

Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías de las
Telecomunicaciones

Algoritmo de inteligencia artificial basado en redes neuronales convolucionales para la clasificación de lesiones de la piel

Autor: Alejandro Gamero Salas

Tutoras: Dra. Carmen Serrano Gotarredona

Dra. Begoña Acha Piñero



Proyecto Fin de Grado
Ingeniería de las Tecnologías de las Telecomunicaciones

Clasificador de Imágenes Médicas basado en Inteligencia Artificial

Análisis de las distintas estrategias de ensamblado para clasificación de imágenes
medicas

Autor:

Alejandro J. Gamero Salas

Tutoras:

Dra. Carmen Serrano Gotarredona

Catedrática de Universidad

Dra. Begoña Acha Piñero

Catedrática de Universidad

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2024

Proyecto Fin de Carrera: Análisis de las distintas estrategias de ensamblado para clasificación de imágenes medicas

Autor: Alejandro J. Gamero Salas

Tutoras: Dra.Carmen Serrano Gotarredona
Dra. Begoña Acha Piñero

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2024

El Secretario del Tribunal

Agradecimientos

A mis padres por su paciencia, a mis hermanos y amigos por su apoyo y a mis profesores por sus enseñanzas.

Alejandro J. Gamero Salas

Alumno del Grado de ingeniería de Telecomunicaciones

Sevilla, 2024

Resumen

Durante este proyecto hemos diseñado tres redes de inteligencia artificial que generan un vector de probabilidades con tantos elementos como clases son capaces de distinguir. El objetivo es crear varios sistemas de ensamblaje de redes neuronales que mejoran las capacidades de las redes por separado.

Para ello, hemos obtenido las imágenes para entrenar y analizar las redes de la base de datos del ISIC[1][3][7][10]. Las distintas técnicas de ensamblaje que se han usado han sido inspiradas por métodos que ayudan en clasificadores basados en machine learning, como puede ser la votación por mayoría o por distancia euclídea con respecto a unos valores medios. Esto ha provocado que no todas las técnicas consigan una mejora con respecto a las redes neuronales independientes, pues algunas técnicas deshacen el aprendizaje e impiden una clasificación correcta.

Durante el proyecto se monitorizan varias métricas que permitan comparar los sistemas entre sí. Siendo la principal métrica, la precisión multiclase normalizada, esto es, la media de las precisiones del sistema por clase. Sin embargo, obtener buenos resultados en esta métrica exclusivamente no implica que sea mejor clasificador, como veremos para el caso de ensamblaje por red neuronal. Esta técnica obtiene los mejores resultados para esta métrica, pero peores en el resto de métricas elegidas.

El proyecto analizará cinco modelos de ensamblaje de redes neuronales distintos, cada uno obteniendo unas métricas propias. Siendo uno de estos modelos, el ensamblaje de tres redes neuronales dentro de una red neuronal que engloba a las otras tres. Esta técnica cumple con el mejor de los resultados de la métrica principal, pero falla más que el resto de técnicas a la hora de distinguir las clases minoritarias, como queda reflejado en las métricas secundarias.

During this project we have designed three artificial intelligence networks that generate a vector of probabilities with as many elements as there are classes they are capable of distinguishing. The goal is to create several neural network assembly systems that improve the capabilities of the networks separately.

To do this, we have obtained the images to train and analyze the networks from the ISIC database[1][3][7][10]. The different assembly techniques that have been used have been inspired by methods that help in classifiers based on machine learning, such as majority voting or Euclidean distance with respect to average values. This has meant that not all techniques achieve an improvement over independent neural networks, as some techniques undo learning and prevent correct classification.

During the project, several metrics are monitored that allow the systems to be compared with each other. The main metric being the normalized multiclass precision, that is, the average of the system's precision per class. However, obtaining good results in this metric exclusively does not imply that it is a better classifier, as we will see in the case of neural network assembly. This technique obtains the best results for this metric, but worse results for the rest of the chosen metrics.

The project will analyze five different neural network assembly models, each obtaining its own metrics. One of these models being the assembly of three neural networks within a neural network that encompasses the other three. This technique meets the best of the results of the main metric, but fails more than the rest of the techniques when it comes to distinguishing the minority classes, as reflected in the secondary metrics.

-translation by google-

Índice

Agradecimientos	7
Resumen	9
Abstract	10
Índice	11
Índice de Tablas	12
Índice de Figuras	13
Notación	14
1 Introducción	16
1.1. <i>Estado del Arte</i>	
1.2. <i>Inteligencia Artificial</i>	
1.3. <i>Redes Neuronales Convolucionales</i>	19
1.3.1 Modelos ensamblados	
1.4. <i>Lesiones dermatológicas</i>	22
1.4.1.1 Trastornos benignos	
1.4.1.2 Trastornos malignos	
1.4.1.3 Trastornos intermedios	
2 Metodología del proyecto	25
2.1. <i>Base de Datos</i>	25
2.2. <i>Procesamiento de datos</i>	26
2.3. <i>Redes Neuronales</i>	26
2.3.1. VGGnet de 18 Capas	
2.3.2. Inception V3	
2.3.3. RESNET50 V2	
2.4. <i>Técnicas de Ensamblaje</i>	31
2.3.3. Ensamblaje por mayoría simple	
2.3.3. Ensamblaje por mayoría cuadrática	
2.3.3. Ensamblaje por minimizador de distancia	
2.3.3. Ensamblaje Neuronal	
2.5. <i>Métrica utilizada</i>	32
2.3.3. Precisión Multiclase Normalizada (BMA)	
2.3.3. Sensibilidad y Especificidad	
2.3.3. F1 score	
2.3.3. Valores Predictivos (Positivo y Negativo)	
2.6. <i>Matriz de Confusión</i>	34
2.7. <i>Obtención de resultados</i>	36
3 Resultados	38
3.1. <i>Matriz de Confusión y análisis estadístico</i>	38
3.2. <i>Primer modelo a ensamblar: Distancia Euclídea respecto al vector prototipo</i>	39

3.3. Segundo modelo: Adición simple de las 3 redes	41
3.4. Tercer modelo de ensamblaje: Adición cuadrática de las 3 redes	42
3.5. Modelo de ensamblaje directo	44
3.6. Modelos independientes	45
3.6.1. Modelo VGG	
3.6.2. Modelo RESNETv2	
3.6.3. Modelo Inceptionv3	
3.7. Cálculo de la métrica	48
4 Conclusiones	49
Referencias	50

ÍNDICE DE TABLAS

Tabla 2-1: Matriz de confusión del modelo cuadrático	34
Tabla 3-1. Matriz de Confusión ideal	37
Tabla 3-2. Matriz de Confusión de ensamblaje con Atlas	39
Tabla 3-3. Matriz de confusión del modelo simple	40
Tabla 3-4: Métricas por clase del modelo simple	41
Tabla 3-5. Matriz de confusión del modelo cuadrático	42
Tabla 3-6: Tabla de métricas del modelo cuadrático	42
Tabla 3-7. Matriz de confusión del modelo de ensamblaje directo	43
Tabla 3-8: Tabla de métricas por clase del modelo de ensamblaje directo	44
Tabla 3-9: Tabla de métricas del modelo VGG	45
Tabla 3-10: Tabla de métricas del modelo Resnet50v2	45
Tabla 3-11: Tabla de métricas del modelo InceptionV3	46
Tabla 3-12: Principal métrica del concurso	47
Tabla 3-13: Resultado de las métricas secundarias pedidas	47
Tabla 3-14: Resultados de la métrica secundaria: Sensibilidad	48

ÍNDICE DE IMÁGENES

Imagen 1-1: Estructura de entrenamiento de redes	16
Imagen 1-2: Estructura de una neurona en una IA	17
Imagen 1-3: Ejemplo de límites de decisión ideal de un modelo binario	18
Imagen 1-4: Estructura genérica de una Red Neuronal Convolutacional modelo VGG	19
Imagen 1-5: Estructura genérica VGG16	19
Imagen 1-6: Estructura de InceptionV3	20
Imagen 1-7: Estructura de Resnet	20
Imagen 1-8: Ejemplo de modelo ensamblaje basado en el espacio de señal de [imagen. 1-3]	21
Imagen 1-9: Representación del problema de desbalance de clases.	22
Imagen 2-1: Estructura general de una red neuronal	26
Imagen 2-2: Estructura de la red Inception	26
Imagen 2-3: Estructura de la red VGG	27
Imagen 2-4: Evolución de métricas con los ciclos	29
Imagen 2-5: Estructura de capas de la red RESNET	30
Imagen 2-6: Distribución de los elementos de la matriz de confusión	33
Imagen 3-1: Implementación de métricas	38

$\sum_x(\cdot)$	Sumatorio en el rango de x
$f(\cdot) _x$	Función supuesto/conocido el valor de x
CNN	Convolutional Neural Network, Red Neuronal Convolutacional
ANN	Artificial Neural Network, Red Neuronal Artificial
ISIC	International Skin Imaging Collaboration
IA	Inteligencia Artificial
DL	Aprendizaje profundo, Deep Learning
TP	Verdaderos Positivos, True Positives
TN	Verdaderos Negativos, True Negatives
FP	Falsos Positivos, False Positives
FN	Falsos Negativos, False Negatives
P	Positivos Totales
N	Negativos totales
Train_X	Conjunto de Imágenes de entrenamiento
Train_Y	Conjunto de Diagnósticos previos de las imágenes de Train_X
Val_X	Conjunto de Imágenes de Validación
Val_Y	Conjunto de Diagnósticos previos de las imágenes de Val_X
Test_X	Conjunto de Imágenes de Prueba
Test_Y	Conjunto de Diagnósticos previos de las imágenes de Test_X
UINT8	Unsigned INTeget 8 Bits, variable numérica sin signo 8 bits (valores $\subseteq [0:255]$)
Matriz [AxBxC]	Matriz con A filas, B columnas y C canales
ReLU	REctified Linear Unit
$X \subseteq [a:b]$	X puede tomar valores desde a hasta b
$\min\{\cdot\}$	Valor mínimo de $\{\cdot\}$
$(\cdot)^2$	Función al cuadrado
MC	Matriz de Confusión
ANG	Angioma

NEV	Nevus, pecas
AK	Queratosis Actínica (Actinic Keratosis)
BK	Queratosis Benigna (Benign Keratosis)
MEL	Melanoma
CBC	Carcinoma BasoCelular
CCE	Carcinoma de Células Escamosas
VASC	Lesión Vascular
F_{XY}	Función cuyos índices dependen de X e Y
BMA	Métrica, Balanced Multi-class Accuracy, Precisión Multiclase Normalizada
Rayos UV	Rayos UltraVioleta

1 INTRODUCCIÓN

En este proyecto vamos a desarrollar tres inteligencias artificiales que clasifiquen entre ocho tipos de lesiones distintas en la piel. El objetivo final de desarrollar dichas IAs será ensamblar sus salidas para generar un clasificador de imágenes médicas.

Para determinar el ensamblaje de dichos clasificadores hemos desarrollado varios modelos y los hemos comparado bajo las mismas métricas:

- Votación por mayoría simple
- Votación por mayoría cuadrática
- Decisor de distancia mínima con respecto a valores medios
- Inteligencia Artificial como conjunto

Como veremos posteriormente, la última técnica contempla las demás, por lo que obtendrá mejores resultados.

Las IAs que desarrollaremos serán modelos predictivos basados en redes neuronales convolucionales (CNNs). Estos modelos se basan en la agrupación secuencial de capas de neuronas, donde cada capa ejecuta una función con unos pesos internos. Estos pesos se adaptan a las imágenes que se les proporciona durante la etapa de entrenamiento. De este modo, el sistema aprenderá a clasificar imágenes entre las ocho categorías establecidas. Las imágenes utilizadas para este proyecto han sido categorizadas previamente por expertos.

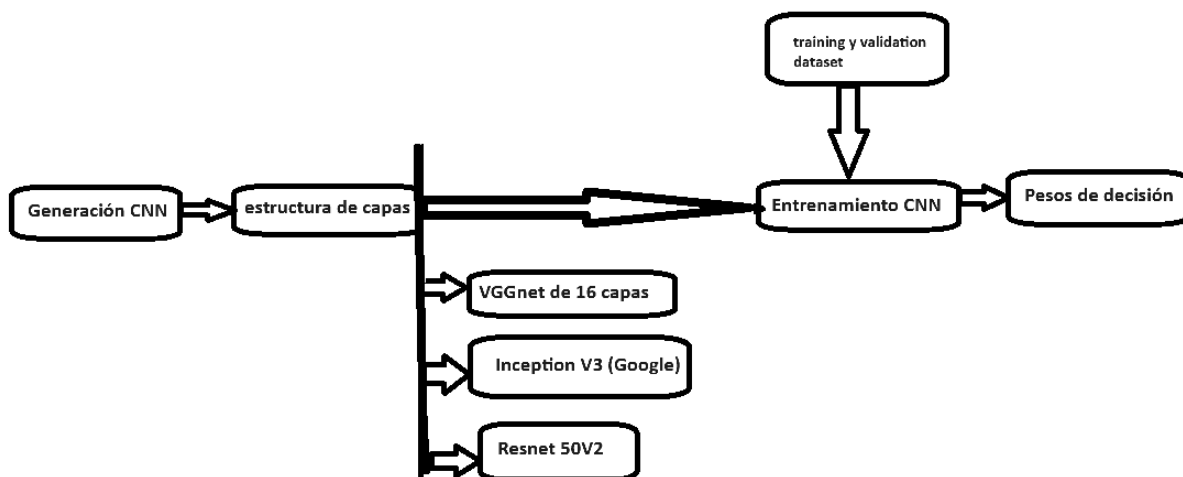


Imagen 1.1 Estructura de Entrenamiento de Redes (fuente propia)

Las tres IAs se basarán en modelos de CNN, y cada una será entrenada individualmente para clasificar imágenes entre las categorías señaladas.

Este proyecto ha tenido tres etapas claramente diferenciadas.

- Descarga y organización de los conjuntos de imágenes de muestra.
- Diseño y entrenamiento de las IAs por separado. (Véase Imagen 1-1)
- Implementación de los modelos de ensamblaje para comprobar si dichos modelos suponen una mejora con respecto al uso de cada una de las IAs por separado.

Como veremos a lo largo del documento, la mayoría los modelos de ensamblaje analizados en este proyecto obtienen mejoras de precisión en la clasificación con respecto a los modelos por separado, sin embargo, algunos de estos modelos consiguen estos resultados en detrimento del balance en las clases minoritarias, esto es, las clases que han obtenido un muestrario menor pueden tener resultados peores en los modelos

ensamblados que en los modelos independientes.

1.1. Estado del Arte

Existen diversos métodos de análisis de imágenes para realizar diagnósticos o ser un apoyo al experto que los realiza.

La mayoría de los métodos predictivos o de diagnóstico de la epidermis no requieren un proceso invasivo para el paciente. Las técnicas de procesamiento de imagen permiten realizar un diagnóstico mediante una imagen de la zona afectada. Dicha imagen permite la identificación y clasificación de enfermedades a través del uso de un sistema de inteligencia artificial.

Las técnicas más utilizadas hoy en día para identificar y diagnosticar anomalías a partir de tecnologías de imágenes radiológicas suelen estar basadas en las redes neuronales artificiales, según [5] y [8] y las redes neuronales convolucionales, según [1], [6] y [9], aunque, para el diagnóstico de enfermedades de la piel, los resultados más precisos se obtienen mediante el enfoque de las redes neuronales convolucionales.

Entre las distintas técnicas para clasificación de imagen destacan los modelos de redes ensamblados, donde cada red intenta emular la respuesta de un experto en la materia, para obtener resultados en base a la combinación de las respuestas de las redes que forman el modelo[22].

Durante 2019, ISIC[1][3][7][10] archive publicó un concurso donde pedía entrenar un modelo de inteligencia artificial para distinguir entre 8 categorías de lesiones de la piel. Los primeros puestos fueron ocupados por modelos de ensamblaje de redes neuronales convolucionales, donde se ensamblan modelos de efficientNet con SENet154[20] o Resnet con SeResNext[21].

1.2. Inteligencia Artificial

La inteligencia artificial trata de emular la inteligencia humana en el ordenador. Los modelos de inteligencia artificial sirven para analizar datos complejos mediante aprendizaje automático. Para ello, los sistemas deben ser entrenados con una serie de entradas previamente catalogadas por expertos, para así poder aprender de ellas y ser capaz de catalogar otras entradas que no hayan sido catalogadas previamente[22].

Entre los distintos modelos de inteligencia artificial se encuentran las redes neuronales. Un modelo de red neuronal consta de una acumulación de capas interconectadas[24]. Estas capas contienen un conjunto de neuronas que siguen la siguiente estructura:

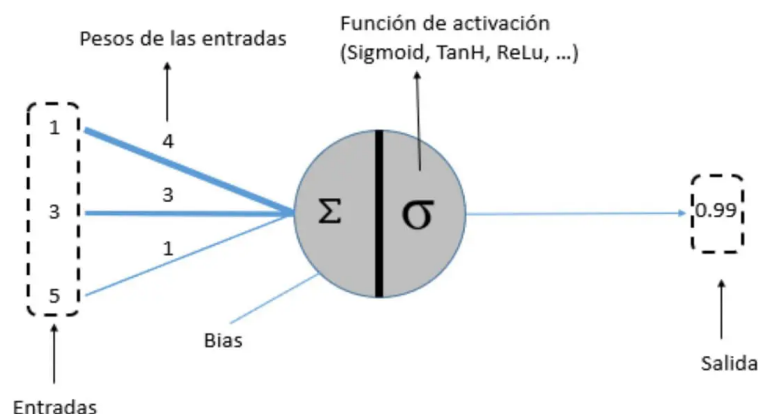


Imagen 1-2: Estructura de una neurona en una IA (fuente: <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>)

Cada capa consta de un número determinado de neuronas y tienen las siguientes características:

La última capa de una inteligencia artificial contiene tantas neuronas como clases pueda discernir. Las salidas de estas neuronas representan la probabilidad de que una entrada arbitraria pertenezca a la clase asociada a la neurona.

El entrenamiento de las entradas permite modificar los pesos de las neuronas que forman el modelo. Esto provoca que el sistema actualice sus pesos para intentar que la salida que produce el sistema sea lo más cercana posible a la salida ideal.

Este proceso requiere una gran cantidad de muestras si se quiere entrenar una red desde cero, del orden de medio millón de muestras, pero es posible reducir la cantidad de muestras necesarias si se tiene un modelo con pesos preentrenados, aunque no tengan nada que ver con las categorías del sistema. A esta técnica se le conoce como “Transfer Learning”.

La función de activación modifica la salida de las neuronas, esto es importante en las últimas capas, siendo necesaria que la última tenga una activación “softmax”, de este modo acotamos la salida entre 0 y 1.

Durante la etapa de entrenamiento el sistema recibe entradas con sus salidas ideales y compara la salida que obtiene con la que recibe. Las entradas de entrenamiento pasan por el sistema varias veces para acercarse cada vez más a la salida ideal, esto puede provocar que el sistema se adapte demasiado a las entradas de entrenamiento y obtenga muy buenos resultados con estas entradas sin llegar a distinguir correctamente entradas ajenas al entrenamiento. A este suceso se le conoce como sobreentrenamiento.

En [fig. 1-3] se puede comprobar cómo sería un clasificador con pesos ideales que le permitan distinguir entre dos clases:

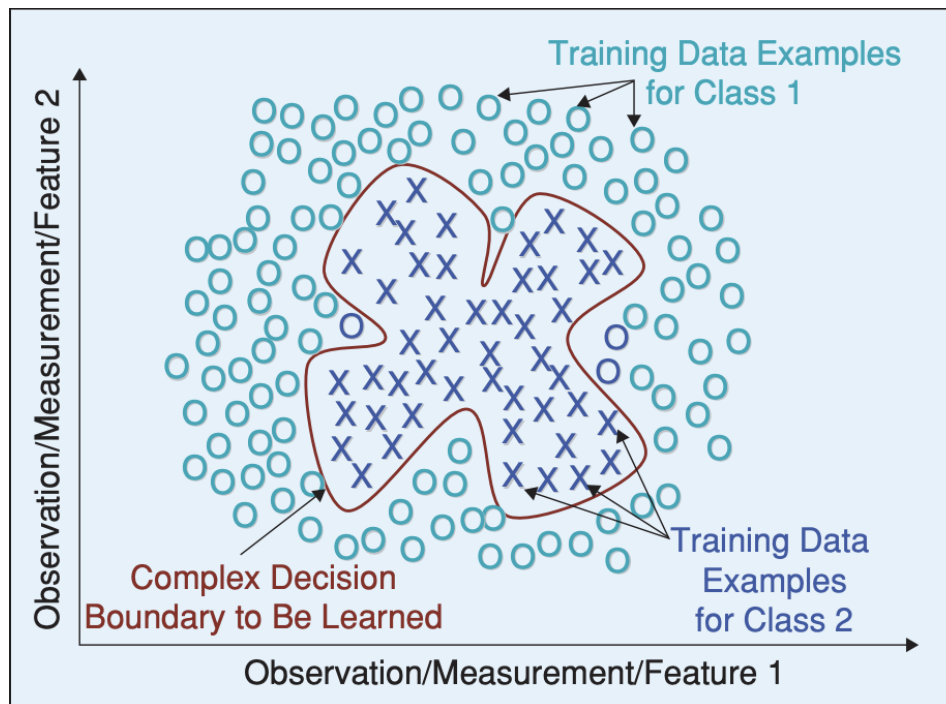


Imagen 1-3: Ejemplo de límites de decisión ideal de un modelo binario. (fuente: [22])

1.3. Redes Neuronales Convolucionales

Entre los distintos tipos de IAs se encuentran las Redes Neuronales Convolucionales (CNN). Estos modelos estructuran cada neurona de una capa de modo que sólo se conecte con un vecindario de la siguiente capa. Además, los pesos se repiten para todas las neuronas de la capa, es decir, todas las neuronas se conectan con su vecindario de la capa siguiente usando los mismos pesos. Por eso, el paso de la información de una capa a la siguiente emula una convolución.

Este modelo permite que las primeras capas de la red funcionen como extractores de características, generando un mapa de características que permite comparar la dependencia de las clases proporcionadas.

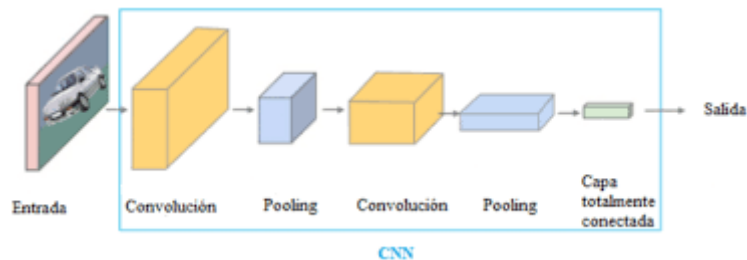


Imagen 1-4: Estructura genérica de una Red Neuronal Convolutional modelo VGG (Fuente: https://www.researchgate.net/figure/Figura-1-Descripcion-del-funcionamiento-de-una-red-neuronal-convolucional-CN-N-10_fig1_348825166)

En este proyecto hemos optado por diseñar y entrenar tres modelos de Redes neuronales convolucionales diferentes para que cada uno genere un mapa de características distinto. Esto lo hacemos para que cada red identifique las imágenes de una forma distinta, lo que nos permite reducir la dependencia entre las distintas predicciones.

Existen múltiples modelos de redes neuronales convolucionales, a continuación se explicará el modelo genérico de cada red diseñada:

1.3.1 Modelos ensamblados

Cada red entre las descritas en los apartados anteriores obtiene un espacio de decisión lineal propio para distinguir entre las clases entrenadas. Tomando como ejemplo [imagen 1-3], el espacio de decisión necesario para distinguir entre las dos clases no es lineal, siendo muy difícil obtener un modelo con un espacio de señal cercano al óptimo.

En esta situación destaca los modelos de redes ensambladas, donde cada red obtiene un espacio de señal lineal distinto. La combinación de los distintos espacios de señal pueden alcanzar un modelo cercano al óptimo, como se puede observar en [imagen 1-8].

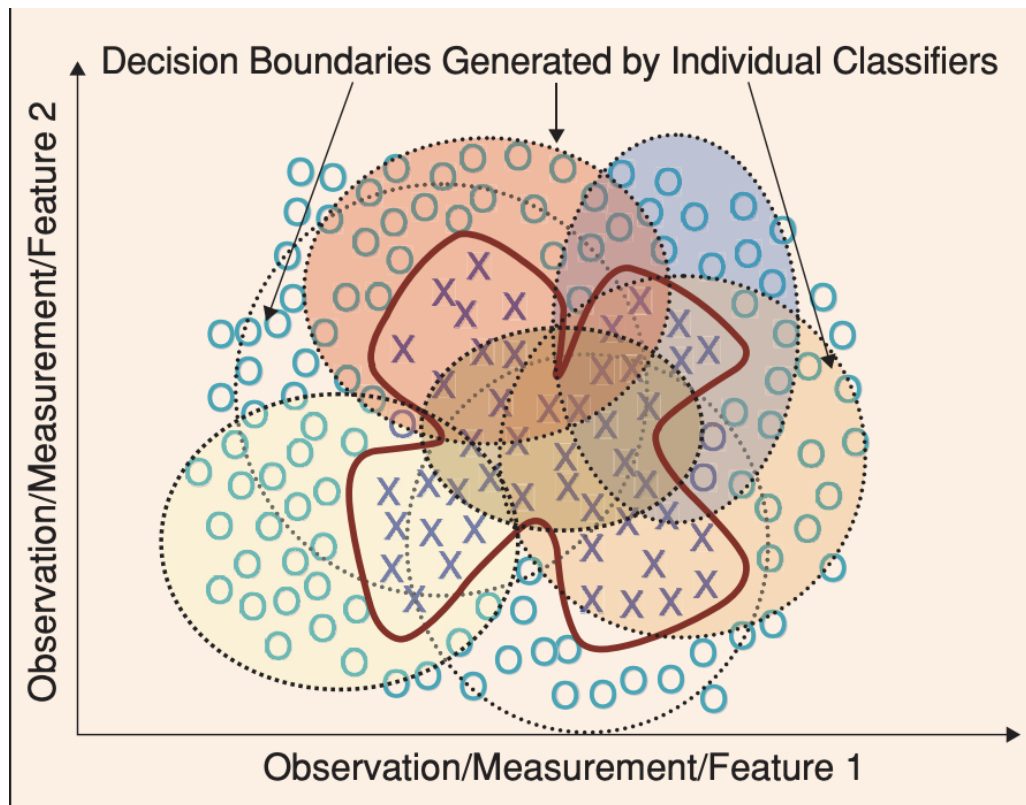


Imagen 1-8: Ejemplo de modelo ensamblaje basado en el espacio de señal de [imagen. 1-3], donde cada espacio coloreado representa el espacio de decisión de las redes que forman el modelo. (fuente: [22])

Entre los distintos métodos para ensamblar redes se pueden distinguir en dos tipos según [22], los que combinan las votaciones de las redes que forman el modelo y los que combinan las salidas de dichos modelos.

Las técnicas de ensamblado son muy efectivas además para un problema muy común en cualquier modelo de aprendizaje máquina, el desbalance de clases [24]. La imagen 1-9 explica perfectamente qué es este problema:

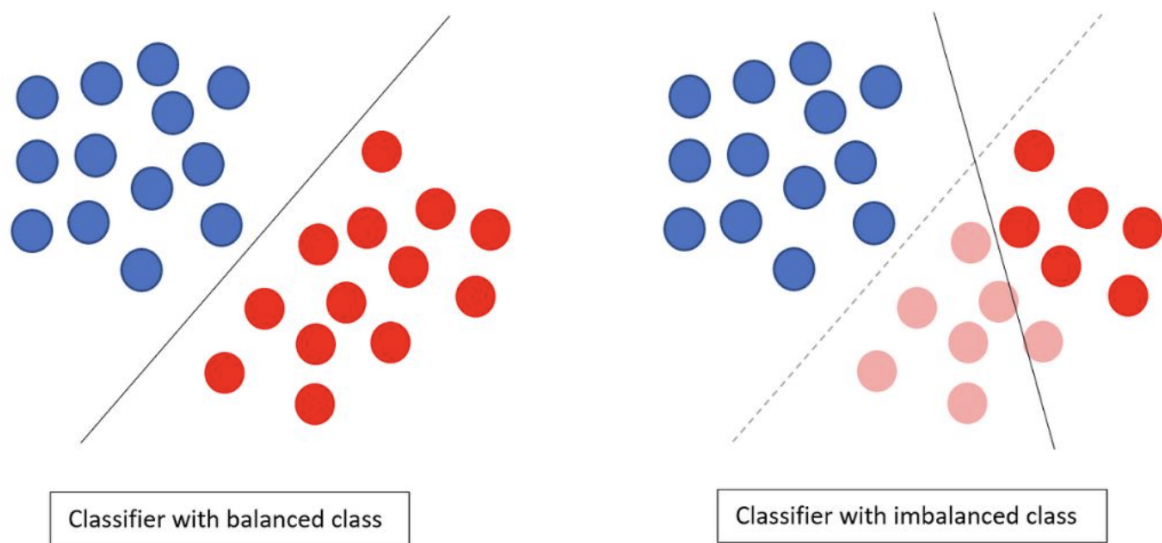


imagen 1-9: Representación del problema de desbalance de clases.

(fuente: [24])

1.4. Lesiones dermatológicas

Las ocho categorías de imágenes médicas que se manejan en el proyecto se pueden dividir, a su vez, en tres subcategorías, según el impacto en el paciente. Pasamos a describirlas a continuación.

1.4.1. Trastornos Benignos

Se llama así al tipo de lesiones que no suponen un riesgo para la salud del paciente, por lo que no es necesaria ninguna intervención quirúrgica o médica. Suele ser el caso de pecas o lunares no invasivos. Dentro de este tipo de trastornos se encuentran las siguientes categorías:

1.4.1.1. Angioma

Tumores benignos relativamente comunes en recién nacidos causados por una agrupación de vasos sanguíneos.[15]

1.4.1.2. Nevus

Nombre médico de las pecas, lesiones benignas que se encuentran en la mayoría de la población.[1]

1.4.1.3. Queratosis Benigna

Conjunción de tres lesiones benignas en las que se encuentran el lentigo solar, queratosis seborreica y queratosis liquenoide.

1.4.2. Lesiones Malignas

Estos trastornos suponen un riesgo para la salud del paciente si no son diagnosticados en una etapa temprana e intervenido por expertos y cirujanos. Estos tipos de trastornos comprenden las siguientes categorías:

1.4.2.1. Melanoma

Lesión maligna formada en las células que se encargan de la pigmentación de la piel (Melanocitos), también puede manifestarse en ojos y raramente en el interior, como en la nariz o garganta.[12]

1.4.2.2. Carcinoma Basocelular

Lesión de las células basales producida por un daño a la cadena ADN que puede tener un aspecto muy diferente de una persona a otra, aunque no suelen ser contagiosas. Si bien esta lesión no amenaza directamente la vida del paciente, si no es tratada puede extenderse localmente y atravesar incluso tejido óseo.[16]

1.4.2.3. Carcinoma de Células Escamosas

Lesión provocada por exceso de exposición a los rayos UV. Esto puede provenir tanto de los rayos solares como de una cama solar. El cuerpo intenta reparar las células, pero aquellas que no pueden ser reparadas sufren mutaciones en forma de cáncer de piel.[14]

1.4.3. Trastornos Intermedios

Estos tipos de trastornos se llaman así porque por sí mismos no tienen por qué representar un problema inmediato para el paciente, pero pueden derivar en una enfermedad maligna si no se trata adecuadamente. En esta categoría también se incluyen algunos trastornos congénitos. En estos casos, se sugiere que un experto analice el resultado obtenido y trate con el paciente de manera personalizada para determinar un procedimiento específico. Dentro de los trastornos intermedios se incluyen las siguientes categorías:

1.4.3.1. Lesiones Vasculares

Diversos defectos congénitos o que aparecen al poco de nacer. Estos varían mucho en naturaleza y gravedad, aunque la mayoría suelen solucionarse sin necesidad de intervención.[17]

1.4.3.2. Queratosis Actínica

Manchas contagiosas de la piel producidas por un exceso de exposición a rayos UV. Si estas manchas no se tratan, algunas de ellas pueden evolucionar a carcinoma de células escamosas. [13]

2 METODOLOGÍA DE PROYECTO

Este proyecto tiene como objetivo la generación de un predictor complejo de imágenes dermatológicas formado por tres predictores simples cuyas combinaciones permiten obtener una mejora con respecto a los predictores simples.

Para ello, hemos optado por diseñar tres redes neuronales, entrenadas por separado, cuyas salidas serán vectores probabilísticos cuyos elementos representan la posibilidad de que la entrada pertenezca a la clase asociada.

Estos vectores se combinarán después para obtener un vector probabilidad conjunta por cada modelo de ensamblaje que analizaremos.

De esta manera, el presente proyecto consta de tres partes diferenciadas:

- La generación y estructuración de las imágenes de entrenamiento, prueba y validación.
- La generación y entrenamiento de las redes neuronales.
- El diseño y posterior análisis de los modelos de ensamblaje en función de las métricas recomendadas.

A continuación pasamos a comentar cada una de estas partes.

2.1 Base de datos

La base de datos que se utilizará ha sido obtenida del ISIC[1][3][7][10]. Dicha base de datos consta de un cliente que funciona en Python, con el cual podemos descargar tanto las muestras que vamos a utilizar para el proyecto como los metadatos que nos permiten saber qué diagnóstico, hecho por un experto, tienen las imágenes que usaremos de muestra. Esta información se utilizará para indicar a las CNNs a qué clase pertenece cada imagen de muestra que reciba. Además, será fundamental para obtener los datos de True Positive, False Positive, True Negative y False Positive, necesarios para las métricas recomendadas por el concurso y la matriz de confusión del clasificador.

Esta base de datos contiene una cantidad limitada de imágenes para cada clase y como veremos posteriormente, será necesaria aplicar técnicas de aumento de muestras (Data Augmentation) para poder entrenar correctamente una de nuestras CNNs. Concretamente, se ha multiplicado el número de imágenes para las clases VASC, NEV, BK y CBC desde un x8 para las dos primeras hasta un x20 para las últimas, para así poder equiparar el número de muestras de estas clases con las otras.

Tras aplicar dicha técnica de aumento de datos, hemos tenido acceso a 16.649 imágenes dentro del conjunto de entrenamiento, 4.750 imágenes de validación y 2.375 imágenes de test.

Dichos conjuntos de datos vendrán organizados dentro de la función “Datasets.py” donde aprovecharemos que las imágenes están separadas en carpetas distintas según su clase con la función ‘image_dataset_from_directory()’, que establecerá los sets de datos con una semilla preestablecida. De este modo, siempre que se llame a la función devolverá los datasets con las mismas imágenes. Esto permitirá entrenar las CNNs por separado con las mismas imágenes de entrenamiento y no contaminar el set de testeo con imágenes entrenadas previamente.

2.2 Procesamiento de datos

Todo lo anterior se hace para obtener tres conjuntos de datos (train, validation y test) que contienen tanto las imágenes (train_X, val_X y test_X) como la clase asociada a cada imagen (train_Y, val_Y y test_Y) que serán necesarias para el postprocesado y análisis estadístico.

2.3 Redes Neuronales

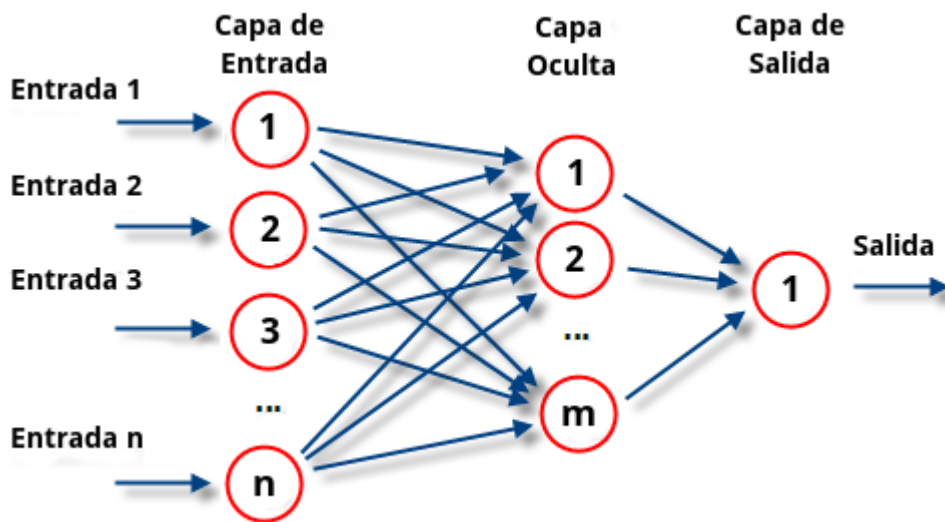


Imagen 2-1: Estructura general de una red neuronal (fuente: wikipedia).

Este proyecto usa modelos con pesos preentrenados gracias al método de transfer learning, esto permite entrenar las redes con un número de muestras mucho menor al necesario para entrenarlas desde cero.

Estas redes estudian los metadatos del sistema a partir del conjunto de datos de entrenamiento. Es decir, en base a las imágenes de entrenamiento (Train_X) buscan los parámetros que interfieren en mayor medida con el resultado que los diagnósticos (Train_Y) asociados a dichas imágenes.

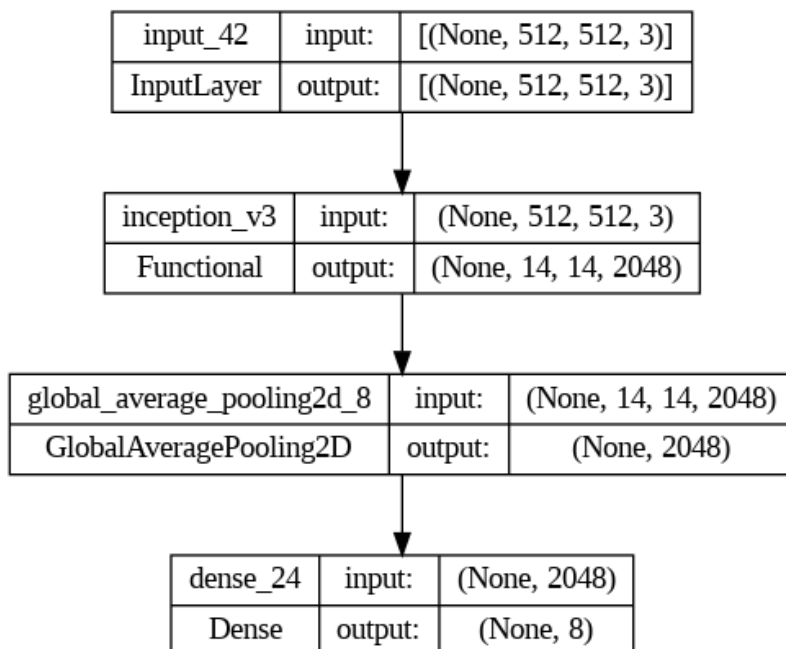


Imagen 2-2: Estructura de la red Inception

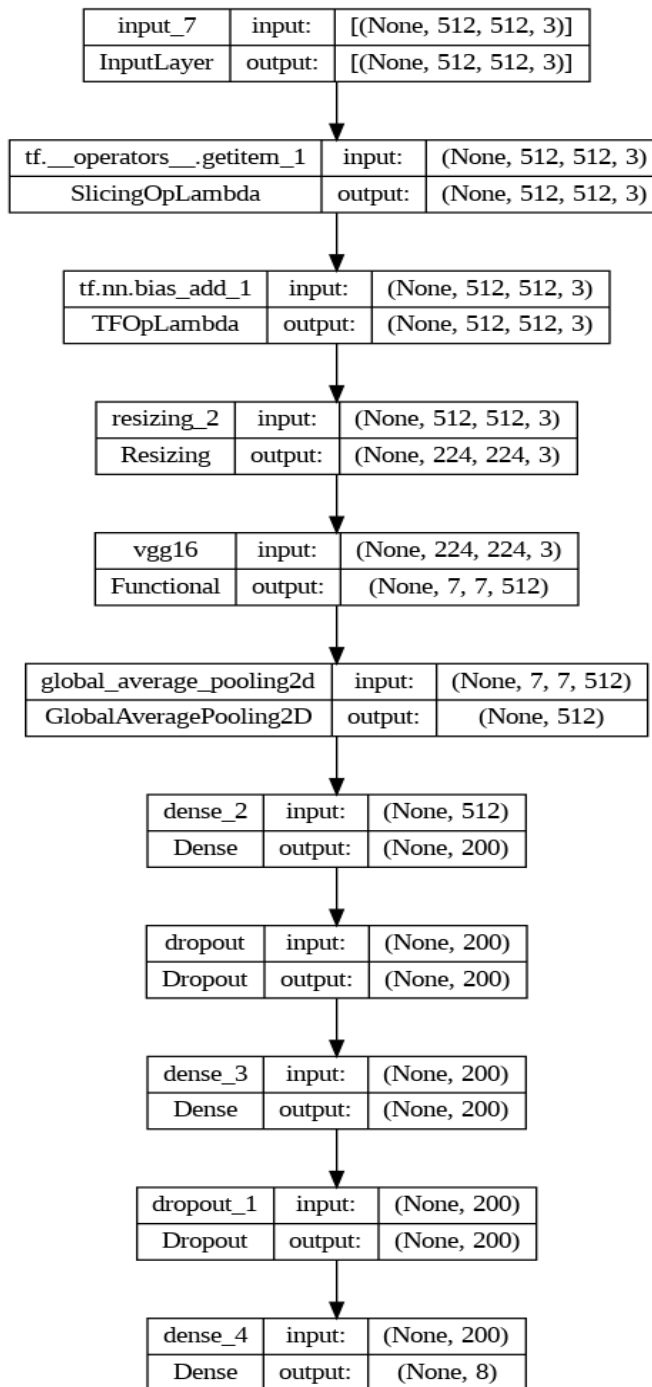


Imagen 2-3: Estructura de la red VGG

2.3.1 Capas de la CNN

Los entrenamientos de estas redes suelen dividirse en 4 etapas claramente diferenciadas: las capas de convolución, max pooling, flattening y fully connected.

- Las capas de convolución son las encargadas de escoger las características, de hacer un mapeo de las entradas y de guardar las coincidencias para priorizar unas características frente a otras.
- Las capas Pooling son las encargadas de agrupar información de las entradas que le llegan, de modo que la salida de dichas capas será de menor tamaño que la entrada correspondiente.
- Las capas Flattening son las capas complementarias a las de Pooling y se encargan de combinar diferentes canales de la entrada. Por ejemplo, para diferenciar entre las capas Pooling y Flattening en una matriz $[100 \times 100 \times 2]$, al pasar dicha matriz por una capa MaxPooling, la salida pasaría a ser $[50 \times 50 \times 2]$. En cambio, si la misma matriz pasa por una capa flattening, la salida será $[100 \times 100 \times 1]$.
- Las capas fully connected (FC) necesitan de un vector de entrada de una dimensión. Este vector se adapta desde la entrada gracias a la combinación de las capas de flattening y pooling. La capa FC se encarga de conectar cada componente del vector de entrada con cada componente de su vector de salida. Modificando el parámetro de activación no lineal de las capas FC podemos establecer el tipo de salida que tendrá dicha capa, de modo que toda CNN acaba en una capa FC con activación “Softmax” con tantas neuronas como clases a distinguir para hacer la clasificación, de modo que cada neurona representa la probabilidad de que la imagen clasificada pertenezca a la clase asociada.

2.3.2 Etapa de entrenamiento

Una vez presentada la estructura de las CNNs, vamos a hacer unos comentarios sobre el entrenamiento del sistema. El objetivo de la red neuronal es establecer una salida en función de una serie de entradas arbitrarias, por lo que será necesaria durante la etapa de entrenamiento, en la que se requiere un gran número de muestras (de un orden mayor a 500.000 entradas), que éstas estén clasificadas previamente por expertos.

Durante la etapa de entrenamiento se desarrolla un mapa de características y una matriz de coocurrencia, que intenta alcanzar el punto óptimo de salida en función de las entradas proporcionadas. Así se consigue que la CNN pueda clasificar correctamente entradas para las que no ha sido entrenada.

Desafortunadamente, alcanzar el punto óptimo de un clasificador no es una tarea trivial. Por ello, introducimos la variable de metacaracterísticas denominada “Epoch”, que indica el número de ciclos que las entradas de entrenamiento pasan por el sistema, de forma que durante cada uno de estos ciclos, el sistema pretende acercarse un paso hacia el punto óptimo.

En este punto es importante incidir en el sobreentrenamiento, que tendrá lugar si el número de epochs es tan alto que el sistema se sobreadapta a las muestras dadas, generando así una falsa estadística de precisión en la aprobación del sistema.

Para evitar este problema de sobreentrenamiento existen puntos de control que revisan los parámetros durante la etapa de entrenamiento, usando el conjunto de imágenes de validación (val_X). Las IAs no aprenden de este conjunto, de modo que el decisor final será aquel que, aprendiendo del conjunto de entrenamiento, maximiza la precisión del conjunto de validación.

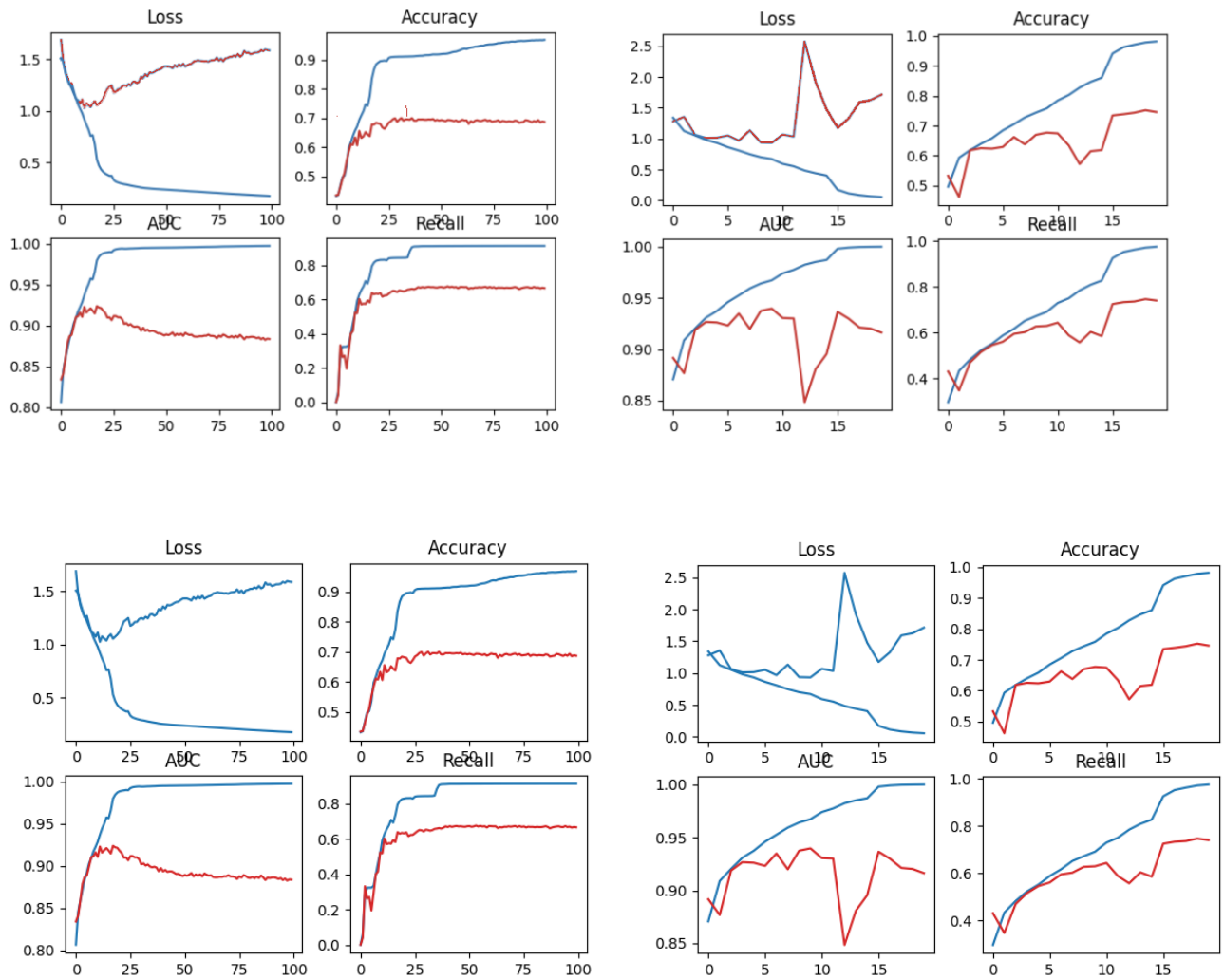


Imagen 2-4: Evolución de métricas con los ciclos (Rojo: imágenes de validación, Azul: imágenes de entrenamiento) (izquierda: completo, derecha: con EarlyStopping)

2.3.3 Diseño de las CNNs

En este proyecto nos apoyaremos en la librería Keras de Python para desarrollar las tres redes neuronales. Keras es una biblioteca de código abierto especializada en la generación de modelos de inteligencia artificial cuyo objetivo es acelerar la creación de redes neuronales.

Existen múltiples estructuras prediseñadas que la librería Keras pone a nuestra disposición para no tener que diseñar capa a capa las redes neuronales. Estas estructuras tienen la ventaja de que pueden venir con pesos preentrenados, gracias a la técnica de transfer learning, provocando que el número de muestras necesaria para el entrenamiento de dicha red neuronal sea mucho menor al que sería necesario para crear la red neuronal desde cero.

Por nuestra parte, hemos usado las siguientes estructuras: VGG, Inception y ResNET, tres modelos que han demostrado tener buen desempeño en la clasificación de imágenes dermatológicas.

2.3.3.1 VGGnet de 16 capas

La primera red neuronal que se ha diseñado para el sistema es una red basada en la estructura VGG, que actualmente se divide en VGG16 o VGG19 según el número de capas que contenga, como se muestra en la siguiente imagen:

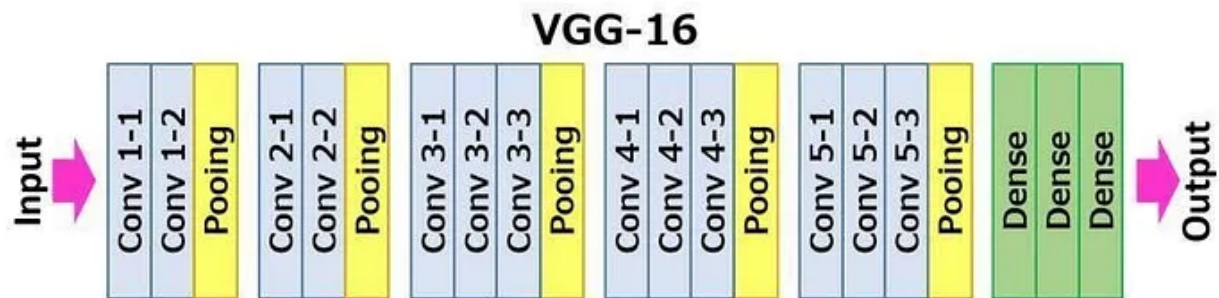


Imagen 1-5: Estructura genérica VGG16 (Fuente:

<https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>)

Esta estructura tiene una alta carga en la primera etapa de capas de red. Es decir, contiene muchas capas de convolución+ReLU, por lo que tendrá un mapa de características muy amplio. Esto quiere decir que su decisor tendrá una alta influencia en factores morfológicos de la imagen.

Dicha red procesa las imágenes de forma óptima cuando está en formato BGR, en contraste con el modelo clásico RGB, por lo que se ha añadido una capa de preprocesamiento a la red que permite transformar cualquier imagen de entrada al formato adecuado.

En la figura 4-3 se puede ver la estructura de capas de la red neuronal, siendo representada la capa funcional VGG16 en la figura 4-5.

2.3.3.2 Inceptionv3 de Google

La segunda red neuronal que utilizamos en el sistema es InceptionV3, de Google. Este modelo fue diseñado como modelo de reconocimiento de imágenes con el objetivo de estudiar parámetros más complejos a los de VGG, su estructura viene detallada en la siguiente imagen:

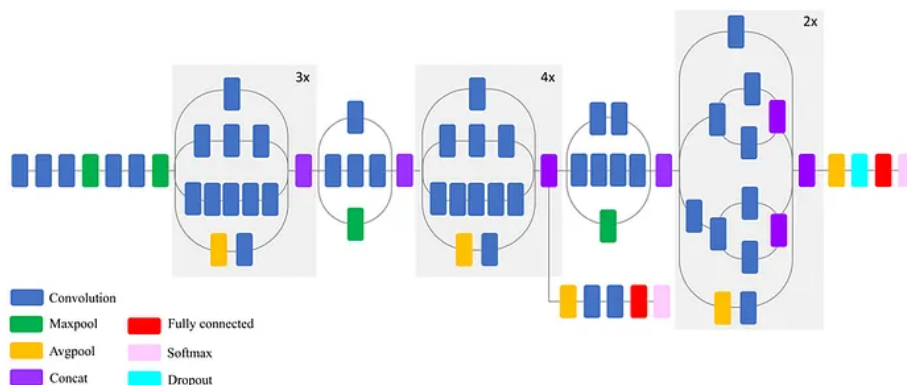


Imagen 1-6: Estructura de InceptionV3 (fuente:

<https://medium.com/@792/the-inceptionv3-experiment-a-shallow-sight-on-transfer-learning-74e8ef4a4533>)

Hemos descargado el modelo genérico de Keras, sin incluir las últimas capas de flattening y fully connected, pues el sistema requiere de un fine tuning específico. En otras palabras, vamos a crear nosotros las últimas capas, para transformar la salida de Inception en un vector de 8 elementos (uno por cada clase).

2.3.3.3 RESNET50 v2

La tercera red neuronal que hemos utilizado es Resnet50V2, que está dando muy buenos resultados para la clasificación de imágenes médicas, véase [1][6].

Este modelo se basa en una red neuronal residual. Esto es, las capas contemplan tanto el mapa de características generado por el conjunto de capas anteriores como la salida de las mismas capas, generando una estructura como la mostrada en la siguiente imagen.

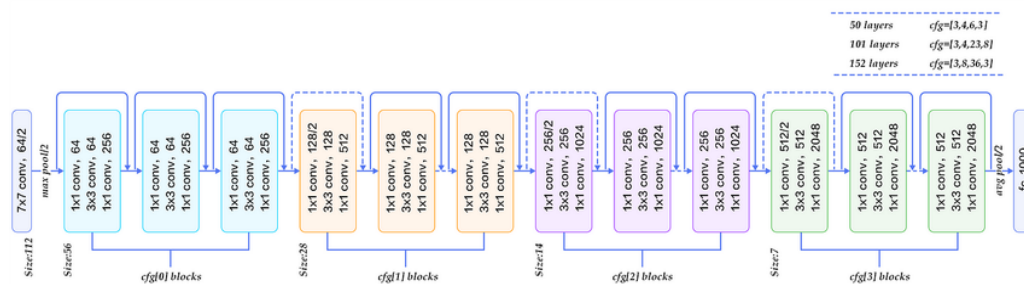


Imagen 1-7: Estructura de Resnet (fuente:

https://www.researchgate.net/figure/Diagram-of-ResNet50V2-architecture-11_fig5_373570488)

En este caso, al igual que en el anterior, al ser un modelo genérico con pesos preestablecidos, se requiere de un fine tuning en el que pasa por una capa Global Average Pooling, para poder adaptar mejor desde los pesos preentrenados hasta los pesos óptimos requeridos por el sistema.

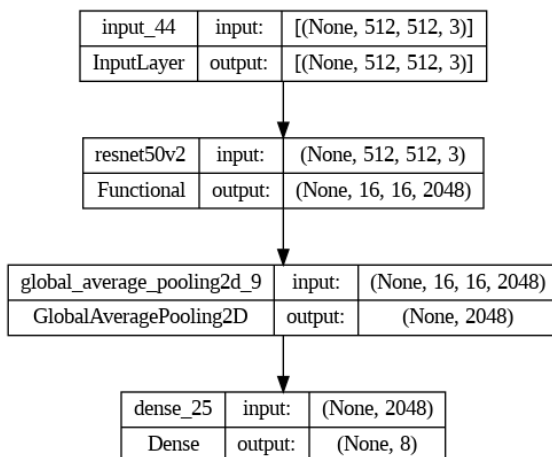


Imagen 2-5: Estructura de capas de la red RESNET

2.4 Técnicas de ensamblaje

La salida de una red neuronal convolucional es un vector con tantos elementos como categorías es capaz de distinguir dicha red, en nuestro caso son ocho categorías. Esto hace que la entrada a nuestro sistema de ensamblaje esté formada por tres vectores con ocho elementos cada uno.

Las técnicas de ensamblaje que vamos a diseñar y analizar están basadas en distintas técnicas de decisión utilizados en visión artificial y en algunos sistemas de comunicaciones.

2.4.1 Ensamblaje por mayoría simple

Esta técnica se basa en la adición de los vectores de salida de las redes. Para ello, cada elemento de los vectores de salida está acotado entre 0 y 1. Estos elementos representan la probabilidad de que la entrada pertenezca a la clase correspondiente a la posición de dicho elemento.

El vector de salida final, por tanto, está formado por ocho elementos que siguen la siguiente fórmula:

$$P_{total|clase} = (P_{VGG|clase} + P_{Inception|clase} + P_{Resnet|clase})/3 \quad \text{Ec. 1}$$

Siendo P total por clase, la probabilidad conjunta de que la entrada pertenezca a dicha clase.

El decisor que sale de esta técnica será aquel cuya probabilidad de pertenencia a una clase sea máxima con respecto a las demás.

La división entre el número de redes independientes la mostramos para que sea más fácil ver la interpretación de los elementos del vector de salida, ya que el argumento máximo de un vector de 8 elementos da el mismo resultado que el argumento máximo del mismo vector dividido por el mismo número.

2.4.2 Ensamblaje por mayoría cuadrática

La segunda técnica de ensamblaje que pondremos en práctica se basa en técnicas de reducción de ruido de algunos sistemas multimedia. En esta técnica aprovechamos que los vectores probabilidad de los sistemas independientes están escalados entre 0 y 1. Bajo esta premisa, elevamos al cuadrado cada elemento de los vectores independientes antes de sumarlos entre ellos.

Con esto conseguimos que aquellos valores cercanos a 0 se vuelvan despreciables, sin que aquellos valores cercanos a 1 se vean muy trastocados. La cota de estos nuevos vectores no cambia, pues uno al cuadrado sigue siendo uno, al igual que el cero.

Tras aplicar esta técnica de reducción de ruido, hemos conseguido que los pesos de decisión se vean menos afectados por aquellas probabilidades que sean más cercanas a cero.

2.4.3 Ensamblaje por minimizador de distancia

Esta técnica de ensamblaje está inspirada en técnicas de decisión por Machine Learning, donde el decisor aprende un modelo prototipo denominado Atlas, formado por la salida media que debiera obtener por cada clase. Al hacer esto, basamos el decisor del sistema en un minimizador de distancia euclídea de la entrada recibida con respecto a dicho valor de salida medio.

Dicho valor medio de salida debe tener información desconocida, es decir, debemos tener cuidado con el conjunto de datos que usamos para obtener dicho valor. Esto se debe a que la salida del conjunto de entrenamiento es conocida, por lo que no genera información útil.

2.4.4 Ensamblaje Neuronal

La última técnica de ensamblaje que se va a diseñar y analizar es un modelo de inteligencia artificial, formado por la concatenación de las salidas de las tres redes independientes en capas *Fully Connected*, para así generar un modelo complejo donde aproveche estas últimas capas para estudiar la dependencia de las redes.

Para esta técnica ha sido necesaria una nueva etapa de entrenamiento con pocas *Epochs*, para que el sistema no sobreaprenda el conjunto de entrada.

2.5 Métrica utilizada

La génesis de este proyecto se encuentra en el concurso propuesto en 2019 por la Sociedad International Skin Imaging Collaboration (ISIC)[1][3][7][10], que puso a disposición de los participantes un total de 25.331 imágenes dermatológicas para hacer una clasificación dermoscópica entre las clases descritas en esta memoria anteriormente. Aprovecharemos esta base de datos como muestrario para entrenar nuestras redes neuronales.

La organización tomó como principal métrica la precisión multiclase normalizada, ordenando a los concursantes según dicha métrica. No obstante, por interés científico, solicitó que se añadieran varias métricas secundarias, que usan una serie de variables basadas en los resultados obtenidos por el detector.

Para obtener dichas métricas, hay que pasar las muestras de validación por el sistema predictor para poder hacer un estudio estadístico de los resultados.

Para aquellos casos en los que la clase de una imagen coincida con la predicción del sistema se les llama verdaderos positivos (TP).

Para los casos en los que la detección predicha no coincida con la clase de la imagen correspondiente se les denomina falso positivo (FP), pues está dando por positivo un diagnóstico incorrecto, siendo la clase adecuada de dicha imagen un falso negativo (FN).

Para el resto de casos, en los que una clase no es la de la muestra, pero tampoco ha sido diagnosticada, como pasará en la mayoría de los casos, se consideran verdaderos negativos (TN).

Por último, se cuentan los positivos totales (P) y los negativos totales (N) para poder obtener las métricas que cumplen las siguientes ecuaciones:

$$\text{Precisión multiclase normalizada} = \sum([TP/P]|clase)/Nclases \quad \text{Ec. 2}$$

$$\text{Sensitividad} = TP/(TP + FN) \quad \text{Ec. 3}$$

$$\text{Especificidad} = TN/(TN + FP) \quad \text{Ec. 4}$$

$$F1 \text{ Score}(Dice \text{ score}) = 2 * TP/(2 * TP + FP + FN) \quad \text{Ec. 5}$$

$$\text{Valor de predicción positivo}(PPV) = TP/(TP + FN) \quad \text{Ec. 6}$$

$$\text{Valor de predicción negativo}(NPV) = TN/(TN + FP) \quad \text{Ec. 7}$$

El análisis del desempeño del sistema se llevará a cabo utilizando estas métricas.

2.5.1 Precisión Multiclase Normalizada (BMA)

De entre las distintas métricas la más importante es la precisión multiclase normalizada. Esta métrica se usa en distintos tipos de clasificadores. Representa la cantidad de positivos reales que ha conseguido en un conjunto de clasificación sobre el total de positivos que ha clasificado por cada clase.

Esta se obtiene aplicando [Ec. 2], que mide la probabilidad de que una predicción sea realmente lo que dice ser.

Este valor oscila entre 0 y 1 y representa la probabilidad, en tanto por 1, de la fiabilidad del sistema.

Un valor cercano a 0 para esta métrica implica que la máquina confunde mucho las clases entre sí, mientras que un valor cercano a 1 implica que el sistema es capaz de diferenciar claramente entre las distintas clases.

2.5.2 Sensibilidad y Especificidad

Estas métricas representan la capacidad que tiene el sistema de distinguir entre los positivos de los negativos. Estas métricas se obtienen mediante [Ec. 3] y [Ec. 4].

A la sensibilidad también se le conoce como “Recall” y es otra métrica muy utilizada en el análisis de sistemas predictivos.

La sensibilidad es la probabilidad de distinguir una clase cuando la entrada pertenece a dicha imagen. La especificidad es la probabilidad de que el sistema no categorice una entrada como una clase a la que no pertenece.

Los valores de estas métricas oscilan entre 0 y 1. Un valor alto en estas métricas indica que el sistema tiene pocos fallos en sus predicciones, mientras que un valor bajo implica que se ha equivocado mucho en comparación con los fallos totales presentes.

2.5.3 F1 Score

Esta métrica se obtiene de la precisión y la sensibilidad, siguiendo [Ec. 4]. Un valor alto en esta métrica indica que tanto la sensibilidad como la precisión tienen un valor alto.

2.5.4 Valores Predictivos (Positivo y Negativo)

Estos valores representan la probabilidad intrínseca de un sistema, es decir, el valor predictivo positivo es la probabilidad de que una entrada pertenezca a una clase, habiendo predicho dicha clase.

El valor predictivo negativo representa la probabilidad de que una entrada no pertenezca a una clase, cuando el sistema ha predicho que no pertenece a dicha clase.

2.6 Matriz de Confusión

Para obtener los valores de TP, TN, FP, FN, P y N, nos basaremos en matrices de confusión. Este tipo de matrices contiene tantas filas y columnas como clases hay en nuestro modelo de predicción y permite medir cómo se comporta el modelo con muestras para las cuales no ha sido entrenado.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Imagen 2-6: Distribución de los elementos de la matriz de confusión. (fuente: <https://rpubs.com/chzelada/275494>)

Cada modelo de predicción tendrá asociado una matriz de confusión propia donde se reflejarán los resultados de predicción del conjunto de imágenes de prueba.

La formación de las matrices de confusión se obtiene durante la etapa final del proyecto. Cada modelo hará una predicción del conjunto de imágenes que no han intervenido durante la etapa de entrenamiento.

Tendremos, por tanto, tres predicciones, a partir de las cuales se recuentan los casos siguiendo la distribución de la imagen 3-1. Desde un punto de vista de la programación, hemos utilizado los índices de una matriz [8x8] cuyos elementos representan el número de veces que cada imagen de la clase observada (es decir, categorizada por expertos) ha sido catalogada como clase posible.

Aunque más adelante se hablará de la interpretación de resultados, expondremos los resultados de uno de los modelos a modo explicativo:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	591	0	36	31	9	15	3	9
CBC	9	10	5	18	6	3	0	0
Derm	28	1	960	279	35	8	4	6
Mel	32	0	135	1789	33	0	1	0
Nev	20	0	36	64	192	2	0	4
BK	49	1	23	16	7	70	0	8
SCC	5	0	1	3	1	0	28	0
Vasc	22	1	18	7	26	8	0	70

Tabla 2-1: Matriz de confusión del modelo cuadrático, enfatiza los valores de predicción de Melanomas. Azul: Falsos Negativos, Naranja: Falsos positivos, Verde: Verdaderos Positivos, Gris: Verdaderos negativos, Azul oscuro: Clases predichas, Rojo: Clases Observadas.

En esta matriz de confusión para la clase “Melanoma” hemos obtenido 1789 verdaderos positivos, 418 falsos positivos, 201 falsos negativos y 2330 verdaderos negativos. Esto hace un total de 2207 positivos y 2531 negativos.

A partir de estos datos podemos calcular las ecuaciones [2-7] obteniendo los siguientes resultados:

$$Sensitividad_{Melanoma} = 0.898995 = 89.8995\%$$

$$Especificidad_{Melanoma} = 0.847889 = 84.7889\%$$

$$Precisión_{Melanoma} = 0.869354 = 86.9354\%$$

$$F1\ Score_{Melanoma} = 0.852514 = 85.2514\%$$

$$PPV_{Melanoma} = 0.898995 = 89.8995\%$$

$$NPV_{Melanoma} = 0.847889 = 84.7889\%$$

Para obtener la métrica principal, hay que reunir el valor de precisión para cada clase y hacer una media con todos los valores.

Una métrica que no se ha tenido en cuenta en este proyecto es la precisión global del sistema, donde se calcula la precisión de todo el conjunto de imágenes sin discernir entre clases. Esta métrica sirve, sobre todo, para sistemas de predicción binarios, pues puede proporcionar resultados engañosos para otros casos. Por ejemplo, si el sistema aprende mucho de una clase que contiene más muestras que las demás, puede obtener resultados de precisión más altos de los que debería aunque no haya aprendido nada de otras clases.

2.7 Obtención de resultados

Para las diversas combinaciones de los tres modelos de predicción, obtendremos la correspondiente matriz de confusión y métricas.

A partir de la matriz de confusión se puede ver qué predicciones ha hecho el sistema para cada clase, además de cuántas de estas predicciones han errado y en qué clases ha fallado.

Las métricas ayudan a ver el comportamiento del sistema por clases, indican probabilidades de acierto, fiabilidad, etc, y permiten visualizar de forma fácil dichas probabilidades.

De este modo podemos establecer el comportamiento del sistema en base a los resultados obtenidos.



3 RESULTADOS

A continuación procederemos a exponer los resultados obtenidos en los distintos modelos.

Para ello y como hemos comentado, nos hemos apoyado en la matriz de confusión y las métricas explicadas para entender el desempeño del sistema en cuestión.

3.1 Matriz de Confusión y análisis estadístico

Como comentamos anteriormente, disponemos de un total de 40.433 imágenes después de la aplicación de técnicas de Data Augmentation. Las hemos dividido en 23.784 imágenes de entrenamiento, 11099 imágenes de validación y 5.550 imágenes de prueba.

Las técnicas de Data Augmentation se han aplicado exclusivamente a 4 clases debido a la baja cantidad de muestras disponibles. Esto ha permitido que la red VGG sea entrenada correctamente, pues sin dichas técnicas la red sería incapaz de aprender sobre cuatro de las ocho clases presentes, afectando no sólo a su propia métrica sino al rendimiento de todo el sistema de ensamble.

3.1.1 Métricas

Una vez entrenadas las redes neuronales, obtendremos la denominada matriz de confusión, una matriz que relaciona las predicciones y las clases originales de un conjunto de datos:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	694	0	0	0	0	0	0	0
CBC	0	51	0	0	0	0	0	0
Derm	0	0	1321	0	0	0	0	0
Mel	0	0	0	1990	0	0	0	0
Nev	0	0	0	0	318	0	0	0
BK	0	0	0	0	0	174	0	0
SCC	0	0	0	0	0	0	38	0
Vasc	0	0	0	0	0	0	0	150

Tabla 3-1: Matriz de confusión ideal

La parte sombreada en azul se refiere a la clase original del conjunto, la parte sombreada en verde hace referencia a la clase predicha por el decisor, los recuadros naranjas cuentan el número de imágenes donde el decisor ha predicho que sea la misma clase que la original y los recuadros rojos cuentan el número de veces que el decisor ha predicho una clase siendo ésta distinta a la original.

A partir de la tabla de confusión descrita anteriormente, podemos obtener los valores de TP, TN, FP, FN, P y N necesarios para analizar la métrica establecida por el concurso.

El valor de TP se obtiene por cada clase. Éste corresponde con la celda naranja correspondiente a cada clase, es decir, las veces que el decisor ha coincidido con la clase original. El valor completo de TP corresponde al total de la suma de la diagonal de la matriz de confusión.

El valor de FP se obtiene por clase, y corresponde a la suma de todas las celdas rojas de cada columna, o lo que es lo mismo, la suma de cada columna restándole el valor correspondiente del recuadro naranja.

El valor de FN es el análogo a FP, corresponde a la suma de las celdas rojas de cada fila.

Por último, para obtener el valor de TN debemos quedarnos con la suma de todos los elementos de la matriz de confusión menos las celdas que hayan interferido con cada clase, o lo que es lo mismo:

$$TN = MatrizConfusion - TP - FN - FP \quad \text{Ec. 8}$$

```
for i in range(C_original):
    TP=np.sum(M_Confusion[i,i])
    FP=np.sum(M_Confusion[i,:])-TP
    FN=np.sum(M_Confusion[:,i])-TP
    TN=np.sum(M_Confusion)-TP-FP-FN

    sensibilidad=TP/(TP+FN)
    especificidad=TN/(TN+FP)
    precision=TP/(TP+FP)
    f1_score=2*((precision*sensibilidad)/(precision+sensibilidad))
    PPV=TP/(TP+FN)
    NPV=TN/(TN+FP)

    SENSIBILIDADES.append(sensibilidad)
    ESPECIFICIDADES.append(especificidad)
    PRECISIONES.append(precision)
    F1_SCORES.append(f1_score)
    PPVS.append(PPV)
    NPVS.append(NPV)
```

Imagen 3-1: Implementación de métricas

Como se puede comprobar en la imagen de arriba, hemos establecido las métricas por clase para posteriormente hacer una media de las métricas establecidas.

La principal métrica que usaremos para determinar la eficiencia de los modelos será la precisión multiclase normalizada. Hay que tener cuidado, pues para usar esta métrica dentro de la etapa de entrenamiento se puede confundir con ‘accuracy’, es decir, la precisión simple que es el total de predicciones acertadas dividido por el número total de intentos de predicción. La métrica utilizada en esa etapa ha sido, por tanto, la precisión categórica (categorical accuracy), que hace una media de las precisiones por clase.

3.2 Primer modelo a ensamblar: Distancia Euclídea respecto al vector prototipo

3.2.1 Generación del vector prototipo

Aprovechando las 11.099 imágenes de validación, hemos calculado los valores medios de las tres salidas de las redes para crear una media estadística para cada red. Esto lo hacemos para analizar el primer modelo de ensamblaje.

El motivo que nos ha llevado a usar el conjunto de validación reside en el vector de salida del conjunto de entrenamiento, que será muy próximo a uno en el elemento cuyo índice coincida la clase predicha y un valor muy cercano a cero para el resto de los elementos de dicho vector. Este valor de salida está muy alejado del valor que proporciona en el caso de que la entrada no pertenezca al conjunto de imágenes de entrenamiento.

Resultados

Esto es debido a que las CNNs han usado este conjunto para aprender de las clases, mientras que el conjunto de validación se usa en el entrenamiento pero las redes no aprenden de él, por lo que sigue siendo válido. El conjunto de prueba es el que proporciona las métricas finales, ya que no intervienen en ningún punto del entrenamiento, siendo, por tanto, el conjunto que más se asemeja a una puesta en escena real.

Hacemos un recorrido del conjunto de validación en lotes de 32 imágenes, para así obtener una matriz [32x8], cuyos elementos corresponden a la probabilidad de que la imagen asociada pertenezca a cada una de las 8 clases. A partir de aquí obtenemos ocho vectores que van acumulando las probabilidades de cada clase siguiendo la siguiente fórmula:

$$(V_{acumulativo})|_{clase} = (V_{acumulativo})|_{clase} + \sum (P_{imagen})|_{clase} \quad \text{Ec. 9}$$

La clase asociada a cada imagen es conocida previamente, ya que esa información está dentro de los conjuntos de datos. Aprovecharemos este bucle para poder contabilizar dentro de un vector N a qué clase pertenece cada imagen. Así podremos obtener el vector prototipo, dividiendo cada vector acumulativo por el elemento Nclase adecuado siguiendo la siguiente fórmula:

$$V_{prototipo}|_{clase} = V_{acumulativo}|_{clase} / N_{clase} \quad \text{Ec. 10}$$

De esta forma establecemos una matriz formada por todos los vectores prototipos por clase a la cual denominamos “matriz Atlas” de dimensiones [8x8x3] correspondiente a una matriz [8x8] de vectores prototipos por cada red entrenada.

3.2.2 Ensamblaje

Nuestro decisor para este modelo de ensamblaje será aquel cuya predicción para una imagen menos la matriz Atlas obtiene un valor lo más cercano a cero posible. Esto se consigue siguiendo la siguiente fórmula:

$$C_{pred} = \operatorname{argmin}((V_{pred,Res} - V_{atlas,Res}) + (V_{pred,Inc} - V_{atlas,Inc}) + (V_{pred,VGG} - V_{atlas,VGG})) \quad \text{Ec. 11}$$

3.2.3 Resultados

La matriz de confusión que hemos obtenido para este modelo es la siguiente:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	0	0	11	0	0	0	0	683
CBC	0	0	11	0	0	0	0	40
Derm	0	0	12	0	0	0	0	1309
Mel	0	0	2	0	0	0	0	1988
Nev	0	0	5	0	0	0	0	313
BK	0	0	8	0	0	0	0	166
SCC	0	0	29	0	0	0	0	9
Vasc	0	0	71	0	0	0	0	79

Tabla 4-2: Matriz de Confusión de ensamblaje con Atlas

La idea de este modelo de ensamblaje es estudiar la distancia euclídea con respecto a unos valores medios estadísticos que les permitiera modificar el decisor a un optimizador de distancia mínima.

Como se ve en la matriz de confusión, este modelo deshace la dependencia de la red hacia las imágenes entrenadas, provocando que la mayoría de las imágenes las clasifique como lesiones vasculares y dermatofibromas, por lo que podemos concluir con que no cumple la función para la cual fue diseñada.

3.3 Segundo modelo: Adición simple de las 3 redes

El segundo modelo que hemos diseñado y analizado se denominará modelo simple, en el cual sumaremos los tres vectores de salida de las redes neuronales que están escalados entre 0 a 1, de modo que el decisor se decanta por aquella clase cuyo argumento sea máximo en dicha suma.

$$V_{clase} = \operatorname{argmax}(\sum_i (V_{i,Clase})), i \in [Resnet, Inception, VGG] \quad \text{Ec. 11}$$

Dicho de otro modo, obtenemos las probabilidades acumulativas escaladas de 0 a 3 sumando tres vectores de probabilidad y decidimos como clase más probable aquella clase cuyo elemento del vector se hace máximo frente a las otras.

3.3.1 Resultados

La matriz de confusión del modelo simple es la siguiente:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	593	0	34	28	12	16	3	8
CBC	9	11	5	18	5	3	0	0
Derm	28	0	961	279	35	7	4	7
Mel	31	0	135	1791	31	0	1	1
Nev	20	0	38	64	190	1	0	5
BK	50	1	22	13	7	74	0	7
SCC	4	0	1	3	1	0	29	0
Vasc	20	1	18	6	26	9	0	70

Tabla 4-3: Matriz de confusión del modelo simple

Como se puede comprobar en la matriz de confusión, este modelo sí consigue hacer una predicción correcta que mejora cada red individual, las matrices de confusión de las redes por separado serán proporcionadas más adelante, pero este modelo provoca una mejora de un 11.18% en la precisión multiclase normalizada, con respecto a la mejor de las tres redes por individual.

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.8544	0.2156	0.7275	0.9	0.5975	0.4253	0.7631	0.4667
Especificidad	0.9599	0.9996	0.9259	0.8503	0.9735	0.9921	0.9983	0.9934
Precisión	0.7854	0.8461	0.7916	0.8134	0.6189	0.6727	0.7838	0.7143
F1 score	0.8185	0.3437	0.7582	0.8545	0.608	0.5211	0.7733	0.5645
PPV	0.8545	0.2157	0.7275	0.9	0.5	0.4253	0.4474	0.4267
NPV	0.9599	0.9996	0.9259	0.8503	0.9735	0.9921	0.9983	0.9939

Tabla 4-4: Métricas por clase del modelo simple

Conjunto de métricas de evaluación:

Sensibilidad Media: 0.6187785637739855

Especificidad Media: 0.9616943161225083

Precisión Media: 0.7532778887964939

F1 Score Media: 0.6552364842800422

PPV Media: 0.6187785637739855

NPV Media: 0.9616943161225083

Estos resultados demuestran que este modelo detecta un 21.56% de los Carcinomas Basocelulares que se le presentan. Esta lesión se considera maligna y no detectar un 78.44% de los casos correctamente para esta clase podría considerarse un peligro potencial para los pacientes.

Por otro lado, dentro de la misma clase, este modelo tiene un 99.96% de especificidad, lo que quiere decir que el 99.96% de los CBC que ha detectado este modelo están correctamente categorizados.

3.4 Tercer modelo de ensamblaje: Adición cuadrática de las 3 redes

Este modelo se ha diseñado siguiendo la siguiente idea: se disponen de tres predicciones, cada una generada por una red distinta, esta predicción está compuesta por un vector con ocho términos, uno para cada clase correspondiente, correspondiendo cada término con la probabilidad de que la entrada pertenezca a la clase asociada (tenemos un total de tres vectores con 8 términos cada uno, un total de 24 términos).

Cada término viene escalado entre 0 y 1, corresponde a la probabilidad de suceso en tanto por uno. En el modelo anterior se sumaron estos términos para generar un cuarto vector de 8 elementos generado por la suma de los otros 3 vectores.

Este modelo eleva al cuadrado los términos de los tres vectores previa a la generación del cuarto vector de decisión, de modo que tendríamos una situación muy parecida al modelo anterior: tendremos tres vectores escalados de 0 a 1, con la salvedad de que los valores previos cercanos a 0 se han hecho mucho más bajos que los valores cercanos a 1, provocando que en los casos en que haya términos cercanos entre sí puede hacer de desempate.

3.4.1 Resultados

La matriz de confusión del modelo cuadrático es la siguiente:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	591	0	36	31	9	15	3	9
CBC	9	10	5	18	6	3	0	0
Derm	28	1	960	279	35	8	4	6
Mel	32	0	135	1789	33	0	1	0
Nev	20	0	36	64	192	2	0	4
BK	49	1	23	16	7	70	0	8
SCC	5	0	1	3	1	0	28	0
Vasc	22	1	18	7	26	8	0	70

Tabla 4-5: Matriz de confusión del modelo cuadrático

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.8515	0.1961	0.7267	0.899	0.6038	0.4023	0.7368	0.4533
Especificidad	0.9518	0.9994	0.9256	0.8477	0.9735	0.9921	0.9983	0.9941
Precisión	0.7817	0.8692	0.7908	0.8106	0.6214	0.6604	0.7778	0.7158
F1 score	0.8152	0.3125	0.7574	0.8525	0.6124	0.5	0.7568	0.5551
PPV	0.8515	0.1961	0.7267	0.899	0.6038	0.4023	0.7368	0.4533
NPV	0.9592	0.9994	0.9256	0.8478	0.9735	0.9921	0.9983	0.9941

Tabla 4-6: Tabla de métricas del modelo cuadrático

Conjunto de métricas de evaluación:

Sensibilidad Media: 0.6087035593624434

Especificidad Media: 0.9612468677178819

Precisión Media: 0.7409571952584242

F1 Score Media: 0.6452351940899975

PPV Media: 0.6087035593624434

NPV Media: 0.9612468677178819

Si comparamos esta matriz de confusión con la del modelo anterior, cumple unas especificaciones muy parecidas al modelo simple, siendo este modelo ligeramente peor en este último modelo. No obstante, durante las múltiples pruebas de la realización de este proyecto se han obtenido resultados en los que este modelo superó al modelo anterior en la misma medida que en esta última prueba el modelo simple ha superado al modelo cuadrático.

3.5 Modelo de ensamblaje directo

Este proyecto usa la librería de Python de Tensorflow [19], una librería especializada en aprendizaje automático, concretamente hemos usado Keras [18], una API que trabaja dentro de Tensorflow para el diseño de redes de aprendizaje profundo.

Este modelo se basará en la implementación del ensamblaje como un gran modelo que contendrá los otros tres modelos establecidos.

La idea es usar los tres modelos entrenados previamente como base del modelo completo terminados en una capa densa de ocho neuronas que le permita analizar la dependencia de cada red con el decisor final.

Teóricamente, durante el entrenamiento de la red, se comprobará si alguna red acierta más que las otras para darle más peso y analizará los otros modelos anteriormente propuestos. Por desgracia, como veremos a continuación, no se puede estar seguro de lo que aprenden las redes neuronales profundas, y por tanto, no es seguro que este modelo, que parece más completo que el resto, tenga mejores prestaciones.

3.5.1 Resultados

La matriz de confusión derivada del modelo descrito es la siguiente:

MC	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
AK	603	0	31	40	10	8	0	2
CBC	12	4	6	23	2	2	0	2
Derm	28	0	945	310	29	6	0	3
Mel	27	0	130	1817	16	0	0	0
Nev	24	0	39	83	168	1	0	3
BK	60	0	25	17	6	61	0	5
SCC	6	0	6	8	1	0	17	0
Vasc	25	0	19	13	22	7	0	64

Tabla 4-7: Matriz de confusión del modelo de ensamblaje directo

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.8689	0.0784	0.7154	0.9131	0.5283	0.3506	0.4474	0.4267
Especificidad	0.955	1.0	0.925	0.8201	0.9805	0.9947	1.0	0.9967
Precisión	0.7682	1.0	0.7868	0.7862	0.6614	0.7176	1.0	0.8101
F1 score	0.8154	0.1454	0.7494	0.8449	0.5874	0.471	0.6182	0.559
PPV	0.8689	0.0784	0.7154	0.913	0.5283	0.3506	0.4474	0.4267
NPV	0.955	1.0	0.925	0.8201	0.9805	0.9947	1.0	0.9967

Tabla 4-8: Tabla de métricas por clase del modelo de ensamblaje directo

Conjunto de métricas de evaluación:

Sensibilidad Media: 0.5410814516413369

Especificidad Media: 0.9590142323846012

Precisión Media: 0.8163034812077441

F1 Score Media: 0.5988480285869389

PPV Media: 0.5410814516413369

NPV Media: 0.9590142323846012

Como se puede comprobar en la matriz de confusión, este modelo gana precisión en algunas clases, perdiendo en otras, el mayor problema viene de la clase más restrictiva (CBC), donde cada imagen detectada como esta lesión está correctamente categorizada, pero detecta menos que los otros modelos.

A pesar de lo dicho en el párrafo anterior, este modelo tiene una precisión media de 81.63%, siendo el modelo con mejores prestaciones de todos los estudiados. Esto se debe a que en la etapa de entrenamiento puede ajustar los pesos de las redes internas para que la más efectiva sea la que tenga la voz cantante e incluso puede eliminar la dependencia residual entre las 3 redes.

3.6 Modelos Independientes

Cada una de las tres redes usadas en los modelos anteriormente explicados son, a su vez, modelos predictivos por sí mismos y con sus propias especificaciones. La idea de los modelos de ensamblaje es conseguir una mejora en la métrica principal (precisión multiclase normalizada) con respecto a cada uno de los modelos por separado. Esto se ha conseguido en los modelos de ensamblaje por adición simple y cuadrática, donde consiguen una mejora con respecto a la mejor red independiente.

El modelo de ensamblaje directo tiene unas especificaciones que están en torno a los modelos independientes, provocando que no haya ninguna mejora con respecto a estos últimos, pero funciona como decisor de imágenes médicas.

3.6.1 Modelo VGG:

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.8184	0.1765	0.6866	0.8553	0.6038	0.3218	0.7632	0.4333
Especificidad	0.9589	0.997	0.911	0.8412	0.9545	0.9906	0.9972	0.9932
Precisión	0.7738	0.3913	0.749	0.7961	0.4885	0.5657	0.6905	0.6771
F1 score	0.7955	0.2432	0.7164	0.8246	0.5401	0.4103	0.725	0.5285
PPV	0.8184	0.1764	0.6866	0.8553	0.6038	0.3218	0.7632	0.4333
NPV	0.9589	0.997	0.911	0.8412	0.9545	0.9906	0.9972	0.9932

Tabla 4-9: Tabla de métricas del modelo VGG

Conjunto de métricas de evaluación:

Sensibilidad Media: 0.5823619659272762

Especificidad Media: 0.9554624122191203

Precisión Media: 0.6414938633946301

F1 Score Media: 0.5979499545579829

PPV Media: 0.5823619659272762

NPV Media: 0.9554624122191203

3.6.2 Modelo RESNETv2:

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.7867	0.1569	0.6662	0.8688	0.4623	0.3908	0.5789	0.4733
Especificidad	0.9599	0.9979	0.9063	0.8099	0.9735	0.9879	0.9981	0.9847
Precisión	0.7712	0.4444	0.7333	0.7681	0.5568	0.5528	0.7097	0.5035
F1 score	0.7789	0.2319	0.6981	0.8154	0.5052	0.4579	0.6377	0.488
PPV	0.7867	0.1569	0.6662	0.8688	0.4623	0.3908	0.5789	0.4733
NPV	0.9599	0.9979	0.9063	0.8099	0.9735	0.9879	0.9981	0.9847

Tabla 4-10: Tabla de métricas del modelo Resnet50v2

Conjunto de métricas de evaluación:
Sensibilidad Media: 0.5479952413673297
Especificidad Media: 0.9522836485080883
Precisión Media: 0.6299943140848923
F1 Score Media: 0.57662518999802
PPV Media: 0.5479952413673297
NPV Media: 0.9522836485080883

3.6.3 Modelo INCEPTIONv3:

Métricas/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
Sensibilidad	0.7421	0.1961	0.7116	0.8362	0.5535	0.3563	0.5789	0.3467
Especificidad	0.9552	0.9968	0.8817	0.8503	0.9615	0.9877	0.9981	0.9893
Precisión	0.7399	0.4	0.6994	0.8019	0.5087	0.5254	0.7097	0.5149
F1 score	0.741	0.2632	0.7054	0.8187	0.5301	0.4247	0.6377	0.4143
PPV	0.7421	0.1961	0.7116	0.8362	0.5535	0.3563	0.5789	0.3467
NPV	0.9552	0.9968	0.8817	0.8503	0.9615	0.9877	0.9981	0.9893

Tabla 4-11: Tabla de métricas del modelo InceptionV3

Conjunto de métricas de evaluación:
Sensibilidad Media: 0.5401639240325614
Especificidad Media: 0.9525862438014409
Precisión Media: 0.6124872693789869
F1 Score Media: 0.566887997697852
PPV Media: 0.5401639240325614
NPV Media: 0.9525862438014409

3.7 Cálculo de la métrica

A partir de [ec. 2] podemos calcular la precisión multiclase normalizada. Para ello es necesario saber el número de muestras de cada clase (N_c) así como el número de aciertos por clase (TPc):

$$BMA = [\sum (TP_c / N_c)] / 8$$

Este resultado nos permite saber en qué posición hubiese quedado este proyecto en el concurso ISIC 2019 [1][3][7][10].

Nombre	BMA (Métrica)
Proyecto propio (adición simple)	0.8163
1) DAISYLab	0.636
2) DysionAI	0.607
3) AImageLab & PRHLT	0.593
4) DermaCode	0.578
5) Nurithm Labs	0.569
6) Torus Actions	0.563
7) BITDeeper	0.558
8) SYSU-MIA-Group	0.557
9) MelanoNorm_IITRopar	0.546
10) MH_team	0.544

Tabla 4-12: Principal métrica del concurso de los 10 mejores participantes, incluida la del proyecto propio

Métrica	sensibilidad	especificidad	precisión	F1	PPV	NPV
Valor	54.1081%	95.9014%	81.6303%	59.8848%	54.1081%	95.9014%

Tabla 4-13: Resultado de las métricas secundarias pedidas (en media) para el modelo directo

3.7.1 Análisis de la sensibilidad

Entre las distintas métricas secundarias cabe destacar la sensibilidad como métrica secundaria que limita el sistema. Esto se puede ver en la siguiente tabla:

Modelo/Clase	AK	CBC	Derm	Mel	Nev	BK	SCC	Vasc
VGG	0.8184	0.1765	0.6866	0.8553	0.6038	0.3218	0.7632	0.4333
RESNET	0.7867	0.1569	0.6662	0.8688	0.4623	0.3908	0.5789	0.4733
INCEPTION	0.7421	0.1961	0.7116	0.8362	0.5535	0.3563	0.5789	0.3467
Votacion simple	0.8544	0.2156	0.7275	0.9	0.5975	0.4253	0.7631	0.4667
Votación cuadrática	0.8515	0.1961	0.7267	0.899	0.6038	0.4023	0.7368	0.4533
Modelo Directo	0.8689	0.0784	0.7154	0.9131	0.5283	0.3506	0.4474	0.4267

Tabla 4-14: Resultados de la métrica secundaria: Sensibilidad

Como se puede observar, el modelo que ha obtenido mejor resultado en la precisión multiclase normalizada, obtiene los peores resultados en esta métrica, esto es debido al desbalanceo en la cantidad de muestras por clase que se han obtenido. Sin embargo, tanto el modelo simple como el modelo cuadrático han obtenido sensibilidades a la par o incluso mejores que los modelos independientes.

4 CONCLUSIONES

Como se puede ver comparando los resultados obtenidos, la mayoría de los modelos ensamblados han conseguido unas especificaciones decentes en comparación con los del concurso ISIC 2019[1][3][7][10].

Sin embargo, no se ha conseguido una buena clasificación para la clase de Carcinoma Basocelular, una de las lesiones malignas. Esto se debe al bajo número de imágenes que pertenecen a esta clase que se han obtenido, lo que ha provocado que las redes neuronales hayan aprendido más de aquellas clases que han tenido muestras mayoritarias, creando un desbalance con respecto a las clases minoritarias.

La red que más ha acertado la clase CBC ha sido el modelo simple, que ha conseguido acertar 11 de las 51 imágenes, por lo que estos modelos siguen siendo precarios para su uso práctico, pues fallar prácticamente en 4 de cada 5 predicciones para esta clase provocaría muchos riesgos potenciales para los pacientes.

Esta clase ha obtenido una accuracy de un 44% en la mejor de las redes independientes con un PPV de un 15%, esto quiere decir que ha encontrado un 15% de las imágenes que pertenecen a dicha clase y ha acertado un 44% de ellas. En los modelos de ensamblaje se han obtenido mejores resultados para dicha clase, obteniendo 85% de accuracy para un 21.57% de valor predictivo positivo en el modelo de adición simple, es decir, ha encontrado un 6.57% más de muestras pertenecientes a dicha clase.

El modelo cuadrático ha obtenido un 86.2% de precisión para un PPV de 19.61%. El modelo directo obtiene una mejor accuracy pero detecta menos veces la enfermedad. Esto implica que el modelo directo no cumple bien su función a la hora de resolver el desbalance de clases, mientras que los demás modelos ensamblados funcionales han conseguido una cierta mejora con respecto al mejor de los modelos independientes.

Siendo esta clase una enfermedad maligna, podría poner en riesgo la vida de un paciente en el caso de que se le clasifique incorrectamente como una enfermedad benigna. Por lo que sigue siendo un proyecto que no está preparado para su puesta en marcha en la vida real.

Para otras clases se ha obtenido mejores resultados pero siguen sin ser suficiente en algunos casos, como son los casos de nevus, lesiones vasculares y queratosis benigna, donde se han obtenido una precisión de un 55%, 56% y 67% para un PPV de 46%, 32% y 43% respectivamente. Estos valores pertenecen al mejor de los casos de las redes independientes. Para los modelos de ensamblaje, los modelos aditivos obtienen mejoras en todas estas clases, siendo la menor de estas mejoras en la adición simple donde se ha obtenido un 71% de precisión para un 45% de PPV para las lesiones vasculares, es decir, ha detectado menos veces dicha categoría erróneamente, habiendo acertado más veces.

Para el caso del modelo de ensamblaje directo ha obtenido mejores valores de precisión que los modelos independientes, sin embargo no ha mejorado en otras métricas, como el PPV, esto quiere decir que para la mayoría de clases ha fallado menos en clasificarlas, pero para aquellas que ha tenido en duda las ha agrupado en las clases melanoma y dermatofibroma, esto quiere decir que tiene un peor balance multiclase que los otros modelos de ensamblaje.

Como conclusión, el modelo simple ha sido el que mejores resultados ha dado como modelo ensamblado seguido muy de cerca por el modelo cuadrático.

REFERENCIAS

- [1] N. Gessert, M. Nielsen, M. Shaikh, R Werner and A. Schlaefer, *Skin Lesion Classification Using Loss Balancing and Ensembles of Multi-Resolution EfficientNets*, ISIC Challenge, 2019.
- [2] Tschandl P., Rosendahl C. & Kittler H. *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. 2018
- [3] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, Allan Halpern, *Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI)*, International Skin Imaging Collaboration (ISIC), 2017
- [4] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, Josep Malvehy: *BCN20000: Dermoscopic Lesions in the Wild*, 2019
- [5] L. G. Kabari and F. S. Bakpo, *Diagnosing skin diseases using an artificial neural network* 2nd International Conference on Adaptive Science & Technology (ICAST), 2009
- [6] N. Gouda and J. Amudha, *Skin Cancer Classification using ResNet*, IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 2020
- [7] Josep Malvehy, Allan Halpern, Noel C. F. Codella, *ISIC 2019 Challenge*, <https://Challenge.isic-archive.com/landing/2019/>
- [8] D. -M. Filimon and A. Albu, *Skin diseases diagnosis using artificial neural networks*, 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), 2014
- [9] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, *Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM*, 2021
- [10] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, Allan Halpern, *Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging*, International Skin Imaging Collaboration (ISIC), 2017
- [11] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, Josep Malvehy, *BCN20000: Dermoscopic Lesions in the Wild*, 2019
- [12] Mayo Clinic, *Melanoma - Síntomas y causas*, 2024, <https://www.mayoclinic.org/es/diseases-conditions/melanoma/symptoms-causes/syc-20374884>
- [13] Mayo Clinic, *Queratosis Actínica - Síntomas y Causas*, 2022,

<https://www.mayoclinic.org/es/diseases-conditions/actinic-keratosis/symptoms-causes/syc-20354969>

- [14] Paula Ludman, *Skin Cancer types: Squamous Cell Carcinoma Overview*, 2023, <https://www.aad.org/public/diseases/skin-cancer/types/common/scc>
- [15] Victor Alegre de Miguel, *Angiomas*, <https://www.uv.es/=vicaleg/PTindex/ANGIOMAS.html>
- [16] Skin Cancer Foundation, *El Carcinoma Basocelular/Carcinoma de células basales*, 2023, <https://www.skincancer.org/international/el-carcinoma-basocelular-carcinoma-de-celulas-basales/>
- [17] Denise M. Aaron, *Lesiones vasculares cutáneas*, 2023, <https://www.msdmanuals.com/es-es/professional/trastornos-dermatológicos/tumores-cutáneos-benignos,-proliferaciones-y-lesiones-vasculares/lesiones-vasculares-cutáneas>
- [18] Chollet François and Others, *Keras*, 2015, <https://keras.io>
- [19] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng.
TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [20] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, Ren'e Werner, and Alexander Schlaefer, *Skin Lesion Classification Using Loss Balancing and Ensembles of Multi-Resolution EfficientNets*
- [21] Steven Zhou, Yixin Zhuang, Rusong Meng, *Multi-Category Skin Lesion Diagnosis Using Dermoscopy, Images and Deep CNN Ensembles*
- [22] Robi Polikar, *Ensemble Based Systems in Decision Making*, *IEEE Circuits and System Magazine*, 2006
- [23] Álvaro Artola Moreno, José Antonio Pérez Carrasco, *Clasificación de imágenes usando redes neuronales convolucionales en Python*, Trabajo de fin de grado de la universidad de sevilla, 2024
- [24] Globaldee, usuario de TheContentFarm.Net, *Ensemble Techniques for Handling Class*, <https://thecontentfarm.net/ensemble-techniques-for-handling-class-imbalance/>