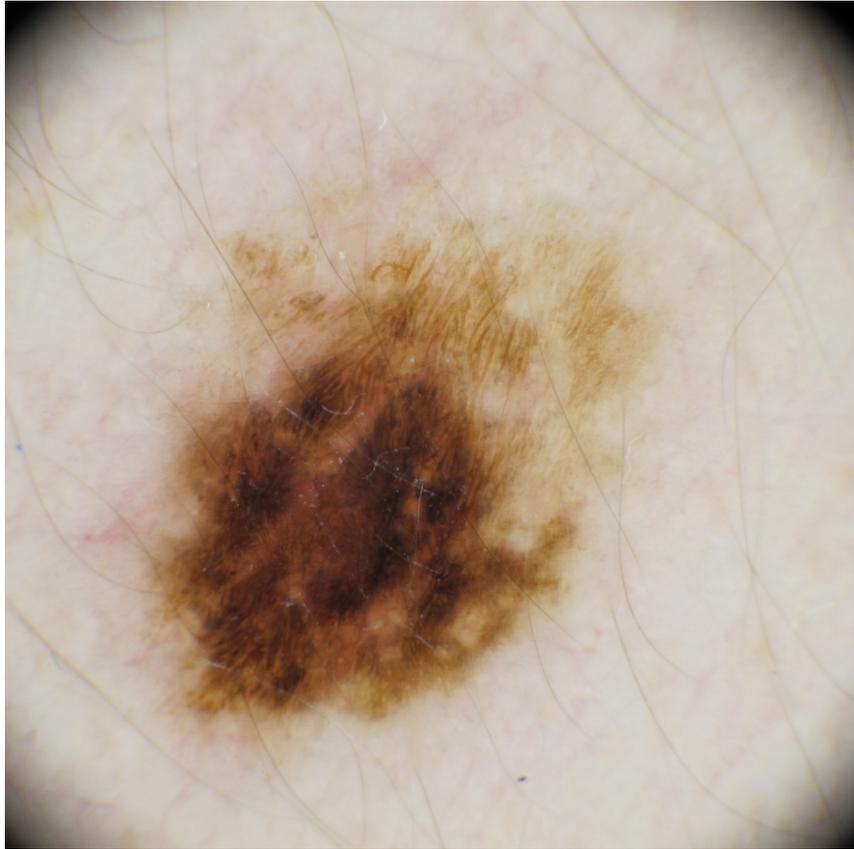


Figura 4.2: Imagen del set de datos ISIC 2019 etiquetada como Melanocytic nevus

## 4.2 Medidas de evaluación

---

En este apartado comentaremos las medidas de evaluación que hemos usado para calificar la eficacia de nuestro sistema.

### 4.2.1. **Balanced multi-class accuracy**

La métrica que se usará para evaluar los modelos de clasificación es la **balanced multi-class accuracy**. El motivo de utilizar esta medida es que para evaluar un problema de clasificación con muestras desequilibradas para cada clase no es suficiente utilizar la precisión global. Usar una métrica como la precisión global puede dar lugar a que un clasificador se centre solo en las clases más frecuentes e ignore las menos probables. Por ello, para asegurar que esto no sucede en este problema, utilizaremos la **balanced multi-class accuracy**. Esta métrica se puede

calcular con la fórmula 4.2.1, donde  $X_c$  es el número de muestras de la clase  $c$  y  $CorrectX_c$  es el número de muestras clasificadas correctamente en la clase  $c$ .

$$\sum_{c=1}^C \frac{X_c}{CorrectX_c} \quad (4.2.1)$$

## 4.2.2. Índice de Jaccard

La métrica que se usará para evaluar los modelos de segmentación es el **índice de Jaccard**. Esta métrica se puede calcular con la fórmula 4.2.2 donde A es el area que ha resaltado nuestro clasificador y B es el area de la imagen resaltada por la etiqueta original. J simboliza el valor de la métrica.

$$J = \frac{|A \cap B|}{|A \cup B|} \quad (4.2.2)$$

## 4.3 Software utilizado

---

En esta sección vamos a introducir brevemente las diferentes tecnologías que se han utilizado para el desarrollo de nuestro experimento. Hemos utilizado Python como lenguaje de programación a lo largo de todo el experimento.

### 4.3.1. Pandas

Pandas es una librería de Python que se utiliza para manipular datos y hacer análisis de forma sencilla. En nuestro caso, hemos utilizado la librería para hacer uso del archivo CSV directamente desde Python, facilitando así su procesado y extrayendo la información requerida en el script.

### 4.3.2. PyTorch

PyTorch es la librería que hemos utilizado para todos los temas relacionados con el Deep Learning. Es la librería con la que hemos creado los modelos, hemos hecho data augmentation y el preprocesado, y hemos realizado el entrenamiento. También ha sido de gran ayuda a la hora de analizar el rendimiento, ya que otras librerías de las que hablaremos a continuación funcionan directamente sobre PyTorch

### 4.3.3. TensorBoard

TensorBoard es una librería que nos permite analizar los datos de forma elegante y en tiempo real. Normalmente se usa para evaluar factores como la pérdida, la precisión o la velocidad del entrenamiento, pero se puede usar para hacer

gráficos en tiempo real de cualquier dato al que tengas acceso. En nuestro caso, lo hemos utilizado para tener una idea del progreso de nuestro modelo en tiempo real.

## 4.4 Hardware

---

Para el entrenamiento de los modelos, se han utilizado varias máquinas pertenecientes al PRHLT con tarjetas gráficas. Para nuestros experimentos, hemos utilizado principalmente tarjetas GTX 1080.



---

---

# CAPÍTULO 5

## Desarrollo

---

En este capítulo vamos a comentar los pasos que hemos seguido para abordar los problemas que hemos presentado previamente.

### 5.1 Análisis previo

---

En primer lugar, hemos realizado un análisis previo de los sets de datos para los problemas a resolver. En el caso de clasificación, con ISIC 2019, hemos descubierto que las clases no están representadas de una forma equilibrada. La distribución de dichas clases se puede apreciar en la Tabla 5.1. Como podemos ver en la tabla, la cantidad de muestras de cada clase es muy irregular, lo que provocará problemas en el aprendizaje de la red. Esto es debido a que las clases con más muestras serán más representadas en el sistema y la red se afinará más para detectar estas muestras que el resto.

A continuación exploraremos las diferentes soluciones que hemos proporcionado a este problema.

### 5.2 Clasificación estándar

---

En esta prueba, hemos intentado abordar el problema sin modificar el número de muestras. Hemos dividido el set de datos en tres conjuntos, el conjunto de

Etiqueta	Número de muestras
Melanoma	4522
Melanocytic nevus	12875
Basal cell carcinoma	3323
Actinic keratosis	867
Benign keratosis	2624
Dermatofibroma	239
Vascular lesion	253
Squamous cell carcinoma	628

**Tabla 5.1:** Distribución de las muestras del set de datos ISIC 2019

entrenamiento, el conjunto de validación y el conjunto de test. El conjunto de entrenamiento está formado por el 70 % de las muestras del set de datos, el conjunto de validación está formado por el 10 % de muestras del set de datos y el conjunto de test está formado por el 20 % de muestras del set de datos.

Una vez tenemos el set de datos dividido en tres conjuntos, hemos procedido al entrenamiento de las redes. Para escoger las versiones de los modelos que guardamos, hemos seguido los siguientes pasos:

1. Se hace una epoch de entrenamiento, es decir, se procesan todas las muestras del conjunto de entrenamiento por el modelo y se hace uso de la función de pérdida y del optimizador para conseguir los pesos adecuados.
2. Se procesa el conjunto de validación, sin propagar los gradientes ni entrenar el modelo.
3. Se analizan las métricas conseguidas sobre el conjunto de validación. Guardaremos el modelo si estas métricas són:
  - La menor pérdida generada por la función de pérdida.
  - La mayor precisión.
  - La mayor **balanced multi-class accuracy**.
4. Se repite este ciclo hasta que se acaban las epochs, en nuestro caso, 100 epochs por modelo

Al final de este proceso, obtenemos 3 versiones diferentes de nuestro modelo, uno por cada métrica que hemos medido.

## 5.3 Clasificación equilibrada

---

En esta prueba, hemos intentado modificar el número de muestras de cada clase para que sea viable. Por tanto, hemos decidido escoger el mismo número de muestras para cada clase. No obstante, la cantidad de muestras de las clases menos pobladas es muy pequeña, por lo que no podemos simplemente entrenar con un número de muestras de cada clase equivalente a la clase con el menor número de muestras. Por lo tanto, para realizar el entrenamiento de este experimento hemos decidido agrupar las cuatro clases menos comunes en una sola clase, para poder coger un número de muestras aceptable en el entrenamiento. Este proceso ha reducido el número de posibles clases de 8 a 5. Hemos escogido 1987 muestras de cada clase para realizar nuestro experimento.

Para entrenar este experimento hemos seguido los mismos pasos que se han explicado en el experimento de clasificación estándar [5.2](#).

A la hora de probar la **balanced multi-class accuracy** de este modelo, hemos de tener en cuenta de que esta métrica ha sido calculada sobre un conjunto de test de 5 etiquetas, lo cual reduciría el valor de esta métrica en el problema original. Si escogiesemos la clase más probable de las cuatro que hemos reducido para todas las muestras clasificadas en esta etiqueta, entonces el valor real de la métrica

**balanced multi-class accuracy** para este experimento sería  $M_r = \frac{5}{8}M_b$ , donde  $M_r$  representa el valor real de la métrica y  $M_b$  es la métrica de **balanced multi-class accuracy** calculada sobre el conjunto de test.

## 5.4 Validación cruzada

Los datos que hemos utilizado para nuestras pruebas de clasificación son las muestras que se otorgaron para el entrenamiento y la validación de los modelos en la competición. No obstante, el conjunto de test de la competición no viene con las etiquetas correspondientes de las muestras. Es por ello que para compararnos con los resultados de la competición hemos utilizado validación cruzada. De estas maneras, creemos que los resultados de los modelos serán más representativos del caso real.

Para realizar la validación cruzada hemos seguido los siguientes pasos:

1. Hemos decidido hacer validación cruzada en 5 particiones.
2. En cada una de las 5 etapas, se usarán 4 particiones como entrenamiento y 1 como test.
3. Cada etapa tendrá una partición de test distinta, para que todas las particiones actúen como test. Esto se puede visualizar en la Figura 5.1.
4. Para entrenar, un 12.5% de las particiones de entrenamiento se ha elegido como conjunto de validación y el resto como conjunto de entrenamiento.
5. Se repite el proceso de entrenamiento que hemos visto en el capítulo de clasificación 5.2.
6. Una vez el modelo está entrenado, los modelos resultantes se prueban sobre el conjunto que ha quedado marcado como conjunto de test.
7. Cuando hemos hecho esto para todas las etapas, sacamos la media aritmética de la métrica de evaluación en todas las etapas. Esta media será la métrica real obtenida por nuestro modelo.

Etapa/Partición	Partición 1	Partición 2	Partición 3	Partición 4	Partición 5
Etapa 1	Entrenamiento	Entrenamiento	Entrenamiento	Entrenamiento	Test
Etapa 2	Entrenamiento	Entrenamiento	Entrenamiento	Test	Entrenamiento
Etapa 3	Entrenamiento	Entrenamiento	Test	Entrenamiento	Entrenamiento
Etapa 4	Entrenamiento	Test	Entrenamiento	Entrenamiento	Entrenamiento
Etapa 5	Test	Entrenamiento	Entrenamiento	Entrenamiento	Entrenamiento

Figura 5.1: Tabla del uso de las particiones en cada etapa de la validación cruzada.

El intervalo de confianza que mostraremos será sobre la precisión de los modelos, y no sobre la **balanced multi-class accuracy**, ya que el intervalo de confianza es para esta métrica.

---

## 5.5 Segmentación

---

Para esta prueba, hemos escogido un modelo con la arquitectura U-Net, debido a la capacidad que ya ha demostrado en otros problemas a la hora de atacar problemas de segmentación. Los pasos que hemos seguido para entrenar el modelo son los siguientes:

1. Se hace una epoch de entrenamiento, es decir, se procesan todas las muestras del conjunto de entrenamiento por el modelo y se hace uso de la función de pérdida y del optimizador para conseguir los pesos adecuados.
2. Se procesa el conjunto de validación, sin propagar los gradientes ni entrenar el modelo.
3. Se analizan las métricas conseguidas sobre el conjunto de validación. Guardaremos el modelo si estas métricas són:
  - La menor pérdida generada por la función de pérdida.
  - El mayor índice de Jaccard
4. Se repite este ciclo hasta que se acaban las epochs, en nuestro caso, 100 epochs por modelo

Una vez hemos acabado el proceso de entrenamiento, hemos probado nuestro modelo con el conjunto de test para obtener los resultados reales sobre este problema.

---

---

# CAPÍTULO 6

## Resultados

---

En este capítulo vamos a comentar los resultados que hemos obtenido en nuestros experimentos.

### 6.1 Clasificación estándar

---

En el caso de la clasificación estándar, los resultados que hemos obtenido se pueden apreciar en la Tabla 6.1. Con estos resultados, podemos concluir que la red VGG16 no ha sido capaz de generalizar suficiente problema, ya que un valor de balanced multi-label accuracy de  $\frac{1}{8}$  significa que ha clasificado todas las muestras recibidas en la clase más probable. Esto explica por que, a pesar de solo clasificar en una clase, ha llegado a un valor de precisión de 50.8%. Este tipo de resultado resalta la importancia de usar una métrica diferente a la precisión en un problema desbalanceado, ya que se podrían aceptar modelos que en realidad están ignorando las clases menos comunes.

En cuanto al resto de modelos, parece que el modelo AlexNet ha sacado resultados menores a los demás, probablemente debido a la poca cantidad de parámetros que tenía esta red. La red residual que hemos creado ha entregado unos resultados aceptables para el problema, aunque las redes ResNet101 y Wide ResNet han sido capaces de lograr los mejores resultados. Esto se debe principalmente a la mayor cantidad de parámetros que pueden aprender respecto al resto de redes.

Modelo	Pérdida	Precisión	Balanced multi-label accuracy
VGG16	0.915	0.508	0.125
AlexNet	0.067	0.643	0.294
Red residual	0.1	0.703	0.547
ResNet101	0.05	0.709	0.630
Wide ResNet	0.063	0.726	0.651

**Tabla 6.1:** Resultados de los modelos en la prueba de clasificación estándar

Modelo	Pérdida	Precisión	Balanced multi-label accuracy
VGG16	0.101	0.125	0.125
AlexNet	0.08	0.233	0.233
Red residual	0.1	0.344	0.344
ResNet101	0.05	0.303	0.303
Wide ResNet	0.063	0.356	0.356

**Tabla 6.2:** Resultados de los modelos en la prueba de clasificación equilibrada

Modelo	Precisión	Intervalo de confianza al 95 %
VGG16	0.125	[0.1224, 0.1276]
AlexNet	0.631	[0.6252, 0.6368]
Red residual	0.703	[0.6962, 0.7098]
ResNet101	0.733	[0.7248, 0.7412]
Wide ResNet	0.742	[0.7358, 0.7482]

**Tabla 6.3:** Resultados de la precisión de nuestros modelos junto a sus intervalos de confianza al 95 %

## 6.2 Clasificación equilibrada

Los resultados para nuestra clasificación equilibrada de las muestras se pueden analizar en la Tabla 6.2. Los resultados de los modelos mantienen una estructura similar a la prueba anterior, aunque son peores generalmente. Como podemos observar, la VGG16 no es capaz de generalizar sobre las muestras, ya que siempre clasifica en la misma clase. Las mejores redes en este caso son nuestra red residual y la Wide ResNet, ya que obtienen los mejores resultados.

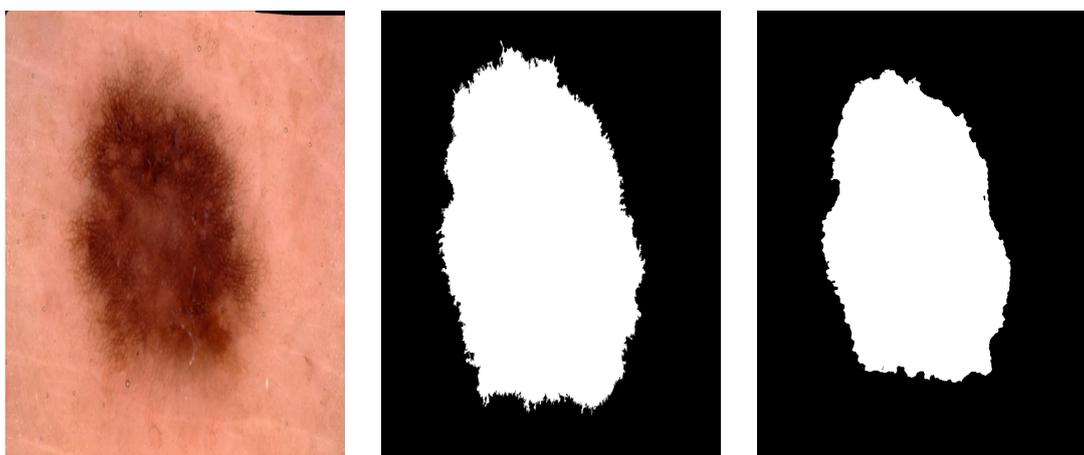
Es evidente en los resultados que agrupar las clases menos numerosas en una sola no ha sido de gran ayuda, ya que impacta en la métrica en gran medida. Además de esto, el sacrificio de usar menos muestras de entrenamiento para que las diferentes clases tuvieran el mismo número de muestras no ha resultado ser provechoso, ya que las redes no son capaces de aprender a generalizar suficiente para compensar la falta de muestras.

## 6.3 Validación cruzada

Para conseguir unos resultados más fiables de nuestros modelos, hemos decidido repetir el método que nos ha otorgado mejores valores de la métrica **balanced multi-label accuracy** con validación cruzada, para dar unos resultados más fiables de nuestros modelos. Los resultados de esta validación cruzada se pueden observar en la Tabla 6.4. Los resultados no han variado demasiado con los que se han sacado en la prueba de clasificación estándar, aunque el valor final de la métrica es más bajo del esperado. En la Tabla 6.3 podemos observar los intervalos de confianza para la precisión de nuestros modelos (Falta añadir la tabla).

Modelos	Partición 0	Partición 1	Partición 2	Partición 3	Partición 4	Media
VGG16	0.125	0.125	0.125	0.125	0.125	0.125
AlexNet	0.294	0.317	0.4	0.366	0.346	0.345
Residual net	0.547	0.528	0.554	0.556	0.524	0.542
ResNet101	0.630	0.601	0.635	0.623	0.580	0.614
Wide ResNet	0.651	0.632	0.609	0.566	0.603	0.612

**Tabla 6.4:** Resultados de la validación cruzada para la métrica **balanced multi-label accuracy**



**Figura 6.1:** Ejemplo de una muestra correctamente segmentada por nuestro sistema, con un índice de Jaccard superior a 0.8 sobre 1. La imagen de la izquierda representa la imagen original, la imagen central representa la máscara original y la imagen de la derecha representa la máscara predicha por nuestro modelo

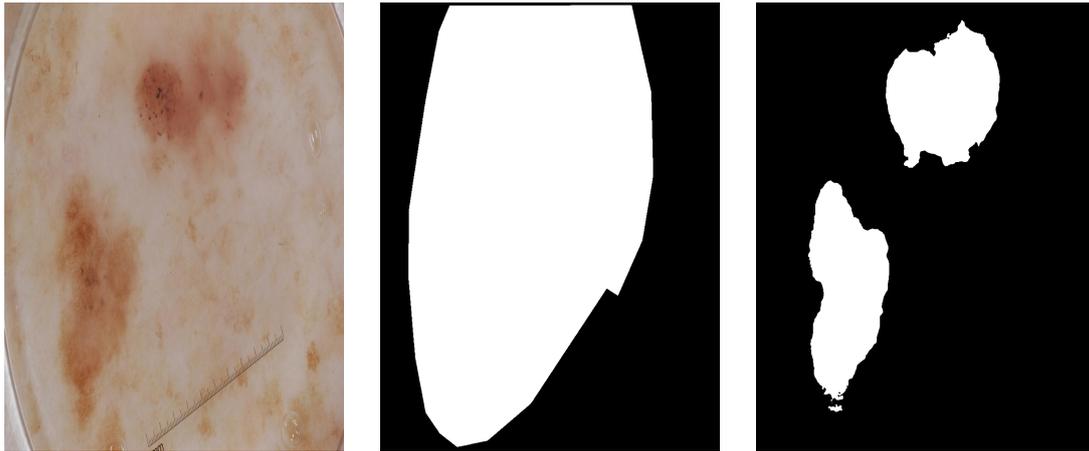
## 6.4 Segmentación

En el caso de la segmentación, hemos logrado un índice de Jaccard de 0.773 . Para analizar la eficacia real del sistema hemos preparado dos ejemplos visuales de las máscaras predichas por nuestro sistema, un ejemplo de máscara predicha correctamente (con un índice de Jaccard superior a 0.8 para esa imagen) y un ejemplo de máscara predicha incorrectamente (con un índice de Jaccard inferior a 0.4 para esa imagen). En el primer ejemplo, en el que la máscara se predice correctamente [6.1](#) podemos observar que la máscara tiene fundamentalmente la misma forma que la original, pero la máscara original abarca una área mayor al ser mas grande. En el segundo ejemplo, en el que la máscara se predice incorrectamente [6.2](#) podemos ver que la máscara predicha esta dividida en dos partes, mientras que la original abarca toda la zona circundante.

Como es nuestro único modelo en este problema, para saber como de válido es este resultado debemos compararlo con las propuestas del estado del arte.

## 6.5 Soluciones del estado del arte

En la [Tabla 6.5](#) hacemos una comparación con las mejores soluciones propuestas para este problema y los resultados que hemos obtenido para nuestros mode-



**Figura 6.2:** Ejemplo de una muestra incorrectamente segmentada por nuestro sistema, con un índice de Jaccard inferior a 0.4 sobre 1. La imagen de la izquierda representa la imagen original, la imagen central representa la máscara original y la imagen de la derecha representa la máscara predicha por nuestro modelo

<b>Modelos propios</b>	<b>Balanced multi-label accuracy</b>
VGG16	0.125
AlexNet	0.346
Residual Net	0.542
ResNet101	0.614
Wide ResNet	0.612
<b>Soluciones del estado del arte</b>	<b>Balanced multi-label accuracy</b>
Ensemble of Multi-Res EfficientNets + SEN154 2	0.636
Ensemble of EfficientNetB3-B4-Seresnext101	0.607
Ensemble, ood threshold 100 %	0.593
13 models + hierarchical approach to select outliers	0.578

**Tabla 6.5:** Resultados de nuestros modelos con los resultados de las propuestas para el set de datos ISIC 2019

los. Como podemos observar, aunque nuestro modelo no consigue la precisión que otorgan las soluciones propuestas a este problema, los otros modelos que hemos probado para resolver este problema superan la mayoría de los enfoques que se han utilizado. Cabe destacar que la mejor propuesta a este problema usa datos externos además de las imágenes, lo que explicaría su valor mayor en la métrica.

En la Tabla 6.6 podemos observar los resultados obtenidos por las mejores propuestas para este problema. Nuestra propuesta obtiene un valor inferior en la métrica usada pero dentro de un margen aceptable.

<b>Modelos propios</b>	<b>Índice de Jaccard</b>
UNet	0.773
<b>Soluciones del estado del arte</b>	<b>Índice de Jaccard</b>
MaskRcnn2+segmentation	0.802
Ensemble_with_CRF_v3	0.799
Automatic Skin Lesion Segmentation by DCNN	0.799
Skin Lesion Segmentation with Adversarial Learning	0.798

**Tabla 6.6:** Resultados para la segmentación, incluyendo nuestro modelo y las soluciones propuestas



---

---

# CAPÍTULO 7

## Conclusiones y trabajos futuros

---

En esta sección explicamos las conclusiones que hemos sacado a partir de nuestras pruebas y los trabajos en los que trabajaremos en un futuro.

### 7.1 Conclusiones

---

En este trabajo hemos sacado las siguientes conclusiones de nuestras pruebas:

- Nuestro acercamiento con modelos de machine learning obtienen resultados comparables a las propuestas del estado del arte para este problema.
- La clasificación equilibrada de las muestras no obtiene mejores resultados si la cantidad de muestras individuales de cada clase no es la suficiente.
- Aunque la segmentación de las imágenes es un problema abordable con la red UNet, todavía es mejorable para alcanzar las propuestas actuales.

### 7.2 Trabajos futuros

---

Como posibles trabajos futuros queremos destacar:

- La descripción de imagen médica, usando modelos recurrentes.
- Un sistema que nos permita el entrenamiento equilibrado de los datos sin desaprovechar las muestras extra de cada clase.
- Una exploración mas exhaustiva en el ámbito de la segmentación para mejorar los resultados obtenidos.



# Bibliografía

---

- [1] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen. Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 844–848, 2014.
- [2] Adrien Depeursinge, Alejandro Vargas, Alexandra Platon, Antoine Geissbuhler, Pierre-Alexandre Poletti, and Henning Müller. Building a reference multimedia database for interstitial lung diseases. *Computerized Medical Imaging and Graphics*, 36(3):227–238, 2012.
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.
- [5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [7] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.
- [8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [9] Dmitry Lachinov, Evgeny Vasiliev, and Vadim Turlapov. Glioma segmentation with cascaded unet. In *International MICCAI Brainlesion Workshop*, pages 189–198. Springer, 2018.
- [10] Umair Javaid, Damien Dasnoy, and John A Lee. Multi-organ segmentation of chest ct images in radiation oncology: Comparison of standard and dilated unet. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 188–199. Springer, 2018.
- [11] Noel C. F. Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen W. Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos

Liopyris, Michael A. Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *CoRR*, abs/1902.03368, 2019.