

DETECCIÓN DE LESIONES CUTÁNEAS

ANÁLISIS DE IMÁGENES CLÍNICAS

Lucía Fernández López
Antonio Huix Pérez
María Dolores López Valle

INTRODUCCIÓN

1.1 MARCO TEÓRICO

La piel es el órgano más grande, está compuesto por tres capas principales: epidermis, dermis y hipodermis. Cumple con funciones como la protección, termorregulación y detección de estímulos sensitivos.

La piel puede sufrir alteraciones y lesiones muy comunes y no necesariamente muy dañinas como quemaduras o acné. Sin embargo, también existen los cánceres de piel que pueden llegar a ser mortales

El desarrollo del proyecto ayuda a la prevención y detección de enfermedades o trastornos dermatológicos. Previamente el diagnóstico de lesiones cutáneas era principalmente visual. Con la ayuda de las nuevas tecnologías, programas e Inteligencia artificial se podrá adquirir, procesar y analizar imágenes detectando de qué tipo de alteración cutánea se trata.

1.2 MOTIVACIÓN

Las lesiones dermatológicas tienen una gran incidencia en la población. Sus consecuencias podrían mitigarse si se diagnostican a tiempo y de forma precisa. Por esto mismo, actualmente el objetivo de numerosos expertos es reducir la probabilidad de dar un mal diagnóstico, asociando un tratamiento que no sirva o que incluso sea contraproducente en base a los síntomas de una enfermedad. (1)

La función del dermatólogo es analizar la lesión en busca de señales estructurales y de colores que revelen un signo de alarma. Infinidad de instituciones médicas marcan las pautas a seguir como un estándar de procedimientos clínicos asociados al diagnóstico del cáncer de piel(2). Hoy en día las tecnologías pueden ayudar con estas detecciones eliminando así el error humano.

En la práctica clínica, la evaluación de las lesiones cutáneas ha dependido históricamente del examen visual y la experiencia del médico. A pesar de la eficacia de la dermatoscopia y otras técnicas avanzadas, la precisión diagnóstica puede verse limitada por la subjetividad y la variabilidad interobservador. En este contexto, surge la oportunidad de incorporar enfoques complementarios que utilicen tecnologías emergentes como la inteligencia artificial (IA) y el aprendizaje profundo para analizar y clasificar imágenes dermatológicas con alta precisión y objetividad.

El incremento en las bases de datos de imágenes de lesiones de piel ha sido la fuente de motivación de estos algoritmos que tienen por objeto diagnosticar estas lesiones. Los especialistas han desarrollado métodos para obtener características como el color,

los bordes, la simetría y muchos otros tipos de criterios de análisis. Esto ayudaría a realizar un diagnóstico precoz permitiendo dar tratamientos menos invasivos y mejorar los pronósticos, especialmente en condiciones potencialmente mortales. Además, la capacidad de discernir entre lesiones benignas y aquellas que requieren intervención médica es fundamental para evitar biopsias innecesarias, cirugías y la ansiedad asociada con los procedimientos de seguimiento.

1.3 CONTEXTO

SOLUCIONES EXISTENTES

Actualmente, la detección y clasificación de lesiones cutáneas se abordan a través de diversas técnicas. Los dermatólogos realizan exámenes visuales y dermatoscópicos (3), complementados en ocasiones con métodos histológicos invasivos como las biopsias. Además, existen sistemas basados en aprendizaje automático que utilizan características extraídas manualmente de las imágenes (4). Recientemente, la inteligencia artificial, especialmente las redes neuronales convolucionales (CNN), ha demostrado ser prometedora en la mejora de la precisión de la clasificación de imágenes dermatológicas [62]. De todas estas opciones nos vamos a centrar en las redes neuronales convolucionales (CNN)

DESARROLLO DE LA SOLUCIÓN

Para la detección y clasificación de lesiones cutáneas, nuestra propuesta de solución se centrará en dos módulos principales:

Módulo de Adquisición: Aprovecharemos los datasets públicos de imágenes dermatoscópicas etiquetadas. Estos datasets no solo ofrecen acceso a una gran cantidad de datos sin la necesidad de dispositivos de adquisición propios, sino que también proporcionan una diversidad de lesiones ya clasificadas por expertos, lo que facilita y mejora el entrenamiento de modelos de aprendizaje profundo.

Módulo de Clasificación: Implementaremos una red de clasificación basada en CNN para analizar las imágenes y determinar el tipo de lesión cutánea. Este enfoque automatizado tiene como objetivo superar las limitaciones de las soluciones actuales al proporcionar evaluaciones consistentes y escalables.

1.4 OBJETIVOS

OBJETIVOS PRINCIPALES

1. Construir un modelo clasificador capaz de clasificar diversos tipos de lesiones a partir de imágenes dermoscópicas. Deberá detectar y reconocer las siguientes patologías

DETECCIÓN DE LESIONES CUTÁNEAS

- MEL: Melanoma
 - NV Melanocytic Nevus
 - BCC: Basal Cell Carcinoma
 - AK: Actinic Keratosis
 - Benign Keratosis
 - DF: Dermatofibroma
 - VASC: Vascular Lesion
 - SCC: Squamous Cell Carcinoma
 - UNK: Unknown
2. Estudiar y analizar distintos modelos de redes neuronales
 3. Estudiar y analizar diferentes tipos de base de datos

OBJETIVOS PARCIALES

Este proyecto tiene como objetivo facilitar la práctica clínica diagnosticando de manera más temprana y más precisamente las alteraciones cutáneas, gracias a la adquisición y clasificación de imágenes médicas utilizando la IA. Esto nos permitirá una mejora en la toma de decisiones, en la salud general de la población y se evitará la aplicación de tratamientos más agresivos o como biopsias innecesarias.

1.5 CRONOGRAMA

DETECCIÓN DE LESIONES CUTÁNEAS

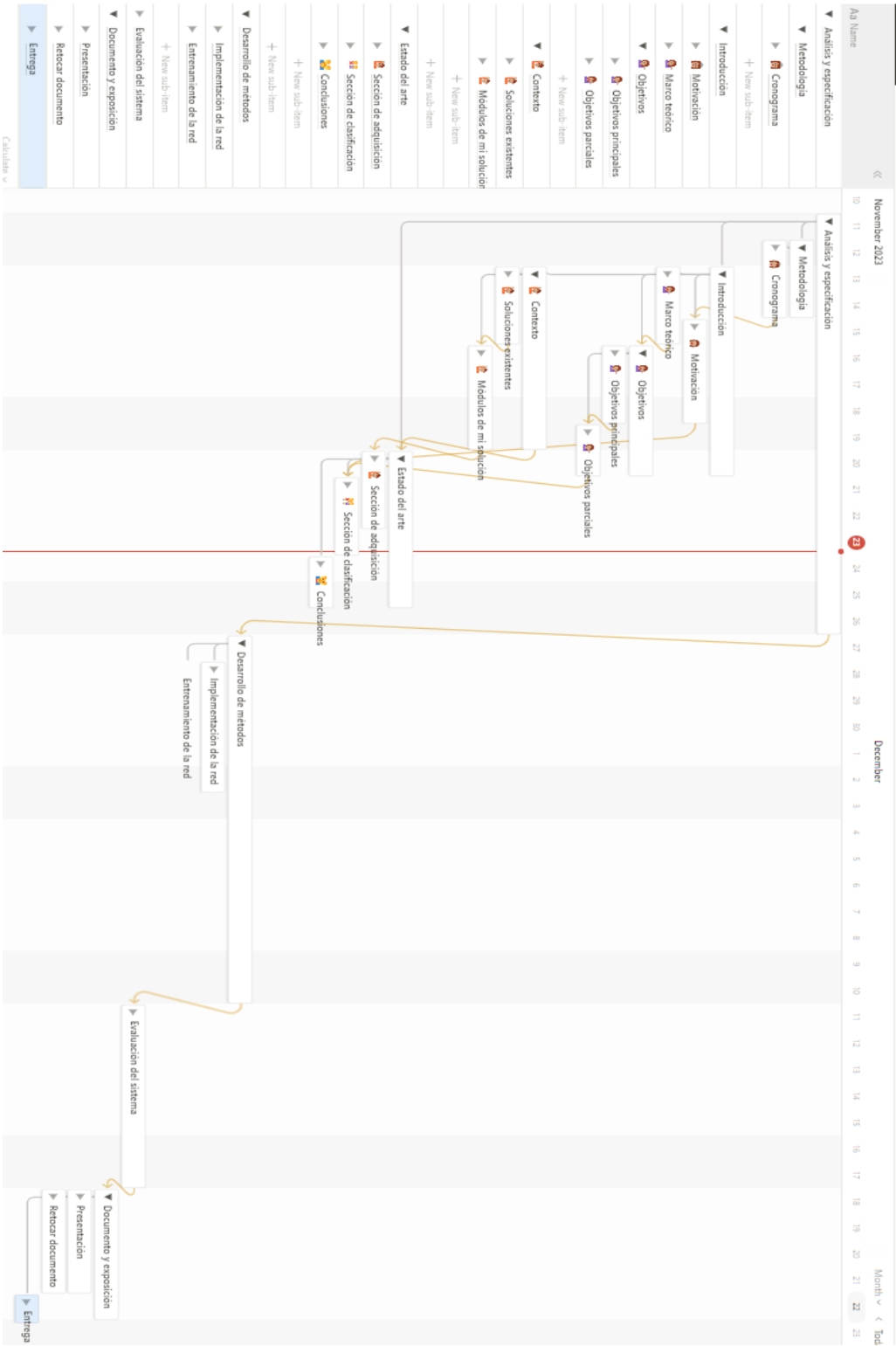


Figure 1.1: Cronograma de nuestro trabajo

ESTADO DEL ARTE

2.1 MÓDULO DE ADQUISICIÓN

La tabla 2.1 muestra una colección de datasets para el avance en el campo del análisis computarizado de lesiones cutáneas. Cada dataset ha sido seleccionado por su relevancia y contribución significativa a la investigación en dermatología. Proporcionan una amplia gama de imágenes dermatoscópicas y clínicas, cada una con metadatos y etiquetas detallados que son cruciales para el desarrollo y la validación de algoritmos de aprendizaje automático.

2.2 MÓDULO DE CLASIFICACIÓN

El modelo de red neuronal artificial (RNA) (16) es el que más se acerca a las conexiones que forman las neuronas en los seres humanos. Las RNAs tratan de predecir de la mejor forma posible la solución que andan buscando, a partir de repetir patrones de una manera organizada.

Estos métodos poseen además capacidades de autoaprendizaje, lo que supone que cuantos más datos se tengan mejores resultados se obtendrán. Un cerebro humano está formado por miles de millones de neuronas, y cada una está formada por células que procesan la información. Del mismo modo, una RNA tiene miles de neuronas artificiales que se suelen llamar unidades de procesamiento. Las personas necesitan reglas para elaborar la salida, al igual que estas redes. En el caso del modelo se denomina retropropagación. Esto consiste en una propagación en sentido inverso del error que se comete, de manera que se intenta mejorar el resultado final. Este algoritmo avanzado comienza con una fase de entrenamiento del set de datos para aprender a reconocer patrones y características que ayuden visualmente a clasificar cada clase. Esta fase está supervisada, ya que es realmente una comparativa. La red contrasta la imagen producida con la que debería producir. La diferencia entre ambas es el parámetro que se va ajustando en el proceso de retropropagación (1).

Se pueden clasificar las redes neuronales según el tipo de arquitectura o tecnología, que ayudarán a ser más efectivas para usos en concreto, de esta forma se podrá saber qué estructura es la más adecuada para el caso de las lesiones cutáneas. La clasificación es la siguiente:

- **Redes Neuronales Transformers(17):** ChatGPT o Bert, utilizan este tipo de redes neuronales, que son muy buenas para usos del procesamiento del lenguaje natural, entendiendo mejor el contexto, que otro tipo de redes. Este tipo de arquitectura, consume muchos recursos y energía.
- **Redes Neuronales Recurrentes(18):** se emplean bastante para la generación

DETECCIÓN DE LESIONES CUTÁNEAS

Dataset	Imágenes	Enfermedades Típicas	Notas
ISIC 2018 (5)	10,015	7 categorías	Datos accesibles con anotaciones expertas.
ISIC 2016 (5)	1,279	Melanoma y nevus	Utilizado en retos de clasificación.
Dermofit (6)	1,300	10 tipos de afecciones	Imágenes de alta calidad adquiridas en entorno clínico.
PH2 (7)	200	Melanomas, nevus y lesiones atípicas	Acompaña metadata y evaluaciones expertas.
MED-NODE (8)	170	Melanoma y lesiones nevas	Sin marca de agua y disponible para investigación.
Dandermin (9)	3,000	91 tipos de enfermedades	Mayormente adquiridas en contexto danés.
SD-198 (10)	6,584	198 categorías	Adquiridas por cámaras digitales y móviles.
DermIS (11)	7,172	735 tipos de enfermedades	Información detallada de diagnóstico y tratamiento.
AtlasDerm (12)	11,009	534 tipos de enfermedades	Imágenes clínicas con anotaciones de especialistas.
Asan dataset (13)	17,125	12 tipos de enfermedades	Originadas de un centro médico asiático.
Dermnet (14)	18,974	Amplio espectro	Disponible para la comunidad de investigación.
Derm101 (15)	22,979	525 tipos de enfermedades	Incluye metadata y anotaciones detalladas.
ISIC 2019 (5)	25,331	8 tipos de enfermedades	Imágenes de alta resolución de lesiones cutáneas con metadata detallada y etiquetas de diagnóstico de oro proporcionadas por expertos. Licencia CC-BY-NC.

Table 2.1: Datasets de enfermedades de la piel.

de texto y trabajar con secuencias, sin embargo, las transformers las están sustituyendo en muchos procesos similares.

- **Redes Neuronales Convolucionales(19):** Están diseñadas específicamente para procesar datos de cuadrícula, como imágenes y videos. Las capas convolucionales ayudan a detectar patrones y características espaciales en los datos de entrada. Generalmente se emplean para computer vision, reconocimiento facial y datos

bidimensionales. Para conducción autónoma o sistemas de vigilancia, pueden llegar a ser bastante útiles.

- **Red Neuronal Residual(20):** funcionan por capas de red y se basa en la introducción de conexiones residuales, es decir, en la conexión entre el output de las capas anteriores con el de las nuevas capas. Estas, aprenden las diferencias de lo que hay entre la entrada y salida del bloque. Cuando este concepto se implementa a las CNN, la velocidad disminuye y el error de entrenamiento aumenta, por lo que habrá que introducir conexiones

Las ResNet varían en función del número de capas presentes. De este tipo, nosotros analizaremos las dos siguientes

- * **ResNet34:** Es una red con 34 capas de profundidad, usa bloques residuales "plain", que son más simples.
- * **ResNet50V2:** Es una red con 50 capas, capaz de cargar más de un millón de imágenes y clasificar en 1000 categorías de objetos.(21)
- * **ResNet101V2** Es una red con 101 capas, por lo que generalmente tiene más parámetros.(22)

La Resnet50V2 y Resnet101V2 usan bloques bottleneck (más complejos) y tienen una mayor capacidad de aprendizaje. Sin embargo, la carga computacional aumenta, y por tanto también lo hace el tiempo de rendimiento, por lo que la Resnet34 nos permite tener una mayor eficiencia.

- **Xception:** Usa bloques de convoluciones que se pueden separar de forma repetida. Tiene menos parámetros, por lo que aportará más eficiencia computacional (23)(24)(25)
- **VGG16:** Tiene únicamente 16 capas y usa bloques de convoluciones con pequeños filtros 3x3. Es una red muy simple, con un reducido tiempo de entrenamiento. (26)
- **Redes Generativas Adversarias(27):** Consisten en dos redes neuronales, un generador y un discriminador, que compiten entre sí. El generador crea muestras de datos, y el discriminador evalúa si son reales o falsas. Este proceso mejora continuamente la calidad de las muestras generadas. Se emplea una red para generar el contenido y con otra red, se comprueba su calidad, en comparación con los datos de entrenamiento. Una vez que el resultado, llega a un mínimo de calidad, se decide el dato final de salida.
- **Redes Neuronales Siamesas(28):** Son utilizadas para comparar dos entradas

y determinar si son similares o diferentes. Se utilizan comúnmente en tareas de reconocimiento de objetos, verificación de identidad y clasificación de pares comparando similitudes, algunos ejemplos podrían ser verificar la autenticidad de un documento.

- **Redes Neuronales de Memoria Diferenciable:** están destinadas a procesos en los que se requieran memoria y capacidades de razonamiento.
- **Variantes de Redes Neuronales Recurrentes(29):** existen algunas variedades como las LSTM (Long Short-Term Memory) y las GRU (Gated Recurrent Unit) que buscan perfeccionar algunas desventajas de las RNN. Están diseñadas para trabajar con datos secuenciales o temporales, como series temporales, texto y audio. Tienen la capacidad de recordar información pasada gracias a las conexiones recurrentes en la red.

Haremos uso de Keras (30), una aplicación de modelos de deep learning. En la tabla 2.2 se muestran algunos de los modelos disponibles.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4
VGG16	528	71.3%	90.1%	138.4M	16	69.5
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7

Table 2.2: Modelos de deep learning en Keras.

2.3 CONCLUSIÓN

RAZONES PARA LA SELECCIÓN DEL DATASET "ISIC 2019"

Hemos elegido el dataset "ISIC 2019" por varias razones clave que lo hacen adecuado para nuestro proyecto de clasificación de lesiones cutáneas:

- **Accesibilidad:** El dataset es de libre acceso para la comunidad investigadora, lo que facilita la colaboración y la verificación de los resultados del estudio.
- **Alta Resolución y Calidad:** Las imágenes en formato JPEG son de alta resolución, lo que es crucial para la identificación de características diagnósticas sutiles en las lesiones cutáneas.
- **Etiquetado de Experto:** Viene con anotaciones detalladas y etiquetas de verdad

de oro para los diagnósticos de lesiones, lo cual es esencial para un entrenamiento supervisado eficaz y preciso de la red de clasificación.

- **Formato de Imagen Estándar:** El uso del formato JPEG es ideal, ya que es ampliamente compatible y fácil de manejar en la mayoría de los entornos de programación y plataformas de IA.
- **Diversidad de Datos:** El dataset incluye una variedad de lesiones cutáneas, lo que permite desarrollar un modelo de clasificación robusto capaz de generalizar a través de diferentes tipos de piel y condiciones.
- **Viabilidad Técnica:** Las imágenes no son excesivamente pesadas en términos de tamaño de archivo, lo que facilita el manejo de los datos y la eficiencia computacional durante el entrenamiento de modelos de aprendizaje profundo.

RAZONES PARA LA SELECCIÓN DE LA RED NEURONAL ResNet34

Este proyecto se centra en la clasificación de imágenes de lesiones cutáneas donde se utilizarán técnicas de deep learning centradas en el diagnóstico por lo que tras estudiar las definiciones de las redes expuestas anteriormente se comprueba que estos sistemas usan principalmente redes neuronales convolucionales (20), es la arquitectura que se ciñe correctamente al objetivo de la clasificación de imágenes donde se pretende detectar patrones y características espaciales en los datos de entrada. Otra de las razones por las que hemos elegido una ResNet, en comparación con Xception por ejemplo, es la familiaridad, ya que las ResNet son muy comunes y muy usadas en el campo del análisis y procesamiento de imágenes biomédicas.

Para la elección de la red neuronal, más concretamente del tipo de Resnet, se tiene en cuenta qué tipo de servidor requiere cada red, cuánto consume, la precisión de etiquetado con nuestros datos, y los recursos que usa.

Tras su análisis finalmente se ha decidido utilizar la Resnet34 frente a la Resnet50v2 o Resnet101V2 porque aunque la capacidad de aprendizaje y de capas sea menor, no necesitamos redes tan precisas y complejas. La principal ventaja es que al tener menos capas, la carga computacional es menor, del mismo modo el tiempo de entrenamiento será más corto y el rendimiento de los equipos no se verá afectado. Además, se ha comprobado que se ha aplicado esta red a nuestra base de datos en otros casos y los resultados fueron exitosos. (31)

DESARROLLO DE MÉTODOS

Los datos que utilizaremos son extraídos del Archivo ISIC (section 2.3) como se ha mencionado previamente, el cual contiene una gran colección de imágenes de lesiones en la piel. Este contiene cerca de 13,000 imágenes dermatoscópicas, las cuales fueron recolectadas por clínicas alrededor del mundo y adquiridas por distintos dispositivos dentro de cada clínica. Esto para asegurarse de tener una muestra lo suficientemente representativa, variada y clínicamente relevante.

Estas imágenes fueron capturadas de forma privada y asegurando su calidad como su diagnóstico. La gran mayoría de las imágenes fueron anotadas por expertos en cáncer de piel, el resto fue extraída de los metadatos de los expedientes electrónicos en la clínica.

Dichas lesiones se encuentran clasificadas en 9 clases diferentes (section 1.4).

Las imágenes tienen varios tamaños desde (767, 1022, 3 píxeles) hasta (120,90, 3 píxeles), y con diferentes ángulos y luces fotográficas. Se utilizan 8238 imágenes como conjunto de prueba. Para evaluar las imágenes se cuenta además con un conjunto de imágenes de validación.

Por otro lado, la red neuronal residual es una renombrada red neuronal artificial. Se ensambla en construcciones obtenidas de las células de la pirámide de la corteza cerebral. Las redes neuronales residuales logran esto usando atajos o “conexiones de salto” para moverse sobre varias capas. En nuestro caso, la Resnet34 (section 2.3 presenta 33 capas (32).

Los expertos implementan modelos tradicionales de redes neuronales residuales con saltos de dos o tres capas que contienen normalización por lotes y no linealidad entre ellas.

En este experimento, utilizamos ResNet porque permite entrenar hasta 1000 capas y aún así ofrece un buen rendimiento. La idea principal de utilizar ResNet es que ayuda a proporcionar una conexión de acceso directo idéntica para que se puedan omitir algunas capas para fines de entrenamiento más rápido. Representa la conexión de red neuronal de retroalimentación, que indica conexiones de atajo. Estas conexiones ayudan a realizar asignaciones de identidad y agregar los resultados de estas capas de acceso directo a las salidas de las capas apiladas. Con este procedimiento, las redes residuales garantizan una fácil optimización, un entrenamiento más rápido con menos épocas y una mayor precisión en comparación con otras redes simples. (31)

El Deep Learning(23), DL, actualmente es más como un “arte” que una ciencia debido a que no siempre sabes si tienes el modelo correcto, los suficientes datos, y

qué hacer al respecto si no está funcionando correctamente; en su mayoría el DL es experimental y requiere de mucha prueba y error para resolver sus problemas.

Por eso mismo, una parte importante en el Deep Learning es conocer los datos, pues no importa que tantos datos tengas sino la calidad de ellos. Mejores datos significa, que tenga una muestra representativa, variada, que no haya sesgo y que la captura sea lo mas cercano a la vida real.

3.1 EXPLICACIÓN DEL CÓDIGO

EXPLORACIÓN DE LOS DATOS

Un aspecto crítico para un buen análisis consiste en que nuestros datos sean de buena calidad, es decir, que sean representativos, diversos y que la captura se asemeje lo máximo posible a la realidad.

Para conseguir esto se empieza por contar cuántas fotos de cada clase hay teniendo en cuenta que cada imagen tienen un 1 en caso de que sí pertenezcan a esa clase y un 0 en caso contrario. Esto es lo que se conoce como etiquetado o 'ground truth' (33).

NV	12875
MEL	4522
BCC	3323
BKL	2624
AK	867
SCC	628
VASC	253
DF	239

Figure 3.1: Recuento de las imágenes de cada clase

Finalmente con este recuento se obtiene que hay varias clases que no están balanceadas, como podemos ver en la figura 3.1 hay muchas más imágenes de la clase NV en comparación con VASC o DF. Esto podría plantear un problema, ya que si el proceso de búsqueda se basa en la tasa de acierto estándar, favorecerá la cobertura de los ejemplos mayoritarios (NV en nuestro caso). Además, las reglas de clasificación que predicen la clase positiva suelen ser altamente especializadas, lo que resulta en un bajo grado de cobertura, y, por lo tanto, son descartadas en favor de reglas más generales. Por otro lado también hará que distinguir entre ejemplos ruidosos y ejemplos de la clase minoritaria no sea sencillo, lo que puede llevar a que el clasificador ignore por completo estos ejemplos (34).

Se usan a continuación las clases PILImage, para convertir las imágenes a Python Imaging Library y mostrar su forma (la altura, la anchura y el número de canales de

color) (Ver figura 3.2), y Transforms para convertir las imágenes a tensores y poder imprimir la forma, la intensidad mínima y máxima de los valores de píxeles en la imagen y el valor mínimo y máximo del tensor (Ver figura 3.3).

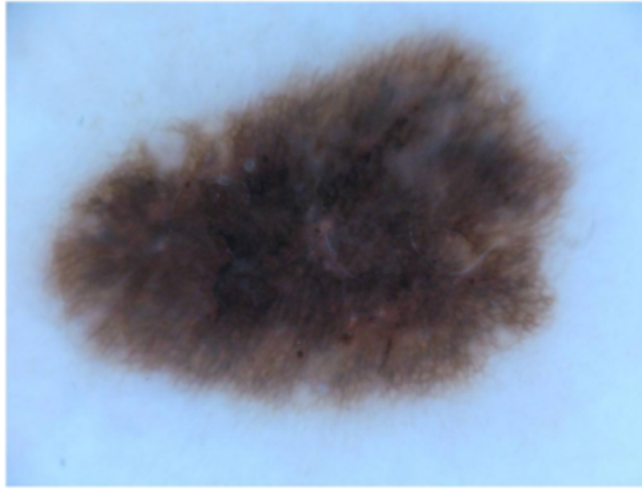


Figure 3.2: Forma de una imagen convertida a PILImage

```
(torch.Size([3, 767, 1022]),
 TensorImage(255, dtype=torch.uint8),
 TensorImage(16, dtype=torch.uint8))
```

Figure 3.3: Información de una imagen convertida a tensor

Para poder extraer las etiquetas de los datos, se ha creado una función que captura el nombre y la dirección del archivo de la imagen y retorna la etiqueta correspondiente. Esta función nos va a ser útil para acceder a las etiquetas asociadas a las imágenes.

Para procesar los datos se ha creado un proceso de procesamiento de datos gracias a Pipeline. Primero importamos la imagen en el formato PIL y después la convertimos a tensor. De esta forma, podremos imprimir de nuevo la imagen, obteniendo un resultado como el de la figura 3.2, y un tensor de PyTorch (35) que representa la imagen 3.4

Extraemos una lista con las direcciones de los archivos de las imágenes, esta es crucial ya que contiene la información necesaria a la hora de cargar y manipular las imágenes durante la preparación de datos y entrenamiento del modelo.

```

TensorImage([[251, 232, 243, ..., 237, 238, 255],
              [230, 175, 165, ..., 177, 184, 243],
              [245, 171, 147, ..., 156, 169, 243],
              ...,
              [238, 178, 169, ..., 186, 194, 242],
              [232, 183, 182, ..., 199, 201, 244],
              [252, 239, 249, ..., 238, 234, 255]],

             [[255, 241, 255, ..., 255, 250, 255],
              [234, 183, 181, ..., 203, 200, 246],
              [251, 179, 163, ..., 189, 190, 253],
              ...,
              [255, 196, 188, ..., 220, 217, 254],
              [244, 195, 194, ..., 225, 217, 249],
              [255, 246, 255, ..., 255, 245, 255]],

             [[255, 248, 255, ..., 255, 255, 255],
              [243, 194, 197, ..., 230, 216, 251],
              [255, 198, 186, ..., 222, 211, 255],
              ...,
              [255, 234, 231, ..., 245, 235, 255],
              [255, 211, 216, ..., 242, 230, 255],
              [244, 238, 255, ..., 255, 251, 255]]], dtype=torch.uint8)

```

Figure 3.4: Tensor de PyTorch que representa una imagen

Un paso imprescindible en el deep learning (23) es dividir el conjunto de datos en entrenamiento y validación, puesto que nos permite identificar si el modelo está sobreajustando los datos de entrenamiento, facilita el ajuste de parámetros, ayuda a prevenir fugas de información entre ambos conjuntos, proporciona una mejor comprensión del rendimiento, etc. Para conseguirlo, usamos la función `RandomSplitter` (36). Ver figura 3.5

```

((#20265) [2130,16042,22717,3207,11007,24567,21824,22594,18073,20246...],
 (#5066) [19971,21806,2927,15824,11246,5569,15166,802,6033,8178...])

```

Figure 3.5: Resultado de conjunto de datos divididos

La función `RandomSplitter` por defecto divide los datos en un 80% para entrenamiento y un 20% para validación. En este caso vemos que para el entrenamiento se van a utilizar 20.265 imágenes y para la validación 5.066. Entre corchetes aparece la posición de cada imagen. Ver figura 3.5

Fastai (37)

Permite reajustar el tamaño a todas las imágenes para asegurarnos que nuestro modelo rinda mejor, asimismo se puede personalizar cómo se quiere visualizar algunas imágenes. Además, nos permite mostrar metadatos específicos de una imagen, proporcionando información adicional que puede ser crucial para el análisis de datos

Transform

La clase Transform cuenta con dos métodos `encode` y `decode` los cuales se encargan de transformar los datos y revertirlos a su estado original. Esto nos da una gran ventaja ya que hace el procesamiento de datos más sencillo y manejable, gracias a ello podemos definir cómo preprocesar y etiquetar imágenes, así como revertir esas transformaciones.

DataSets

Se configura el conjunto de datos con listas; la primera se encarga de convertir y realizar las modificaciones a imágenes, mientras que la segunda asigna etiquetas numéricas a esas imágenes.

Tras una secuencia de transformaciones (creación de imágenes en formato PIL, redimensionamiento, conversión a tensores, asignación de etiquetas y asignación de números a las categorías) se crea un conjunto de datos que ya se puede utilizar para entrenar modelos de aprendizaje automático (38).

Para usar los datos en el entrenamiento es necesario cargarlos, se usará la clase `TfmdDL`, que además incluye otras transformaciones. Más tarde, los datos cargados se tienen que codificar y se procederá a acceder a las imágenes y etiquetas decodificadas, e imprimir las formas de las imágenes, con el fin de inspeccionar el estado de los datos después de aplicar las transformaciones y antes de pasarlo a un modelo de aprendizaje automático.

Aumentación de datos

Antes de pasar a la parte de entrenamiento, se realiza una aumentación de datos, es decir, se adquieren variaciones de los datos originales (ver figura 3.7), para que nuestro modelo funcione mejor, un ejemplo puede ser voltear la imagen, ponerla en blanco y negro, etc (ver figura 3.6 Normalmente las imágenes se muestran en cuadrículas.

DETECCIÓN DE LESIONES CUTÁNEAS

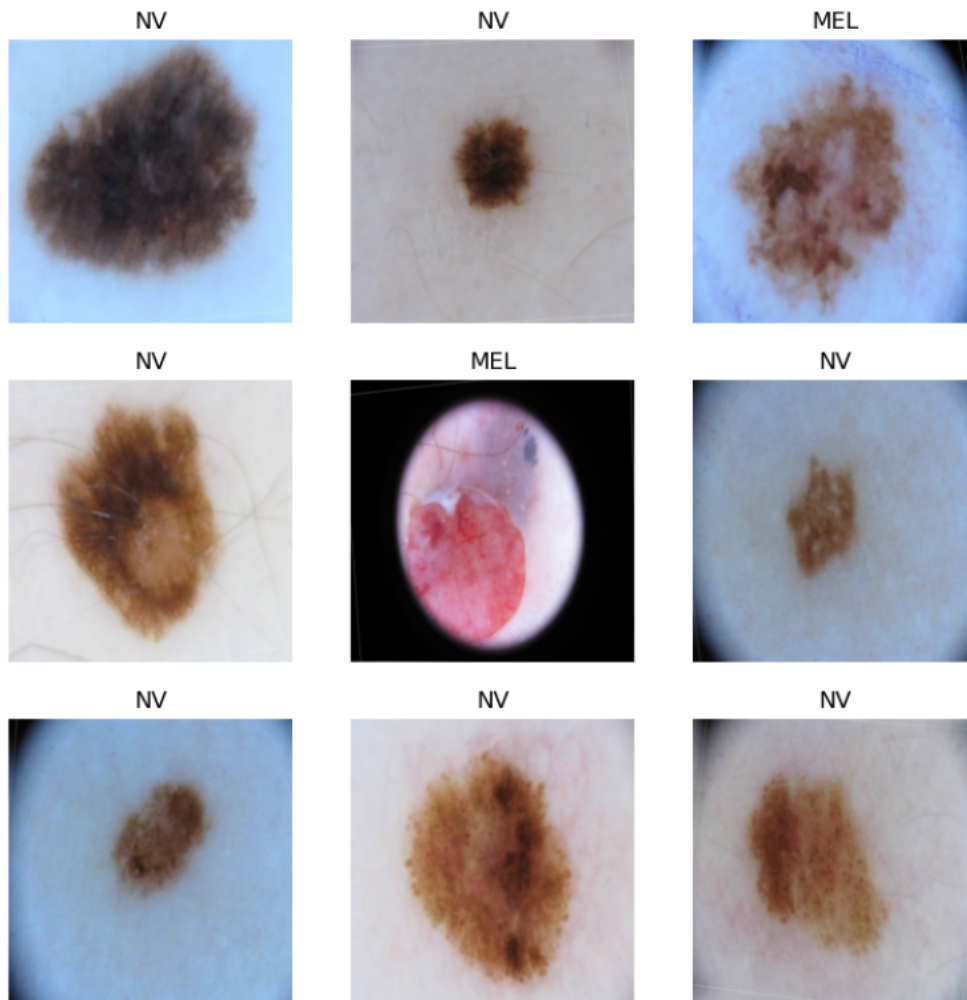


Figure 3.6: Imágenes con datos no aumentados

DETECCIÓN DE LESIONES CUTÁNEAS

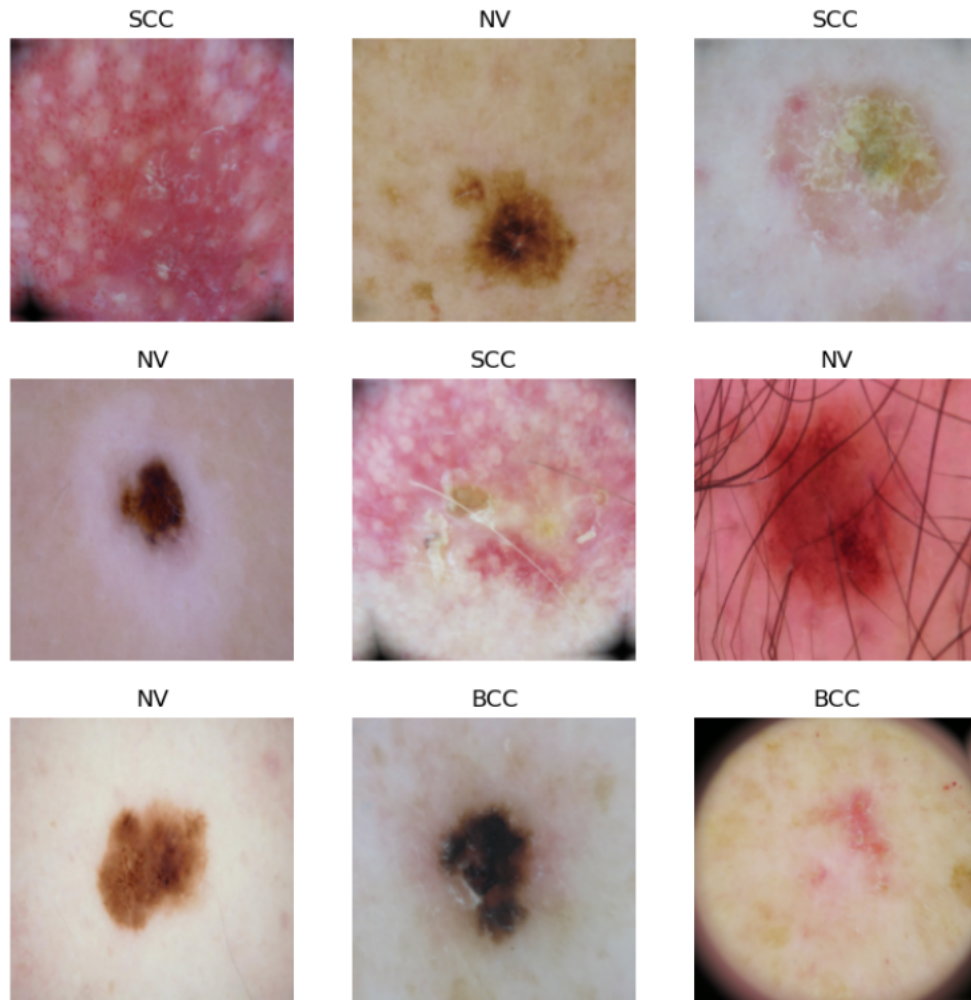


Figure 3.7: Imágenes con datos aumentados

ENTRENAMIENTO

El algoritmo aprende patrones presentes en los datos para poder clasificar las imágenes.

1. **Adquisición de datos y Procesado** Se obtienen los datos en la ruta especificada, se dividen los conjuntos de datos en entrenamiento y validación. Seguidamente, se aplican transformaciones específicas.
2. **Creación del modelo (Resnet34) de clasificación** "Cnnlearner" toma los datos usando el modelo resnet34 (gracias a una versión preentrenada de resnet34 de fastai(39)), también es necesario una función error y un algoritmo de opti-

mización. Más tarde, se entrena al algoritmo y se busca el learning rate (40) más óptimo, entrenando sobre un batch (41) del conjunto entrenamiento mientras incrementa el learning rate y captura como cambia la función de error. De esta forma elegimos al LR más óptimo como aquel que tiene la mayor pendiente (1e-2). Observar figura 3.8

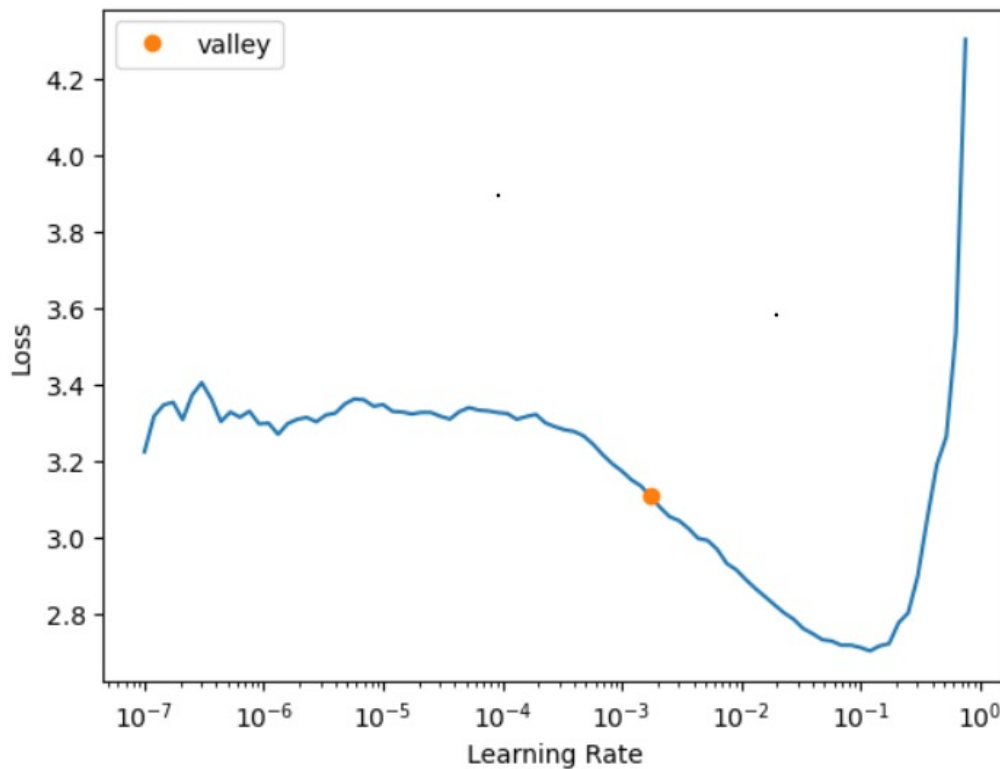


Figure 3.8: Gráfica de Learning Rate en la primera etapa del entrenamiento

A continuación, se procede a entrenar el algoritmo y encontrar los parámetros que tengan el mínimo error de la función. En este caso, se entrena en 10 épocas (Ver figura 3.9), posteriormente, con 4 (Ver figura 3.10) y finalmente se guardan los parámetros del modelo.

epoch	train_loss	valid_loss	accuracy	time
0	1.151631	0.891973	0.686143	5:31:24
1	1.011714	0.876300	0.685551	9:22:43
2	0.926046	0.997233	0.653968	6:57:05
3	0.877665	0.781302	0.713186	1:48:32
4	0.818360	0.906972	0.671141	2:05:19
5	0.768627	0.799926	0.713186	1:41:30
6	0.731356	0.704609	0.740426	11:47:35
7	0.689796	0.692229	0.748717	1:53:21
8	0.637964	0.667251	0.755626	1:33:02
9	0.643585	0.667025	0.756218	1:37:44

Figure 3.9: Resultado de conjunto de datos divididos

La imagen 3.9 muestra las métricas de rendimiento durante el entrenamiento para 10 épocas.

- **epoch:** Número de la época de entrenamiento.
- **train_loss:** Valor de la función de pérdida en el conjunto de entrenamiento.
- **valid_loss:** Valor de la función de pérdida en el conjunto de validación.
- **accuracy:** Precisión del modelo en el conjunto de validación.
- **time:** Tiempo que tardó cada época.

Los resultados indican una disminución continua tanto en *train_loss* como en *valid_loss*, y un aumento en la *accuracy*, lo cual sugiere un aprendizaje efectivo del modelo. El tiempo por época es bastante constante.

La imagen 3.10 presenta información similar, mostrando 4 épocas. Los patrones de mejoran en el rendimiento siendo similares a los de la Figura 3.9, con disminuciones en las pérdidas y mejoras en la precisión.

En resumen, la interpretación de los resultados es la siguiente:

- **Mejora con el Tiempo:** El modelo mejora con cada época.
- **Sobreajuste:** No hay señales de sobreajuste, ya que ambas pérdidas disminuyen.
- **Tiempo de Entrenamiento:** El tiempo por época se mantiene constante.

DETECCIÓN DE LESIONES CUTÁNEAS

Esto sugiere que el proceso de entrenamiento es estable y que el modelo está aprendiendo de manera efectiva.

epoch	train_loss	valid_loss	accuracy	time
0	0.777401	0.800626	0.699763	1:37:31
1	0.782614	0.729296	0.733912	1:31:53
2	0.693378	0.677713	0.753454	1:42:22
3	0.627891	0.655468	0.760955	1:40:56

Figure 3.10: Resultado de conjunto de datos divididos

Para afinar el modelo, se procede a desbloquear las capas que no se han entrenado inicialmente y así ajustar durante las etapas posteriores en un conjunto de datos específico.

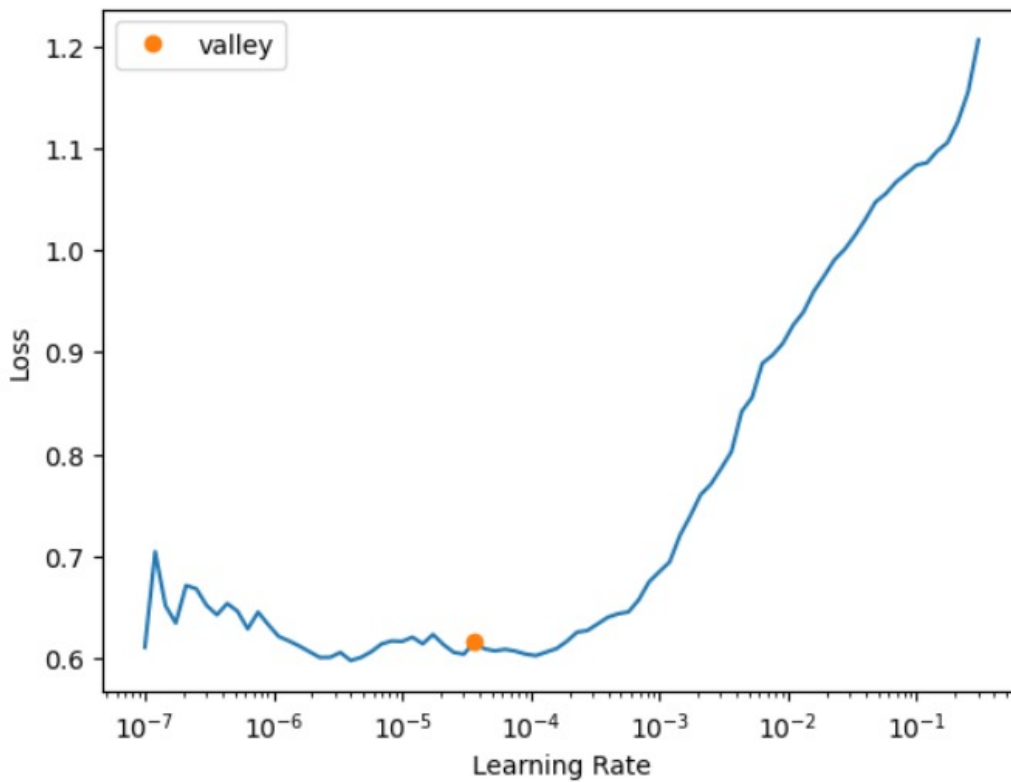


Figure 3.11: Gráfica de Learning Rate en la segunda etapa del entrenamiento

DETECCIÓN DE LESIONES CUTÁNEAS

Por lo que se vuelve a buscar el learning rate más óptimo (Ver figura 3.11), a guardar el modelo que se encuentra en la segunda etapa de entrenamiento y a cargarlo.

EVALUACIÓN

En el ámbito del aprendizaje profundo, la evaluación de modelos es crítica para determinar su efectividad y su capacidad para generalizar ante nuevos datos. Se ha utilizado un conjunto de métricas y técnicas para evaluar el modelo clasificador Resnet34.

Primero, se utiliza la clase `ClassificationInterpretation.from_learner` de Fastai para obtener una interpretación de las predicciones del modelo. Uno de los componentes clave de esta interpretación es la matriz de confusión (42) (Ver figura 4.1).

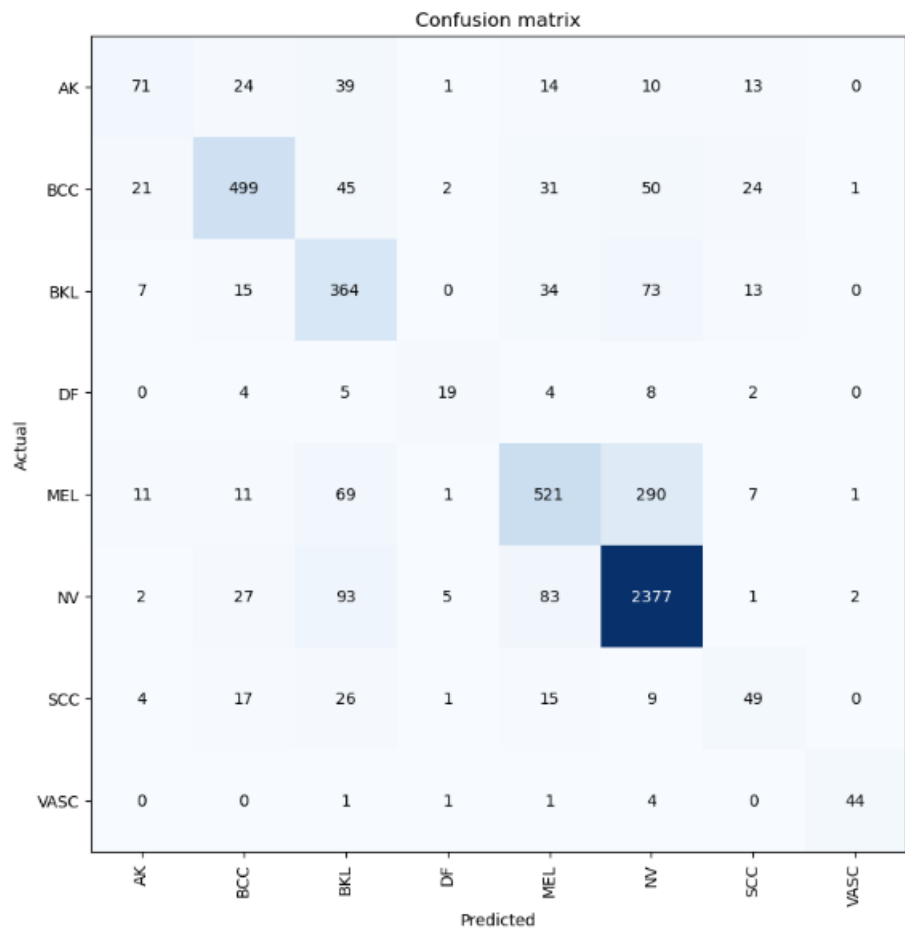


Figure 4.1: Matriz de Confusión

La matriz de confusión, representada en la Figura 4.1, es un resumen visual de cómo Resnet34 ha rendido a través de todas las clases. Los elementos en la diagonal principal

muestran el número de predicciones correctas para cada clase. Por ejemplo, para la clase BCC, el modelo ha predicho correctamente 499 casos. Sin embargo, también ha confundido 24 casos de BCC con AK y 39 con MEL, como se muestra en la segunda fila de la matriz.

Los valores fuera de la diagonal principal indican los errores de clasificación. La columna de MEL muestra que el modelo ha confundido 69 casos de AK como MEL, y la fila de MEL muestra que 290 casos de MEL han sido confundidos como NV. Estos errores son cruciales ya que indican áreas donde el modelo puede mejorar, particularmente en la distinción entre clases similares.

Las predicciones y los verdaderos valores de las etiquetas (`y_true`) se obtienen con el método `learner.get_preds()`. A partir de estos, se calcula el reporte de clasificación usando `classification_report` de Scikit-learn, el cual incluye métricas como la precisión, el recall y el F1-score (43) para cada clase.

La precisión mide la proporción de identificaciones positivas que fueron realmente correctas. Como vemos en la figura 4.2, para la clase AK, por ejemplo, la precisión es de 0.61, lo que significa que el 61% de las veces que el modelo predijo la clase AK, fue correcto.

	precision	recall	f1-score	support
AK	0.61	0.41	0.49	172
BCC	0.84	0.74	0.79	673
BKL	0.57	0.72	0.63	506
DF	0.63	0.45	0.53	42
MEL	0.74	0.57	0.65	911
NV	0.84	0.92	0.88	2590
SCC	0.45	0.40	0.43	121
VASC	0.92	0.86	0.89	51
accuracy			0.78	5066
macro avg	0.70	0.64	0.66	5066
weighted avg	0.78	0.78	0.77	5066

Figure 4.2:

El recall, también conocido como sensibilidad o tasa de verdaderos positivos, mide la proporción de positivos reales que fueron identificados correctamente. Para la clase AK, observando la figura 4.2 nuevamente, vemos que el recall es de 0.41, lo que indica que el modelo identificó correctamente el 41% de todos los casos reales de AK.

El F1-score es la media armónica de la precisión y el recall, y es una medida general del rendimiento de la precisión y el recall. Un F1-score (44) alto indica un buen equilibrio entre la precisión y el recall. Por lo que podemos concluir que las clases con un buen equilibrio entre la precisión y el recall son: BCC, NV y VASC.

Las métricas de la parte inferior de la figura 4.2 proporcionan un resumen del rendimiento global del modelo.

La accuracy por un lado indica el porcentaje de predicciones correctas en general. En este caso es del 78%, lo que significa que el modelo predijo correctamente el 78% de todas las instancias de prueba.

El macro AVG (45) es el promedio de las métricas calculadas independientemente para cada clase y posteriormente tomando el promedio de ellas. No tiene en cuenta el desequilibrio entre clases (es decir, el número de instancias de cada clase). La precisión macro, recall macro y F1-score macro son todos aproximadamente 0.7, lo que indica un rendimiento razonablemente uniforme a través de todas las clases, sin considerar el desequilibrio de clase.

El weighted Avg (46) calcula las métricas para cada clase, ponderadas por el soporte de la clase, lo que significa que las clases con más instancias contribuyen más al promedio. Esta métrica da una idea de la precisión y el recall, teniendo en cuenta el número de instancias para cada clase. En este caso, todas están alrededor de 0.77-0.78, lo que indica un buen rendimiento general teniendo en cuenta el soporte de cada clase.

Tras esta clasificación con el método `classification_report` podemos concluir que en general el modelo parece tener un rendimiento bastante sólido en algunas clases como BCC, MEL y VASC, con altas puntuaciones en todas las métricas. Sin embargo, otras clases como SCC y AK tienen un rendimiento más bajo, lo que podría indicar que al modelo le cuesta identificar estas clases correctamente esto se podría deber a que están entre las cuatro clases con menos datos para entrenar el modelo (Ver figura 3.1). La precisión y el recall ponderados y macro son buenos indicadores de que, en promedio, el modelo está actuando de manera aceptable en todas las clases de todas formas.

DETECCIÓN DE LESIONES CUTÁNEAS

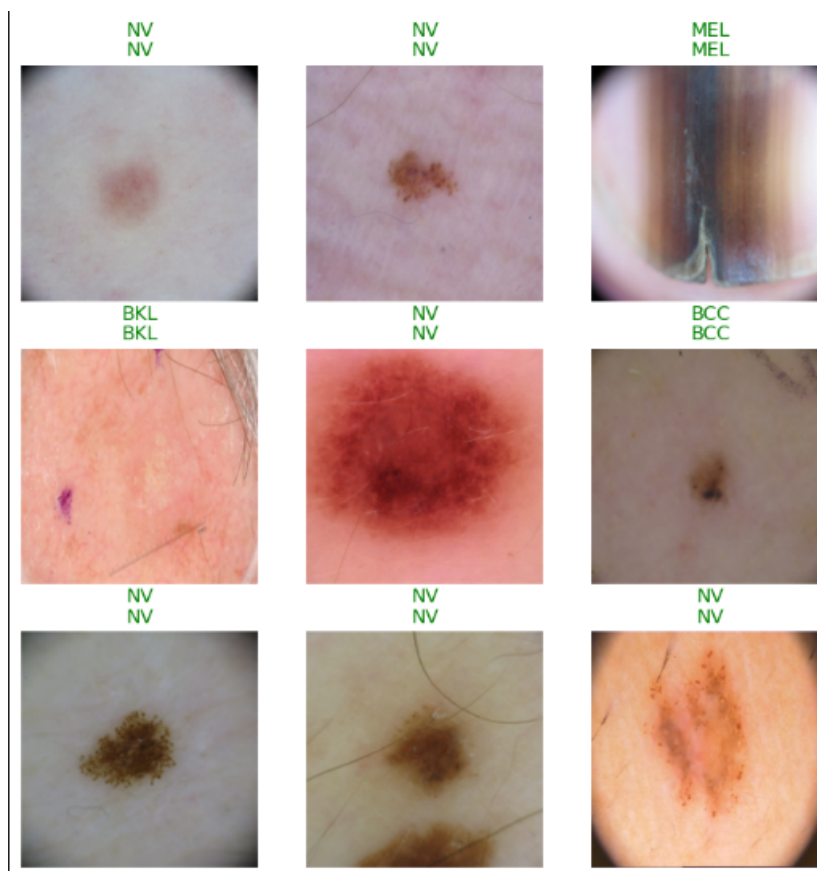


Figure 4.3: Comparación de las etiquetas predichas y etiquetas verdaderas

En la figura 4.3 se muestran las etiquetas predichas por el modelo en comparación con las etiquetas verdaderas. Es imprescindible que coincidan ambas etiquetas para verificar que el modelo es capaz de distinguir correctamente los diferentes tipos de lesiones, porque de lo contrario, se obtendría un diagnóstico incorrecto.

En este caso, todas las clasificaciones predichas coinciden con las etiquetas verdaderas como podemos observar en la figura 4.3, por lo que nuestro modelo ha hecho predicciones correctas.

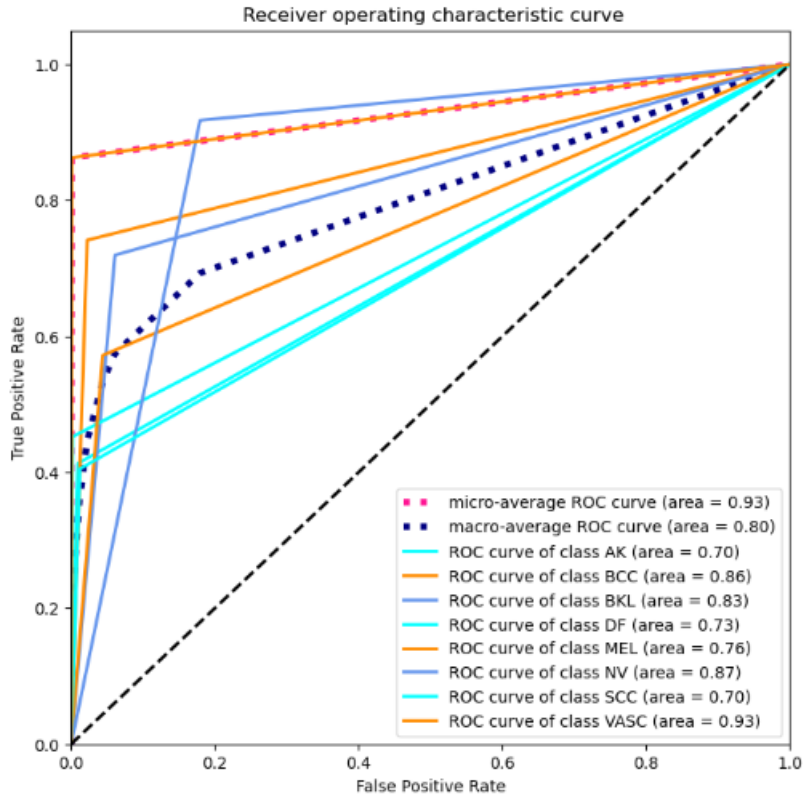


Figure 4.4: Curva Característica de operación del receptor

Ahora se procede a calcular la curva ROC (47) y el área bajo la curva (AUC) (48) para cada clase, así como una media micro y macro de todas las clases. La media micro calcula métricas globalmente, considerando cada elemento de la etiqueta como igual, mientras que la media macro calcula métricas para cada etiqueta y encuentra su media no ponderada. Esto es especialmente útil cuando las clases son desbalanceadas como es nuestro caso, como ya vimos anteriormente (ver sección 3.1).

La curva ROC (47), representada en la figura 4.4 evalúa la capacidad de un modelo de clasificación para distinguir entre clases, es una representación gráfica que muestra el rendimiento del modelo de clasificación en todos los umbrales de clasificación. Esta curva traza la tasa de verdaderos positivos (eje y) contra la tasa de falsos positivos (eje x). Por lo que la línea de un buen modelo clasificador estará lo máximo posible hacia la esquina superior izquierda (donde la tasa de falsos positivos es 0 y la de verdaderos positivos es 1).

Respecto a la evaluación del modelo utilizando el AUC, este proporciona una me-

didada agregada del rendimiento a lo largo de todos los posibles umbrales de clasificación, siendo 1 el mejor valor posible y 0.5 no mejor que una elección aleatoria. En la figura 4.4 cada línea representa cada una de las clases del modelo con su respectivo valor de área que varía entre 0 y 1.

La línea de promedio micro tiene un área de 0.93, es decir, que el modelo posee un buen rendimiento cuando considera todas las instancias. Del mismo modo, la curva del promedio macro tiene un área de 0.8, por lo que también tiene una buena capacidad de clasificación a través de las clases aunque no se pondere el número de instancias de cada una de ellas.

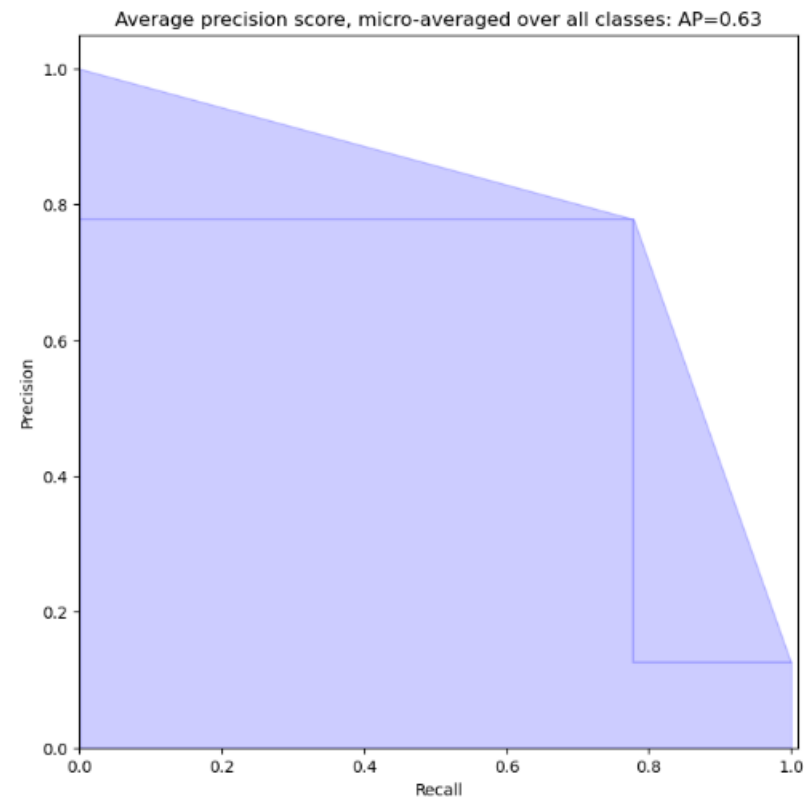


Figure 4.5: Curva de precisión-recall

Finalmente, se obtiene la curva Precision-Recall (PR) (49) (4.5), siendo esta una gráfica que muestra la compensación entre la precisión y el recall para diferentes umbrales. Un área alta bajo la curva PR representa tanto una alta recall como una alta precisión, donde una alta precisión se relaciona con una baja tasa de falsos positivos y una alta recall se relaciona con una baja tasa de falsos negativos. La función `pr_curve` calcula y

plotea estas curvas para todas las clases y para un promedio micro de cada una de ellas.

La curva de precisión-recall es una herramienta para evaluar la calidad de un modelo de clasificación. A diferencia de la curva ROC (50), la curva de precisión-recall pone el foco en la actuación del modelo con respecto a la clase positiva (la clase de interés) y es particularmente útil en situaciones donde las clases están muy desbalanceadas.

En la curva de precisión-recall, un modelo perfecto tendría una precisión de 1 en todo el rango de recall. En la figura 4.5 podemos ver cómo cambia la precisión a medida que aumentamos el umbral de decisión para predecir una instancia positiva.

El área sombreada bajo la curva se interpreta como la precisión promedio cuando el recall varía de 0 a 1. Un área grande bajo la curva es generalmente indicativa de que el modelo es capaz de lograr una alta precisión mientras mantiene un alto recall.

La gráfica de precisión-recall muestra un descenso en la precisión a medida que aumenta el recall, lo cual es normal ya que al intentar cubrir más ejemplos positivos (mayor recall), el modelo comienza a incluir más falsos positivos reduciendo la precisión.

El enfoque de micro-promedio toma en cuenta todas las clases conjuntamente para calcular una única precisión y recall, agregando los verdaderos positivos, falsos positivos y falsos negativos de todas las clases antes de calcular las métricas. Esto significa que el valor de la puntuación de precisión promedio (AP) (51) refleja el desempeño a través de todas las clases, lo cual es útil para evaluar el rendimiento general del modelo, especialmente con clases desbalanceadas.

La AP por lo tanto resume la curva de precisión-recall y es la media ponderada de las precisiones alcanzadas en cada umbral, con el aumento del recall desde el valor anterior como peso. Nuestro valor de AP es 0.63 indicando que, en promedio, el clasificador tiene una precisión del 63% a lo largo de todos los umbrales de decisión.

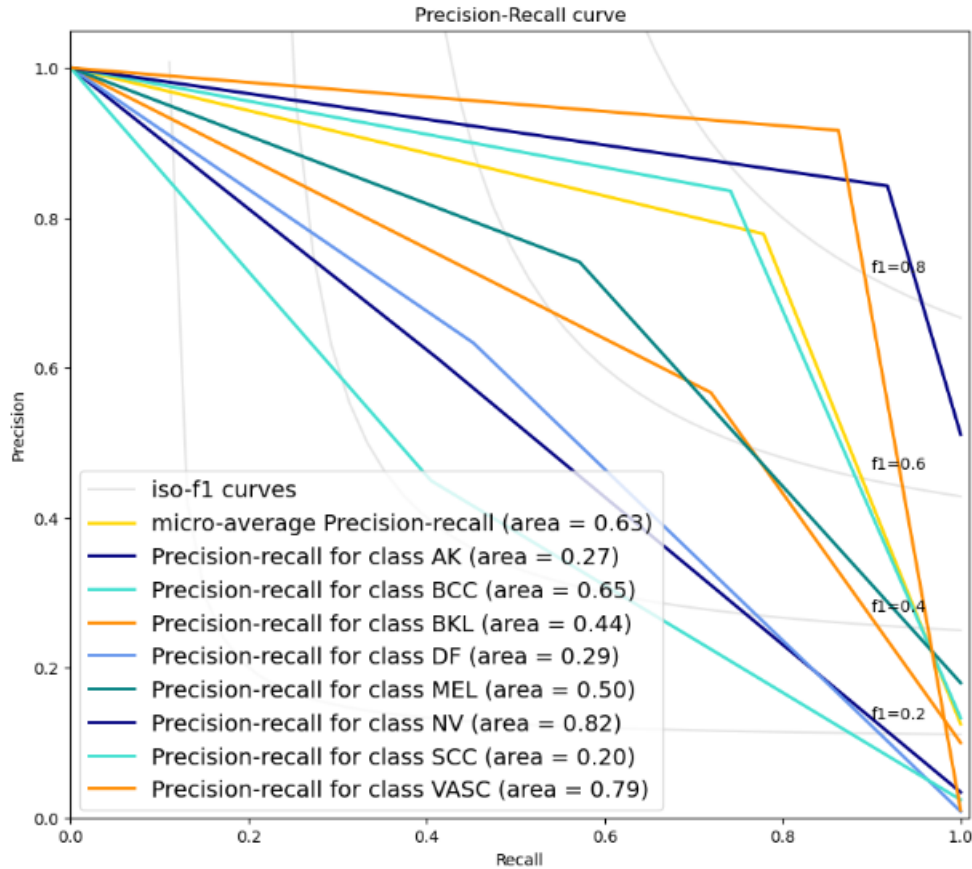


Figure 4.6: Curva de precisión-recall para todas las clases

El gráfico proporciona una visión del rendimiento del clasificador de forma general y para cada una de las clases. Los clasificadores con curvas más cercanas a la esquina superior derecha y mayor área son los más deseables, puesto que indican una mayor precisión. Cada línea de color, representa la curva de precisión-recall para una clase específica. Por ejemplo, podemos observar que el "NV" tiene un AUC de 0.82, que indica que puede identificar con una alta precisión y recall. A diferencia la clase "SCC", con un AUC más bajo de 0.2 cuyo rendimiento es muy pobre.

Las curvas ISO-F1 (líneas grises) representan el equilibrio entre la precisión y el recall (proporción de ejemplos positivos que fueron indentificados como tal) (52), y por tanto, el valor F1-score es constante. De tal manera que cuando la línea iso-F1 toque una curva de precisión-recall indica el valor F1 para ese nivel de recall específico. Por ejemplo, los puntos donde la línea de f1 es 0.8 toca una curva de precisión-recall indicando el umbral donde el modelo tiene un F1-score de 0.8.

Discusión de resultados

El rendimiento del modelo en las métricas anteriores proporciona una comprensión detallada de su capacidad de clasificación. Un modelo con altos valores de precisión y recall es deseable. Sin embargo, en conjuntos de datos desbalanceados, una alta precisión puede no ser representativa si la clase mayoritaria está dominando las predicciones. Del mismo modo, la curva ROC puede presentar una visión optimista del rendimiento del modelo si hay un desequilibrio significativo en la distribución de clases. Es por eso que la curva PR es particularmente útil, ya que se centra en el rendimiento del modelo en la clase minoritaria, que es a menudo de mayor interés en aplicaciones médicas como la detección de lesiones cutáneas.

Teniendo esto en cuenta se hace una recapitulación de las conclusiones a las que se han llegado tras evaluar el modelo:

Al modelo le cuesta identificar las clases con menos datos para entrenar el modelo, mientras que para aquellas clases con más datos el modelo presentó un rendimiento bastante sólido. Sin embargo, en promedio el modelo actúa de manera aceptable en todas las clases de todas formas.

Por otro lado vimos que el modelo posee un buen rendimiento cuando considera todas las instancias y posee también una buena capacidad de clasificación a través de las clases aunque no se pondere el número de instancias de cada una de ellas.

BIBLIOGRAPHY

- [1] C. G. García, “Aprendizaje profundo para la clasificación de lesiones pigmentadas de la piel,” 2020.
- [2] F. J. Montes Mantero, “Clasificación de lesiones dermatológicas a partir de imágenes dermoscópicas mediante el aprendizaje automático,” 2022.
- [3] T. G. Debelee, “Skin lesion classification and detection using machine learning techniques: A systematic review,” *Diagnostics*, vol. 13, no. 19, p. 3147, 2023.
- [4] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [5] International Skin Imaging Collaboration, “Isic challenge dataset.” <https://challenge.isic-archive.com/data>
- [6] “Dermofit image library.” <https://homepages.inf.ed.ac.uk/rbf/DERMOFIT/> A cognitive prosthesis to aid focal skin lesion diagnosis.
- [7] “Ph2 dataset.” <https://www.fc.up.pt/addi/ph2%20database.html> A dataset from the ADDI Project.
- [8] “Med-node database.” http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/ Dermatology database used in MED-NODE.
- [9] “Danderm.” <http://www.danderm.dk/> Dermatology Atlas.
- [10] “Sd-198.” <http://xiaopingwu.cn/assets/projects/sd-198/> Recognition of clinical skin disease images.
- [11] “Dermis.” <http://www.dermis.net/dermisroot/en/home/index.htm> Skin Disease Atlas.
- [12] “Atlasderm.” <http://www.atlasdermatologico.com.br/> Dermatology Atlas.
- [13] “Asan dataset.” https://figshare.com/articles/Asan_and_Hallym_Dataset_Thumbnails/5406136 Dataset available from Asan Medical Center.
- [14] “Dermnet.” <https://www.dermnet.com/> Extensive resource for dermatological diagnoses including a wide range of skin disease images.
- [15] “Dermioi (including dermquest).” <http://www.dermioi.com/> Clinical diagnosis label for skin lesion images.
- [16] R. Salas, “Redes neuronales artificiales,” *Universidad de Valparaíso. Departamento de Computación*, vol. 1, no. 1, pp. 1–7, 2004.

- [17] I. Matas González, “Clasificación de imágenes mediante redes neuronales convolucionales y técnicas de deep learning avanzadas: Transformers,” 2021.
- [18] C. Arana, “Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales,” tech. rep., Serie Documentos de Trabajo, 2021.
- [19] Á. Artola Moreno, “Clasificación de imágenes usando redes neuronales convolucionales en python,” 2019.
- [20] L. Tarrés Benet, “Clasificación de lesiones en la piel con un ensemble de redes neuronales residuales,” B.S. thesis, Universitat Politècnica de Catalunya, 2019.
- [21] “Resnet50v2.” <https://keras.io/api/applications/resnet/> Resnet50v2.
- [22] “Resnet101.” <https://es.mathworks.com/help/deeplearning/ref/resnet101.html> Resnet101.
- [23] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [24] “Matlab.” <https://es.mathworks.com/help/deeplearning/ref/xception.html> A neural red.
- [25] “resumen.” <https://iq.opengenus.org/xception-model/> mismo articulo resumido.
- [26] “Vgg16.” <https://datagen.tech/guides/computer-vision/vgg16/> VGG16.
- [27] N. A. Wolyniec Rojas *et al.*, “Detección de lesiones cutáneas en imágenes basado en redes generativas adversarias,” B.S. thesis, 2020.
- [28] A. Alcaide Martínez, “Redes neuronales convolucionales siamesas aplicadas a la verificación facial,” B.S. thesis, 2020.
- [29] E. Mota García, “Segmentación de imágenes utilizando redes neuronales recurrentes,” 2001.
- [30] “Keras aplications.” <https://keras.io/api/applications/fine-tune-inceptionv3-on-a-new-set-of-classes>.
- [31] N. Gouda and J. Amudha, “Skin cancer classification using resnet,” in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pp. 536–541, IEEE, 2020.
- [32] L. Moreno-Díaz-Alejo, “Análisis comparativo de arquitecturas de redes neuronales para la clasificación de imágenes,” Master’s thesis, 2020.

- [33] Í. Alonso Ruiz, A. C. Murillo Arnal, and A. B. Cambra Linés, “Segmentación semántica con modelos de deep learning y etiquetados no densos,”
- [34] A. F. Hilario, *Sistemas de clasificación basados en reglas difusas lingüísticas aplicadas a problemas con clases no balanceadas*. Editorial de la Universidad de Granada, 2010.
- [35] A. D. I. Pytorch, “Pytorch,” 2018.
- [36] D. Steinberg, M. Golovnya, and N. S. Cardell, “Data mining with random forests™,” 2004.
- [37] J. Howard and S. Gugger, “Fastai: A layered api for deep learning,” *Information*, vol. 11, no. 2, p. 108, 2020.
- [38] A. Moreno, E. Armengol, J. Béjar Alonso, L. A. Belanche Muñoz, C. U. Cortés García, R. Gavalda Mestre, J. M. Gimeno, M. Martín Muñoz, and M. Sánchez-Marré, “Aprendizaje automático,” 1994.
- [39] “Fastai.” <https://sitiobigdata.com/2019/05/01/fastai-simplificando-el-aprendizaje-profundo/>. Fastai.
- [40] “Learningrate.” <https://platzi.com/tutoriales/1687-machine-learning-2019/9832-que-es-el-learning-rate/>. LearningRate.
- [41] “Batch.” <https://www.profesionalreview.com/2018/11/25/que-es-el-procesamiento-batch/>. Batch.
- [42] N. D. Marom, L. Rokach, and A. Shmilovici, “Using the confusion matrix for improving ensemble classifiers,” in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*, pp. 000555–000559, IEEE, 2010.
- [43] C. Goutte and E. Gaussier, “A probabilistic interpretation of precision, recall and f-score, with implication for evaluation,” in *European conference on information retrieval*, pp. 345–359, Springer, 2005.
- [44] D. Chicco and G. Jurman, “The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation,” *BMC genomics*, vol. 21, no. 1, pp. 1–13, 2020.
- [45] T. Gowda, W. You, C. Lignos, and J. May, “Macro-average: rare types are important too,” *arXiv preprint arXiv:2104.05700*, 2021.
- [46] J. Mosig, “The weighted averages algorithm revisited,” *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 4, pp. 2011–2018, 2012.

- [47] L. de Ullibarri Galparsoro and P. Fernández, “Curvas roc,” *Atención Primaria en la Red*, vol. 5, no. 4, pp. 229–35, 1998.
- [48] T. Yang and Y. Ying, “Auc maximization in the era of big data and ai: A survey,” *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–37, 2022.
- [49] J. Miao and W. Zhu, “Precision–recall curve (prc) classification trees,” *Evolutionary intelligence*, vol. 15, no. 3, pp. 1545–1569, 2022.
- [50] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240, 2006.
- [51] M. de Usuario and P. A. Sig, “3.3. métricas y puntuación: cuantificar la calidad de las predicciones,”
- [52] “Recall.” <https://datasmarts.net/es/mas-alla-del-accuracy-precision-recall-y-fi/. recall>.