

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
INGENIERÍA DE LA SALUD

**ANÁLISIS DE IMÁGENES DIGITALES MEDIANTE TÉCNICAS
DE APRENDIZAJE PROFUNDO PARA LA DETECCIÓN DE
MELANOMAS**

**DIGITAL IMAGES ANALYSIS BY DEEP LEARNING
TECHNIQUES FOR MELANOMA DIAGNOSIS**

Realizado por
Carmen Alcaraz Garófano
Tutorizado por
Enrique Domínguez Merino
Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE 2018

Fecha defensa:

El Secretario del Tribunal

Resumen:

Entre los diversos tumores que podemos sufrir en nuestra piel, el melanoma cutáneo no es el más frecuente, pero sí el más agresivo. Aunque últimamente la supervivencia de los pacientes con melanoma cutáneo está mejorando, esta no se debe a que se hayan descubierto mejoras en el tratamiento, sino al aumento de diagnósticos precoces. El melanoma cutáneo se convierte rápidamente en metástasis, ya que crece tanto horizontalmente como verticalmente, pero en sus etapas iniciales la probabilidad de morir a causa del tumor es prácticamente nula.

Por lo tanto, la detección precoz del melanoma supone un elemento clave en la supervivencia del paciente. El objetivo de este trabajo ha sido crear un método informático capaz de analizar y clasificar melanomas a partir de una imagen clínica digital. El método propuesto está basado en técnicas de aprendizaje profundo, también conocido por su nombre en inglés como *deep learning*. Conseguir un algoritmo que realice esta clasificación de forma fidedigna podría ser la antesala de una aplicación de diagnóstico a distancia, propia de la telemedicina.

En primer lugar, se recopilaron imágenes de melanomas y nevos, las cuales fueron tratadas y preprocesadas antes de ser utilizadas por las redes neuronales convolucionales creadas. En cuanto a la arquitectura neuronal utilizada, se crearon cuatro versiones distintas. Tres de ellas fueron redes creadas de cero, mientras que la última se creó a partir de AlexNet.

Palabras claves:

aprendizaje profundo, red neuronal convolucional, melanoma, MatLab.

Abstract:

Among the tumors that we can suffer in our skin, cutaneous melanoma is not the most frequent, but the most aggressive. Although the survival rate of patients suffering from cutaneous melanoma has improved recently, this is not due to improvements in treatment, but to the increase in early diagnosis. Cutaneous melanoma rapidly leads to metastasis, since it grows both horizontally and vertically, but in its initial stages the probability of dying from the tumor is practically nil.

Therefore, the early detection of melanoma is a key element in patient survival. The objective of this work has been to create a computer method capable of analyzing and classifying melanomas from a digital clinical image. The proposed method is based on deep learning techniques. Achieving an algorithm that performs this classification in a reliable manner could be the prelude to a remote diagnosis application.

First, images of melanomas and nevi were collected, which were treated and preprocessed before being used to test the created convolutional neural networks. Regarding the used neural architectures, four different versions were created. Three of them were networks created from scratch, while the last one was created based on AlexNet.

Keywords:

Deep learning, convolutional neural network, melanoma, MatLab.

Índice de contenido

Capítulo 1: Introducción	9
1.1. Contextualización del problema	9
1.2. Objetivo del TFG	9
1.3. Estructura del TFG	10
Capítulo 2: Melanomas	11
2.1. La piel: estructura y componentes	11
2.2. Cáncer de piel	12
2.3. Melanoma y tipos de melanoma	13
2.4. Diagnóstico clínico	15
Capítulo 3: Estado del arte	17
3.1. Aprendizaje profundo	17
3.2. Redes neuronales	18
3.3. Red neuronal convolucional	19
3.4. Neural Network Toolbox en MatLab	20
Capítulo 4: Desarrollo de los modelos neuronales	23
4.1. Recogida de datos	23
4.2. Tratamiento de la imagen	24
4.3. Creación y configuración de la red	26
Capítulo 5: Entrenamiento y resultados	33
5.1. Conjunto de datos	33
5.2. Criterios de evaluación	34

5.3. Resultados experimentales	35
Capítulo 6: Conclusiones y futuro	41
6.1. Conclusión del TFG	41
6.2. Futuras posibles ampliaciones del trabajo	42
Referencias bibliográficas	45

Capítulo 1: Introducción

1.1. Contextualización del problema

El melanoma cutáneo no es el más frecuente de los tumores que puede aparecer en nuestra piel, pero sí es el más agresivo que podemos sufrir. Este tipo de tumor produce la muerte de casi dos terceras partes de las personas que lo padecen [1].

Publicaba el periódico La Vanguardia en junio de este mismo año que anualmente se diagnostican alrededor de 4.000 casos de melanoma en nuestro país, según los datos que ofrecía la Asociación Española de Dermatología y Venerología. Paloma Borregón, dermatóloga, advierte que un 65% de la población reconoce haber sufrido quemaduras con ampollas alguna vez en su vida, práctica que acarrea consecuencias sobre la salud de nuestra piel [2].

Según la Asociación Española Contra el Cáncer (aecc), actualmente se diagnostican unos 160.000 casos de melanomas al año en todo el mundo. En concreto en nuestro país, la incidencia puede considerarse alta, observándose un ascenso importante, especialmente desde los años 90. Además, añade “la supervivencia está mejorando en los últimos años, pero se debe más al aumento de los diagnósticos precoces que a auténticas mejoras en el tratamiento, que apenas las hay” [3].

Cuando se diagnostica en su etapa inicial, la probabilidad de sobrevivir del paciente es prácticamente del 100%. En cambio, cuando se encuentra en una fase avanzada, el pronóstico tiende a ser muy negativo [4]. La detección precoz del melanoma supone un elemento clave en la supervivencia del paciente.

1.2. Objetivo del TFG

El objetivo de este trabajo ha sido crear un método informático capaz de analizar y clasificar melanomas a partir de una imagen clínica digital. El método propuesto está basado en técnicas de aprendizaje profundo, también conocido por su nombre en inglés como *deep learning*.

Crear un método informático capaz de clasificar melanomas y nevos podría ser de gran ayuda para especialistas y pacientes. El especialista podría obtener en cuestión de minutos un segundo diagnóstico en la palma de su mano a través de un dispositivo electrónico. Conseguir un algoritmo que realice esta clasificación de forma fidedigna podría ser la antesala de una aplicación de diagnóstico a distancia, propia de la telemedicina. Los pacientes acudirían al dermatólogo, tras un diagnóstico en casa, antes de que quizás fuera demasiado tarde.

1.3. Estructura del TFG

El desarrollo del trabajo se basa en una metodología clásica en cascada, adecuada para proyectos de investigación. Consta de las siguientes fases de trabajo: recopilación de imágenes, estudio de los melanomas, estudio del arte, implementación del algoritmo, evaluación del método y, por último, la realización de esta memoria.

La memoria se estructura en seis capítulos. El primero es este, el capítulo de introducción, en el que se contextualiza el problema a tratar, el objetivo del trabajo y su estructura. El segundo capítulo es en el que se conocerán aspectos básicos de la piel tales como su estructura y componentes, el concepto de melanoma, los tipos que existen y en qué se basa el diagnóstico clínico de los mismos. En tercer lugar, se encuentra un capítulo dedicado al estado del arte, en este caso dedicado al aprendizaje profundo, las redes neuronales y la plataforma que se ha utilizado para realizar el algoritmo, MatLab.

En el capítulo cuatro, se desarrollan las fases de la creación del método, cuyo entrenamiento y resultados aparecen en el capítulo cinco. Por último, encontramos el sexto capítulo, en el que se encuentra la conclusión del TFG y se exponen las futuras mejoras y ampliaciones que podrían realizarse en relación a este trabajo.

Capítulo 2: Melanomas

Antes de comenzar a hablar sobre los melanomas, para comprender qué son y cómo se desarrollan, deberíamos conocer conceptos básicos sobre el órgano más grande del cuerpo humano: la piel.

2.1. La piel: estructura y componentes

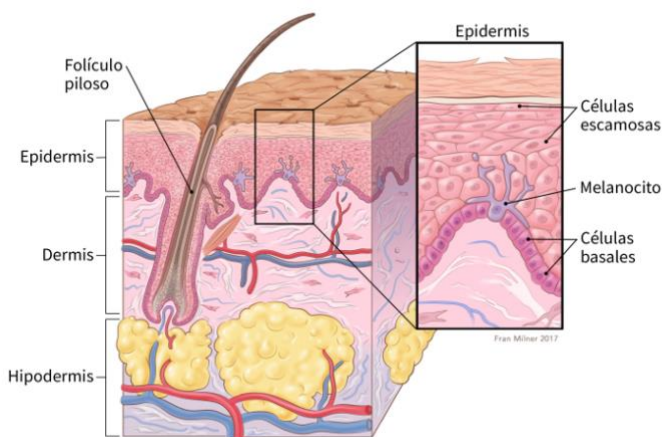


Figura 1: Las capas de la piel.

La piel es el mayor órgano del cuerpo humano. En un adulto medio ocupa una superficie de alrededor de 2 m² y pesa entre 4 y 5 kilogramos. Estructuralmente hablando consta de dos capas principales. La superficial, más fina y compuesta por tejido epitelial, es la epidermis.

La epidermis contiene cuatro tipos principales de células: queratinocito, melanocito, célula de Langerhans y célula de Merkel. Aproximadamente el 90% de las células que forman la epidermis son queratinocitos. De las restantes, alrededor del 8% son melanocitos. Estos derivan del ectodermo embrionario y producen la melanina, un pigmento de color amarillo-rojizo o pardo-negruzco que otorga color a la piel y absorbe los rayos ultravioleta nocivos [5].

Por lo general, cada melanocito está rodeado de entre 5 y 8 queratinocitos, formando unidades “epidermo-melanocíticas”. Estos queratinocitos son los encargados de producir la queratina, la proteína estructural principal de la piel, además de diversos factores de crecimiento que permiten mantener la homeostasis normal de este tejido. A su vez, estos queratinocitos, se encargan de regular la proliferación de los melanocitos [6].

Por otro lado, la región más profunda de la piel es la dermis. Está formada principalmente por tejido conectivo y en ella se encuentran los vasos sanguíneos, nervios, glándulas y folículos pilosos [7]. La combinación de fibras colágenas y elásticas otorga a la piel resistencia, extensibilidad y elasticidad [8].

Debajo de la piel, pero sin formar parte de esta, se encuentra el tejido subcutáneo, también llamado hipodermis. Sirve como depósito de reserva de grasas y contiene numerosos vasos sanguíneos que irrigan la piel [9]. Esta última característica es clave para entender la rápida metástasis producida por los melanomas, tal y como se explicará posteriormente. En la Figura 1 [10] pueden apreciarse las tres capas mencionadas.

2.2. Cáncer de piel

El melanoma no es la única expresión del cáncer de piel. Hay tres formas comunes de este tipo de tumores. Dos de estos tipos se conocen como “cáncer de piel no melanoma”. Estos son el carcinoma de células basales y de células escamosas, los cuales pueden apreciarse respectivamente en las Figuras 2 [11] y 3 [12].



Figura 2: Carcinoma de células basales

El carcinoma de células basales, se originan en las células del estrato basal de la epidermis y raramente dan metástasis. El carcinoma de células escamosas, se originan a partir de células espinosas de la epidermis, con tendencia variable a originar metástasis.



Figura 3: Carcinoma de células escamosas

Pero el tipo más preocupante es el tercero, en el que se centra este trabajo: los melanomas malignos. Debido al adelgazamiento de la capa de ozono, la cual absorbe una parte de los rayos ultravioletas que atraviesan la atmósfera, y la costumbre de pasar tiempo al sol para conseguir un llamativo bronceado, se observó un incremento en el número de tumores de este tipo. Los melanomas malignos dan metástasis rápidamente. Pueden producir la muerte del paciente en

un plazo de meses tras el diagnóstico. La clave de la eficacia del tratamiento es la detección temprana [13].

Por ello, el método que se investiga en este trabajo pretende clasificar este último tipo de tumores, los melanomas malignos. A continuación, se definen en profundidad y se exponen los tipos de melanoma que existen.

2.3. Melanoma y tipos de melanoma

Para hablar de melanomas, debemos centrarnos en los melanocitos, unas células dendríticas situadas en la zona inferior de la epidermis. Los melanocitos producen melanosomas, que son orgánulos especializados en la síntesis y almacenamiento de la melanina [14].

Un melanoma consiste en el crecimiento incontrolado de melanocitos formando una sola mancha. Por lo general estos melanomas no solo aumentan su superficie horizontalmente, también se hacen más profundos. Estos crecen tanto horizontal como verticalmente, atravesando por completo la epidermis y llegando así a penetrar la dermis. Una vez que el melanoma atraviesa la dermis, las células tumorales malignas tiene acceso a la sangre y los vasos linfáticos, dando vía libre a la invasión de cualquier parte del cuerpo, y produciendo así la temida metástasis.

Por ello algunos expertos dicen que el melanoma no nace siendo maligno, ya que en sus primeras etapas no suponen un peligro real para el paciente. Esta es la razón por la que la detección precoz de estos tumores cobra un papel tan importante. En la Figura 4 [15] se representa un melanoma en estadio 0. En total los expertos definen 5 estadios, empezando por el 0 y acabando por el estadio 4. Es en el estadio 3 cuando las células cancerígenas comienzan a realizar la metástasis. En sus primeras etapas es inofensivo, una vez atravesada la dermis, teniendo acceso a la

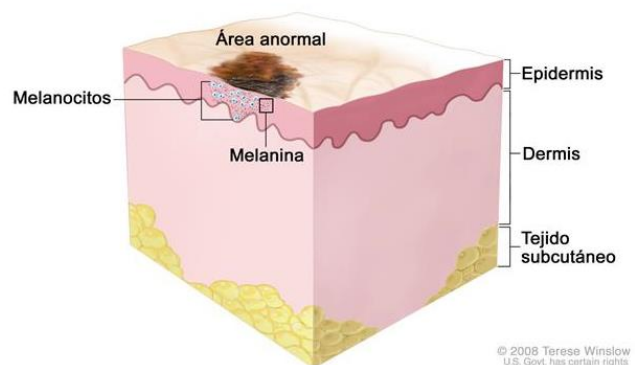


Figura 4: Melanoma in situ en estadio 0.

sangre y los vasos linfáticos, pueden desarrollar tumores en cualquier parte del cuerpo prácticamente.

Dentro de los melanomas, podemos encontrar muchos tipos: de extensión superficial, nodular, lentigo maligno, lentiginoso acral, de las mucosas, ocular o amelanótico [16].

En concreto, el melanoma de extensión superficial es el más frecuente en la población caucásica, puede apreciarse un ejemplo de él en la figura 5 [17]. Pueden aparecer en cualquier parte del cuerpo, pero por lo general aparece en la espalda en el caso de los hombres y en las piernas en el caso de las mujeres.



Figura 5: Melanoma de extensión superficial.



Figura 6: Melanoma nodular

El segundo tipo más frecuente, representado en la figura 6 [18], es el melanoma nodular. Estos son más peligrosos ya que crecen verticalmente desde el inicio, así que incluso los más pequeños aparentemente tienen capacidad metastásica.

Por otro lado, los léntigos malignos son comunes en pacientes mayores que han sufrido una gran exposición solar. Los melanomas lentiginosos acrales presentan tres subtipos: de palmas y plantas, subungueal y mucoso. Estos pueden sufrirlos también los pacientes de raza negra y asiática, menos sensibles a los tipos anteriores.

Las formas clínicas atípicas como los amelanóticos o no pigmentados y los verrucosos presentan problemas a la hora de realizar un diagnóstico precoz, ya

que es fácil confundirlos con procesos benignos de la piel [19]. A continuación, se exponen las pautas que se siguen durante el diagnóstico clínico.

2.4. Diagnóstico clínico

Cualquier lesión pigmentaria en la piel que experimente cambios en color, forma, contorno o tamaño, está reclamando atención y diagnóstico médico. Un lunar o nevo es un tumor benigno de la piel que se origina a partir de melanocitos. Los lunares no suelen ser perjudiciales, pero algunos tipos pueden ser signo de alarma.

Un tipo de lunar que puede producir confusiones es el conocido como “nevo spitz”. Es común en niños y adolescentes, y aunque son tumores benignos sin riesgo de propagarse, a menudo se extirpan como medida de seguridad al no ser posible distinguirlos de un melanoma maligno [20].

Vamos a centrarnos en la detección precoz del melanoma, por lo que no vamos a considerar signos como el sangrado y la ulceración como referencia para el diagnóstico, pues son propias de una enfermedad local avanzada. La regla que se va a aplicar a la hora de realizar un diagnóstico es la conocida por los expertos como “ABCDE”. Cada letra se corresponde con una característica propia de los melanomas.

A de asimetría, los melanomas tienden a ser irregulares, por lo tanto, se estudia cualquier distinción entre mitades en cualquiera de sus ejes. B de bordes, pues que tengan los bordes mal delimitados e irregulares también es signo de mal pronóstico. C de color, que tiende a ser diverso y no homogéneo en los melanomas. D de diámetro, ya que si la lesión es mayor de 6 milímetros podría ser maligno. Por último, E de evolución, es importante vigilar la lesión y observar si cambia su aspecto en el tiempo, sobre todo en periodos cortos [21].

Nuestro método, para el que usaremos imágenes digitales de melanomas y nevos, podrá tomar como referencia 3 de estas características: asimetría,

irregularidad de los bordes y no homogeneidad del color. Las imágenes se obtendrán de internet, por lo que no tendremos datos sobre su diámetro o su evolución.

Capítulo 3: Estado del arte

Para realizar el método se han utilizado técnicas de aprendizaje profundo. A continuación, se exponen conceptos básicos sobre este tipo de técnicas, redes neuronales y la plataforma que se ha usado para realizar el algoritmo: *Neural Network Toolbox* en MatLab.

3.1. Aprendizaje profundo

El aprendizaje profundo o *deep learning* es un tipo de aprendizaje automático que enseña al ordenador a realizar algo natural para el ser humano: aprender a partir de la experiencia. Consigue llevar a cabo tareas de clasificación a partir de imágenes, textos o sonidos directamente, es decir, a partir de los datos sin depender de ecuaciones o modelos predeterminados [22].

Es por eso que las técnicas de aprendizaje profundo suelen necesitar cientos de miles o millones de imágenes para obtener los mejores resultados posibles, lo que la convierte en una técnica computacionalmente intensa [23].

El aprendizaje profundo es especialmente adecuado para aplicaciones de identificación como pueden ser el reconocimiento facial, la traducción de texto o el reconocimiento de voz. Este tipo de aprendizaje se denomina “profundo” por el número de capas que tiene la red neuronal. Mientras una red neuronal tradicional solo consta de 2 o 3 capas, una red de aprendizaje profundo puede tener cientos de ellas [24].

Lo que hace que el aprendizaje profundo sea una tecnología de vanguardia es su precisión. Dicho grado de exactitud es posible principalmente gracias a tres cosas. Una de ellas es el fácil acceso a conjuntos masivos de datos etiquetados, como *ImageNet* o *PASCAL VoC*. Otra, que el poder de la computación se encuentra en aumento, ya que existen GPU de alto rendimiento las cuales aceleran el entrenamiento de montones de datos, reduciendo el tiempo de entrenamiento de semanas a tan solo horas. Por último, el hecho de que se tenga acceso a modelos preentrenados construidos por expertos, como puede ser *AlexNet*, facilita la

creación de nuevas redes neuronales a partir de estos modelos. *AlexNet* ha sido entrenada con más de un millón de imágenes en alta resolución para reconocer mil objetos diferentes [25].

3.2. Redes neuronales

Las redes neuronales están compuestas de elementos simples operando en paralelo, inspirándose en el sistema biológico nervioso. Se usan para llevar a cabo tareas complejas en muchas materias, como el reconocimiento de patrones, identificación, clasificación, lenguaje, visión y control de sistemas.

Una red neuronal de aprendizaje profundo consta de múltiples capas de procesamiento no lineales: una capa de entrada, varias capas ocultas y una capa de salida. Las capas están interconectadas a través de neuronas, y cada capa oculta usa como entrada la salida de la capa anterior. En la Figura 7 se esquematiza la arquitectura de la red [26].

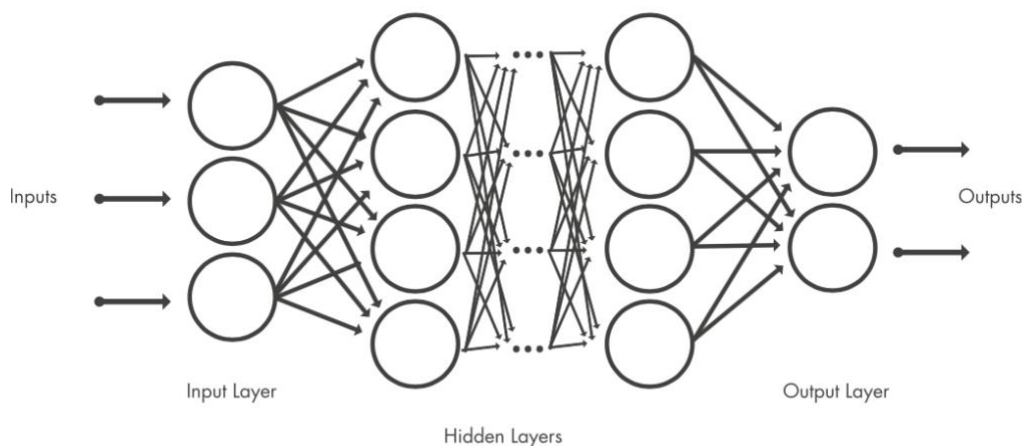


Figura 7: Representación gráfica esquematizada de una red neuronal.

Para entrenar una red neuronal lo principal es seleccionar un conjunto de imágenes suficientemente grande como para que la red aprenda. Si por ejemplo quisiéramos hacer la clasificación de cuatro objetos diferentes, deberíamos conseguir imágenes en las que aparezca una de las cuatro categorías diferentes de objetos que hayamos seleccionado.

El objetivo es que la red sea capaz de reconocer automáticamente alguno de esos objetos en una imagen. Las imágenes que servirán para entrenar la red deben estar etiquetadas, ya que la red comprenderá las características específicas del objeto en cuestión al asociarlas con la categoría a la que pertenece. En cada capa de la red neuronal se toman datos de la anterior, que son transformados y pasados a la siguiente, así que la red se vuelve cada vez más compleja. La red aprende directamente a partir de los datos, por lo que no podremos influenciar sobre las características que esté aprendiendo [27].

3.3. Red neuronal convolucional

Las redes neuronales convolucionales se usan a menudo para técnicas de aprendizaje profundo, especialmente para las que usan imágenes como datos de entrada. La arquitectura de las redes neuronales convolucionales está inspirada en la estructura biológica del córtex o corteza visual, la cual está formada por células simples y complejas. Dichas células se activan en base a las subregiones de un campo visual, llamadas campos receptivos. En ello se inspiran estas redes, ya que una capa se conecta a las subregiones de la capa anterior, en lugar de estar completamente conectadas, como en otro tipo de redes neuronales [28].

Otra característica de una red neuronal convolucional, que la diferencia del resto, es que sus neuronas producen resultados correlacionados espacialmente, mientras que las de otros tipos de redes no comparten ninguna conexión, produciendo resultados independientes. Una red neuronal convolucional consta de distintos tipos de capas: capas convolucionales (convolutional layer), capas de agrupación máxima (max-pooling layer) o de acumulación media (average-pooling layer) y capas completamente conectadas (fully-connected layer) [29].

La arquitectura de una red neuronal convolucional puede variar según los tipos y número de capas incluidas, y se seleccionan en función de si se precisa una respuesta categórica o continua, la cantidad de datos a aprender y si se encuentran en escala de grises o a color, etc. Por lo general, la red constará de una capa de entrada, una de salida y diversas capas entre estas dos. Dichas capas pueden dividirse en dos grupos, las capas de detección de funciones y las de clasificación.

Entre las capas de detección de funciones encontramos tres tipos según la operación que realicen: convolución, agrupación y rectificado de la unidad lineal. La primera coloca las imágenes de entrada a través de un conjunto de filtros convolucionales, cada uno de los cuales activa ciertas características de las imágenes. Las de agrupación simplifican la salida ya que realizan un muestreo descendente no lineal para reducir el número de parámetros sobre los que la red necesita aprender. Por último, las de rectificado, permiten que el entrenamiento sea más rápido y efectivo mapeando valores negativos a cero y manteniendo valores positivos. Estas tres operaciones se repiten en decenas o cientos de capas, con cada una de ellas la red aprende a detectar diferentes características [30].

Una vez se detectan las características, la arquitectura de la red se centra en realizar operaciones de clasificación. La penúltima capa de toda red neuronal convolucional es una capa totalmente conectada, la cual genera un vector cuya dimensión se corresponde con el número de clases en las que puede clasificarse la imagen. En la capa final se utiliza una función *softmax* que proporciona la salida, es decir, el resultado de la clasificación [31].

3.4. Neural Network Toolbox en MatLab

Para realizar la red neuronal convolucional se ha utilizado *Neural Network Toolbox* en MatLab. Esta proporciona comandos simples de MatLab para crear e interconectar las capas de una red neuronal profunda [32].

El flujo de trabajo que suele seguirse para diseñar una red neuronal en MatLab es la característica de un trabajo en cascada, formado por los siguientes pasos principalmente: Recoger datos, crear red, configurar la red, entrenar la red, validar la red y usarla [33].

El entrenamiento de un modelo de aprendizaje profundo puede tomar horas, días o semanas, dependiendo del tamaño de los datos y la cantidad de potencia de procesamiento que tenga disponible. Para evitar la necesidad de usar tantos

datos y agilizar la creación de la red neuronal, se usa una técnica conocida como *transfer learning*.

El aprendizaje de transferencia, toma una red preentrenada y la usa como punto de partida para aprender la nueva tarea que se requiera. Adaptar una red preentrenada a nuestras necesidades es además de más rápido, más fácil que construir y entrenar de cero una nueva red. La red predefinida ya ha aprendido un amplio conjunto de características que puede utilizar en el nuevo entrenamiento con nuestro conjunto de datos [34].

Actualmente existen tres opciones de computación, y elegir la adecuada para el modelo que se esté realizando también es una pieza clave en el trabajo. Estas opciones son: basada en CPU, basada en GPU y basada en la nube. El cálculo basado en CPU es la opción más simple y disponible, recomendado solo para ejemplos simples utilizando una red predefinida. El uso de una GPU reduce el tiempo de entrenamiento de la red de días a horas. El cálculo de la GPU basado en la nube no necesita que se compre ni configure el hardware.

Capítulo 4: Desarrollo de los modelos neuronales

Tal y como se ha mencionado en el capítulo 1 de esta memoria, el trabajo sigue una estructura de desarrollo en cascada, metodología usada generalmente en trabajos de investigación, como es el caso. En este capítulo, se exponen las fases del desarrollo de las distintas redes neuronales utilizadas en el estudio.

4.1. Recogida de datos

La recogida de datos, aparentemente la tarea más sencilla del trabajo, fue una de las más agotadoras. Aunque en una primera instancia parecía que no sería necesario mucho esfuerzo para realizarla, resultó convirtiéndose en un desafío que no he sido capaz de lograr.

Cuando comencé el trabajo, mi objetivo fue recopilar al menos 1000 imágenes, entre melanomas y nevos. Evidentemente, esas imágenes no podían ser realizadas por mí. Las imágenes debían ser de fuentes oficiales, como bases de datos de otros trabajos, o páginas dedicadas al cáncer de piel. Esas imágenes debían haber superado el diagnóstico correspondiente. No tendría sentido usar una imagen de un nevo pensando que es benigno cuando podría no serlo y viceversa, por ello había que ser cauteloso en la selección.

Las primeras imágenes fueron fáciles de conseguir, pero tras un mes estancada en esta tarea, no había podido recopilar más de 533 imágenes (329 melanomas y 204 nevos). De estas imágenes, 93 melanomas y 87 nevos pertenecían a la base de datos de un trabajo sobre la extracción de lesiones cutáneas de imágenes no dermatoscópicas. Esta base de datos no solo tenía las imágenes originales de melanomas y no melanomas, sino que además proporcionaba la segmentación de cada una de ellas [35]. Estas 180 imágenes han sido las seleccionadas para realizar las primeras pruebas de la red.

4.2. Tratamiento de la imagen

La tercera fase consiste en el preprocesamiento de las imágenes. Las imágenes han sido tratadas de cuatro maneras distintas, para determinar cuál era la manera adecuada de usar las imágenes.

La primera manera ha sido la más simple. Se ha usado la imagen original, a la que se le ha aplicado un ajuste de brillo y contraste y un filtro de tipo Gaussiana para suavizarla. En la figura 8 aparece un ejemplo de este procesamiento.



Figura 8: Antes y después de la imagen 'NM1_orig.jpg' tras pasar por el procesamiento del filtro gaussiana. A la izquierda la original y a la derecha la procesada.

En segundo lugar, se ha utilizado la imagen segmentada. Para ello se ha tomado las que proporcionaba la base de datos mencionada previamente en el apartado 4.1.1. En la figura 9 se aprecia a la izquierda la imagen original y la segmentada a la derecha.



Figura 9: A la izquierda imagen original 'NM1_orig.jpg' y a la derecha su segmentada 'NM1_contour.png'.

Por otro lado, he recortado las imágenes originales, tras pasarlas por un filtro gaussiano, en función de la *bounding box* de su imagen segmentada, para así descartar las partes de la imagen que no nos proporcionen información del melanoma o el lunar en sí. Han sido recortadas teniendo en cuenta que las dimensiones fueran las de un cuadrado, ya que es como se las pasaremos posteriormente a la red neuronal. El algoritmo creado para recortarlas puede encontrarse en el script llamado "CropBounBox.m". En la figura 10 se encuentran a la izquierda la imagen original y a la derecha la que ha quedado al aplicarle el procesamiento.



Figura 10: A la izquierda la imagen original. A la derecha la imagen tras realizarse el recorte en función de su bounding box y un filtro gaussiano.

Por último, se ha probado la red con una superposición de la imagen original y la imagen segmentada correspondiente, quedando así solo la información del melanoma o nevo. El algoritmo creado para realizar la superposición se encuentra en el script llamado "Superpos.m". El resto de la imagen se verá negra, tal y como se expone a la derecha de la figura 11.



Figura 11: Imagen original a la izquierda y a la derecha la superposición de la imagen original y su correspondiente imagen segmentada.

Una vez obtenidas las cuatro carpetas con los cuatro tipos de imágenes que vamos a proporcionar a la red, se ha aplicado utilizando MatLab para rotar 90° tres veces cada imagen, consiguiendo así cuadruplicar los datos, como se aprecia en la figura 12. Para ello se han utilizado los scripts “Rotate.m”, “Rotate_all.m”, “Rotate_bounbox.m” y “Rotate_SuPo.m”.



Figura 12: Melanoma número 13 cuadruplicado tras las tres rotaciones.

4.3. Creación y configuración de la red

Cuando uno se dispone a usar aprendizaje profundo dispone de dos opciones: crear la red desde cero o usar *transfer learning*, es decir, crear una red a partir de una ya existente y entrenada. En un primer lugar, se optó por crear la red desde cero. Para seleccionar la arquitectura de la red, en primer lugar, se usó una básica, basada en el ejemplo que Mathworks proporciona en su página [36]. A partir de esta, adaptándola al proyecto actual, se llegó a una versión preliminar del trabajo, la cual puede encontrarse en el script llamado “CNN_0.m”.

El algoritmo utilizado para conseguir esta arquitectura, formada por tres capas convolucionales de 8, 16 y 32 filtros respectivamente, sería la siguiente:

```
layers = [  
    imageInputLayer(sz)  
  
    convolution2dLayer(3,8,'Padding',1)  
    batchNormalizationLayer  
    reluLayer
```

```

maxPooling2dLayer(2, 'Stride', 2)

convolution2dLayer(3, 16, 'Padding', 1)
batchNormalizationLayer
reluLayer

maxPooling2dLayer( 2, 'Stride', 2 )

convolution2dLayer(3, 32, 'Padding', 1)
batchNormalizationLayer
reluLayer

fullyConnectedLayer(2)
softmaxLayer
classificationLayer];

```

Aunque posteriormente se realizaron cambios en la arquitectura para realizar diversas pruebas, en todas se mantenían el tipo de capas utilizadas. Para comenzar, es indispensable usar la *ImageInputLayer*, o capa de entrada de imagen. En ella se especifica el tamaño de la imagen, en nuestro caso hemos almacenado en una variable llamada “sz” el tamaño de la primera imagen de la *datastore* creada en cada caso.

Las últimas tres capas son indispensables. La primera de estas tres es una capa completamente conectada, *fullyConnectedLayer*. Esta capa conecta todas las neuronas de la capa anterior, es decir, combina todas las características aprendidas en las capas anteriores para clasificar la imagen. El parámetro de salida de esta capa deberá ser igual al número de clases de datos, en este caso, dos clases: melanoma y nevo.

La capa que sigue es *softmaxLayer*, cuya función es normalizar la salida de la anterior. La salida de esta capa consiste en una serie de números positivos que en total suman la unidad, los cuales son utilizados por la siguiente capa: *classificationLayer*. Esta es la capa de clasificación y la final de la red. usa las probabilidades de *softmaxLayer* para asignar cada entrada a una de las posibles clases y calcular la pérdida.

Entre la primera y las tres últimas, se encuentran las capas que editaremos para obtener los resultados que buscamos. Usamos cuatro tipos de capas en

diferentes configuraciones cuyos comandos en MatLab son: *convolution2dLayer*, *batchNormalizationLayer*, *reluLayer* y *maxPooling2dLayer*.

En una capa convolucional, el primer argumento se corresponde con el tamaño del filtro, el alto y el ancho que usa la función que escanea las imágenes durante el entrenamiento de la red. En este caso tanto el ancho como el alto se corresponden con el número 3. El segundo argumento es el número de filtros, es decir, el número de neuronas que se conectan a una misma región de entrada. Los siguientes dos argumentos, *Padding* y 1, agregan una fila de relleno a la parte superior e inferior, y una columna de relleno a la izquierda y la derecha de la entrada de la capa.

La *batchNormalizationLayer* normaliza las activaciones y gradientes que se propagan a través de la red, haciendo que el entrenamiento sea un problema de optimización más fácil. Utilizar estas capas acelera el entrenamiento de la red y reduce la sensibilidad a la inicialización de la red. La capa *ReLU* es una capa de activación no lineal que aparece a continuación de la anterior.

La *maxPooling2dLayer* es una capa de agrupación máxima. Se utiliza para reducir las operaciones de muestreo descendente producidas tras las capas convolucionales. Este muestreo tiene la intención de reducir el tamaño espacial del mapa de características y eliminar la información espacial redundante. La capa de agrupación máxima devuelve los valores máximos de las regiones rectangulares de las entradas, especificadas por el primer argumento, *poolSize*. En nuestro caso, el tamaño de la región rectangular es [2,2]. Los siguientes dos argumentos se corresponden con *Stride* y 2, lo que especifica el tamaño de paso, tanto horizontal como vertical, que toma la función de entrenamiento mientras escanea a lo largo y ancho de la entrada.

Tras realizar algunas pruebas con esta versión no se obtuvieron resultados concluyentes, así que aumenté las capas convolucionales de la red, llamando al nuevo script "CNN_1.m". La arquitectura quedó de la siguiente manera:

```

layers = [
    imageInputLayer(sz)

    convolution2dLayer(3,8,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3, 16,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer( 2,'Stride',2 )

    convolution2dLayer(3, 32,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer( 2,'Stride',2 )

    convolution2dLayer(3, 64,'Padding',1)
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer( 2,'Stride',2 )

    convolution2dLayer(3, 128,'Padding',1)
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer];

```

La diferencia es que pasa de tener 3 capas convolucionales a tener 5. Los resultados mejoraron, pero aun así se buscó hacerla más eficiente. En un primer intento, se mantuvo el número de capas convolucionales de la versión 1, pero se cambiaron sus argumentos. La arquitectura de esta red se encuentra en el script de título “CNN_2.m”. El objetivo era conseguir una cobertura completa de la entrada de la capa. Para ellos usamos tanto *Stride* como *Padding* en los argumentos de sus capas convolucionales. Para asegurarnos de que la cobertura sea completa, las dimensiones horizontal y vertical de la salida deben ser iguales. La arquitectura sería la siguiente:

```

layers = [
    imageInputLayer(sz)

    convolution2dLayer([6 4],8, 'Stride',1, 'Padding',[1 0])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride',2)

    convolution2dLayer([6 4],16, 'Stride',1, 'Padding',[1 0])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride',2)

    convolution2dLayer([6 4],32, 'Stride',1, 'Padding',[1 0])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride',2)

    convolution2dLayer([6 4],64, 'Stride',1, 'Padding',[1 0])
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2, 'Stride',2)

    convolution2dLayer([6 4],128, 'Stride',1, 'Padding',[1 0])
    batchNormalizationLayer
    reluLayer

    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer];

```

Al observar que los resultados no eran suficientemente buenos, se creó una nueva red. En este caso, creada a partir de la red preentrenada AlexNet. Para realizarla usaríamos todas las capas de AlexNet, menos las 3 últimas, que debimos adaptarlas a nuestra clasificación [37]. La arquitectura, que puede encontrarse en el script “CNN_3.m”, sería la que se muestra a continuación:

```
net = alexnet;
```

```
layersTransfer = net.Layers(1:end-3);

layers = [
    layersTransfer
    fullyConnectedLayer(2, 'WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10)
    softmaxLayer
    classificationLayer];
```

Los resultados obtenidos con las cuatro versiones se encuentran en el capítulo siguiente.

Capítulo 5: Entrenamiento y resultados

5.1. Conjuntos de datos

Las versiones descritas anteriormente se han probado con diversas bases de datos. Las primeras tres versiones, CNN_0, CNN_1 y CNN_2 se han probado con cuatro conjuntos de datos distintos. Dentro de la carpeta que contiene cada conjunto de datos, las imágenes se dividen en dos subcarpetas de nombre “melanoma” y “nevo”.

La primera carpeta, de nombre “Originales”, contiene en total 180 imágenes. Estas son las originales de la base de datos que seleccionamos para realizar las primeras pruebas. El conjunto de imágenes originales obtenido tras las rotaciones, un total de 720 imágenes, se encuentra en la carpeta “Originales_x4”.

La segunda, “Segmentadas” contiene las 180 imágenes que se obtuvieron al segmentar las originales. Las 720 imágenes que se consiguen tras rotarlas se almacenan en “Segmentadas_x4”.

La tercera y la cuarta carpeta, de nombre “Data_BounBox” y “Data_SuPo”, contienen respectivamente las imágenes recortadas en función de la *bounding box* de la lesión y las creadas al superponer las imágenes originales y sus segmentadas. Al rotarlas se consigue tener 720 imágenes en la carpeta “Data_BounBox_x4” y otras 720 en la carpeta “Data_SuPo_x4”.

Por último, se encuentra la carpeta titulada “all_data”. Esta contiene todas las imágenes recopiladas durante la recogida de datos, un total de 533 imágenes entre melanomas y nevos. Al rotarlas, tal y como hemos hecho con los conjuntos anteriores, contamos con 2132 imágenes que se almacenaron en la carpeta “all_datax4”.

Las pruebas se realizaron con los conjuntos de datos cuadruplicados, es decir, con las carpetas obtenidas tras las rotaciones.

5.2. Criterios de evaluación

Para evaluar los resultados se han tenido en cuenta cinco criterios de evaluación. El primero de ellos ha sido llamado “Precisión”. Este mide la exactitud del método dividiendo todos los casos clasificados correctamente entre el número total de casos evaluados. Así podemos conocer el tanto por ciento de acierto del método. La ecuación utilizada sería la siguiente:

$$\text{Precisión} = \frac{\text{Casos clasificados correctamente}}{\text{Casos evaluados}} \times 100$$

Por otro lado, obtenemos el porcentaje de falsos positivos, “FP”, y falsos negativos, “FN”. He definido los falsos positivos como el conjunto de imágenes clasificadas como melanomas que no lo eran, utilizando la ecuación que se expone a continuación:

$$\text{Falso Positivo o FP} = \frac{\text{Casos clasificados erróneamente como melanoma}}{\text{Casos clasificados como melanoma}} \times 100$$

Como falsos negativos se ha definido el conjunto de imágenes que por el contrario sí eran melanomas, pero fueron clasificados como nevos. Estos últimos son los que más deben preocuparnos, pues dar un resultado erróneo en este sentido podría suponer la despreocupación del paciente, consiguiendo lo contrario a lo que buscamos. La ecuación usada para calcularlo fue:

$$\text{Falso Negativo o FN} = \frac{\text{Casos clasificados erróneamente como nevo}}{\text{Casos clasificados como nevo}} \times 100$$

Además de estos tres criterios de evaluación, tendremos en cuenta la sensibilidad y especificidad de la prueba. La sensibilidad es la probabilidad de clasificar correctamente un melanoma, también conocida como la fracción de verdaderos positivos. La ecuación que calcula este dato es:

$$\text{Sensibilidad o FVP} = \frac{\text{Casos clasificados correctamente como melanoma}}{\text{Casos clasificados como melanoma}} \times 100$$

Por el contrario, la especificidad es la probabilidad de clasificar correctamente un nevo, es decir, la fracción de verdaderos negativos. Para calcularlo usamos una ecuación como la siguiente:

$$\text{Especificidad o FVN} = \frac{\text{Casos clasificados correctamente como nevo}}{\text{Casos clasificados como nevo}} \times 100$$

5.3. Resultados experimentales

En este apartado se exponen las opciones de entrenamiento utilizadas para realizar cada prueba, los resultados obtenidos y algunas gráficas del proceso de entrenamiento de la red. Los resultados obtenidos para la primera versión son los que aparecen en la tabla 1. Para entrenar la red se han usado el 60% de las imágenes, y se han utilizado las siguientes opciones de entrenamiento:

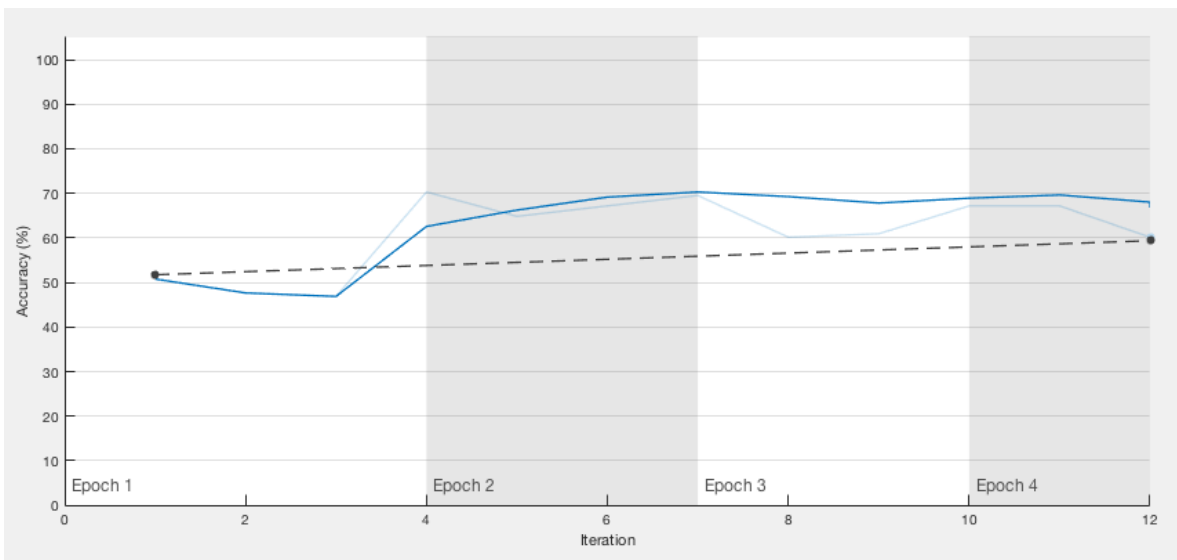
```
options = trainingOptions('sgdm',...
    'InitialLearnRate',0.01,...
    'MaxEpochs',4, ...
    'Shuffle','every-epoch',...
    'ValidationData', EvalDS, ...
    'ValidationFrequency',30, ...
    'Verbose', false,...
    'Plots','training-progress');
```

Podemos observar que los mejores datos se obtienen con las imágenes originales, a las que simplemente hemos aplicado un filtro Gaussiano. A esta le sigue la prueba realizada con las imágenes recortadas en función de su *bounding box*.

CNN_0					
	Precisión (%)	FP (%)	FVP (%)	FN (%)	FVN (%)
Original	59.375	39.0411	60.9589	42.2535	57.7465
Segmentada	47.9167	52.381	47.619	52.0599	47.9401
BounBox	56.25	43.2749	56.7251	44.4444	55.5556
Superposicion	48.9583	48.6486	51.3514	51.8692	48.1308

Tabla 1: Resultados obtenidos al usar la versión CNN_0.

En la gráfica 1, obtenida durante el entrenamiento de la red de CNN_0 con las imágenes originales, se observa como la red no aprende lo suficiente. En el eje de abscisas las iteraciones. En el eje de ordenadas el porcentaje de acierto, es decir, la precisión. La línea azul se corresponde con el entrenamiento “alisado” (training smoothed), la línea celeste con el entrenamiento de la red en bruto (training) y la discontinua con la validación de la red (validation). Cada vez que aparece un punto en la línea discontinua debemos saber que en esa iteración del entrenamiento se ha realizado una validación preliminar de la precisión de la red.



Gráfica 1: Obtenida durante el entrenamiento de CNN_0 con las imágenes originales.

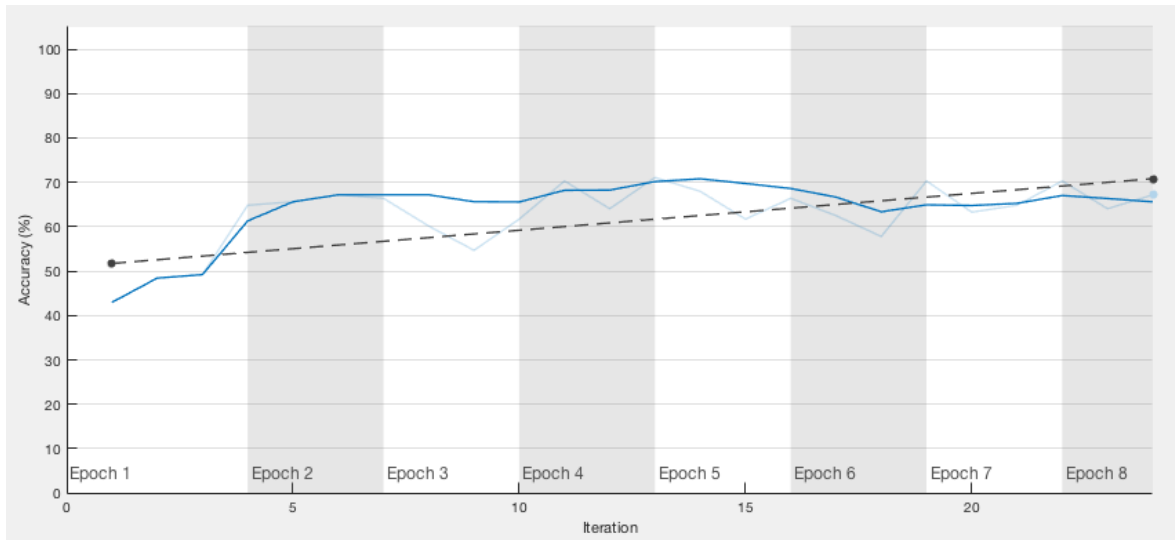
Al probar la versión siguiente, la CNN_1, se observó una mejora considerable en los resultados al usar las imágenes originales. Los resultados con los otros tres conjuntos de datos se mantuvieron estables, cuyos resultados pueden consultarse en la tabla 2.

CNN_1

	Precisión (%)	FP (%)	FVP (%)	FN (%)	FVN (%)
Original	70.83	21.24	78.76	34.29	65.71
Segmentada	50.3472	48.8638	51.1362	58.3333	41.6667
BounBox	53.125	37.037	62.963	49.1453	50.8547
Superposicion	51.0417	45.6522	54.3478	50.5102	49.4898

Tabla 2: Resultados obtenidos al usar la versión CNN_1.

En la gráfica 2, obtenida tras el entrenamiento de la red de CNN_1 con las imágenes originales, se aprecia como en este caso la validación mejora, dando un resultado final del 70.83 % de acierto.



Gráfica 2: Obtenida durante el entrenamiento de CNN_1 con las imágenes originales.

Tal y como puede apreciarse en la gráfica, las opciones de entrenamiento se variaron levemente, aumentando el número de épocas a 8, aunque el porcentaje de imágenes que se destinaría al entrenamiento se mantuvo en el 60%:

```
options = trainingOptions('sgdm',...
    'InitialLearnRate',0.01,...
    'MaxEpochs',8, ...
    'Shuffle','every-epoch',...
    'ValidationData', EvalDS, ...
    'ValidationFrequency',30, ...
    'Verbose', false,...
    'Plots','training-progress');
```

Tras observar los resultados se realizó una nueva prueba con esta versión. En este caso se usarían todos los datos obtenidos, es decir, 2.132 imágenes. Además, aumentamos el porcentaje de imágenes que se usan para el entrenamiento. Mientras que todas las pruebas expuestas en las tablas anteriores se realizaron usando un 60% de los datos para entrenar y un 40% para evaluar el entrenamiento, en esta nueva prueba se usará un 70% de imágenes para entrenar

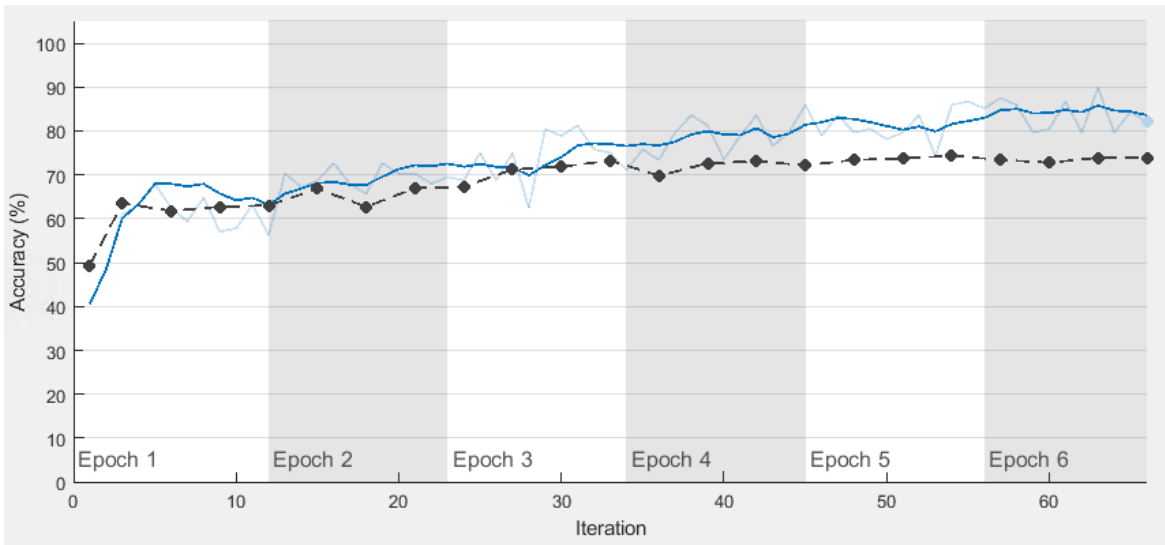
y un 30% para evaluar. El resultado obtenido fue el que aparece a continuación en la tabla 3.

CNN_1

	Precisión (%)	FP (%)	FVP (%)	FN (%)	FVN (%)
Todas	72.956	20.2667	79.7333	36.7816	63.2184

Tabla 3: Resultados al probar la versión CNN_1 con todos los datos, 2132 imágenes.

Las opciones de entrenamiento variaron, seleccionando el número 6 para la variable 'MaxEpochs' y minimizando drásticamente el valor que se le había dado a 'ValidationFrequency', pasando de 30 a 3 como se aprecia en la gráfica 3. De esta manera, podemos apreciar de forma más clara como van mejorando los valores de precisión a medida que aprende la red.



Gráfica 3: Obtenida durante el entrenamiento de CNN_1 con todas las imágenes.

La tercera versión, CNN_2, no proporcionó los resultados esperados en un primer momento. Se esperaba una mejora de los resultados, al realizarse una cobertura completa de los datos, pero por el contrario la precisión de la red disminuyó, tal y como puede apreciarse en la tabla 4. Las opciones de entrenamiento fueron las mismas que se usaron en las primeras pruebas con CNN_1. No se observó mejora, pero sí vuelve a destacar la prueba realizada con las imágenes originales frente a las demás.

CNN_2

	Precisión (%)	FP (%)	FVP (%)	FN (%)	FVN (%)
Original	64.24	38.94	61.06	27.50	72.5
Segmentada	38.1944	57.2139	42.7861	72.4138	27.5862
BounBox	54.5139	45.5882	54.4118	45.2381	54.7619
Superposicion	53.125	25	75	49.2308	50.7692

Tabla 4: Resultados obtenidos al usar la versión CNN_2.

Viendo que los resultados no mejoraban demasiado, y que podría deberse a la poca cantidad de datos, se probó la opción de realizar una versión que usara *transfer learning* a partir de AlexNet. Para entrenarla y evaluarla se usaron las imágenes originales, es decir, 720 imágenes. Los resultados fueron favorables tal y como puede apreciarse en la tabla 5.

CNN_3

	Precisión (%)	FP (%)	FVP (%)	FN (%)	FVN (%)
Originales 1	81.4815	20.9677	79.0323	15.2174	84.7826
Originales 2	87.037	14.4068	85.5932	11.2245	88.7755

Tabla 4: Resultados obtenidos al probar la versión CNN_3 con las 720 imágenes originales.

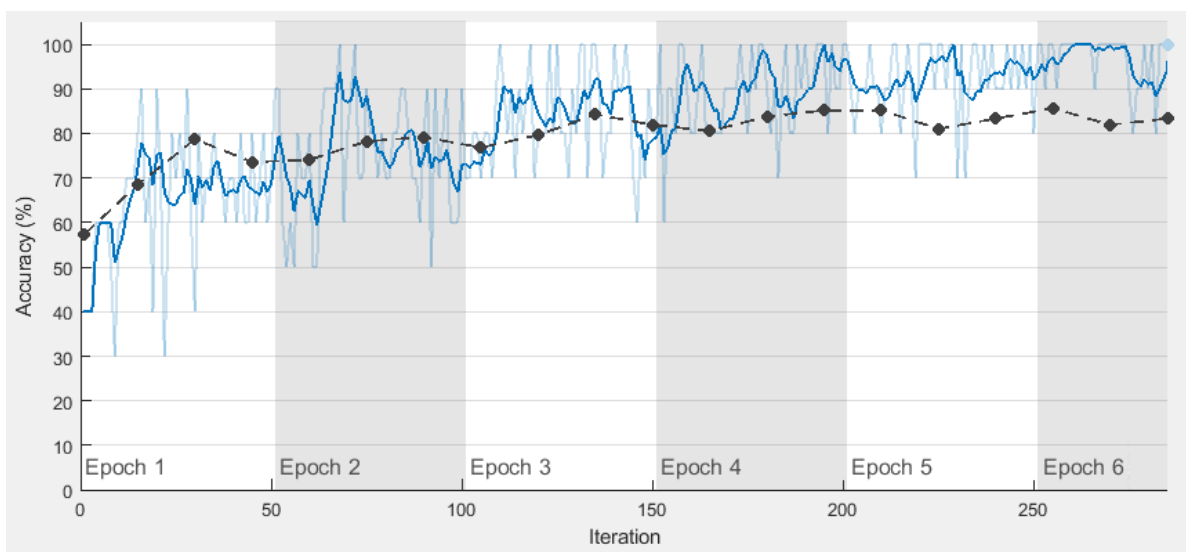
En este caso se usaron un 70% de los datos para entrenar y un 30% para evaluar el método. Las opciones de entrenamiento usadas en un primer momento fueron las siguientes:

```
% options 1
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',6, ...
    'InitialLearnRate',1e-4, ...
    'ValidationData',EvalDS, ...
    'ValidationFrequency',3, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

Siguiendo en esta línea se variaron las opciones de entrenamiento y se obtuvieron resultados aún mejores:

```
%options 2
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',15, ...
    'InitialLearnRate',1e-4, ...
    'ValidationData',EvalDS, ...
    'ValidationFrequency',15, ...
    'Verbose', false, ...
    'Plots', 'training-progress');
```

Como puede observarse en la gráfica 4, obtenida durante el entrenamiento usando la opción 2, este método no finaliza el entrenamiento cuando acababa de realizar todas las iteraciones sino cuando este cumplía el criterio de validación (*met validation criterion*). Por defecto, si el error del subconjunto de validación aumenta durante 6 épocas seguidas, el entrenamiento termina. La primera prueba, con la opción 1 cuya frecuencia de validación era cada 3 iteraciones, acabó en la iteración 87 de 300. La segunda opción siguió aprendiendo hasta la iteración 285 de 750, ya que la frecuencia de validación era menor, cada 15 iteraciones.



Gráfica 4: Obtenida durante el entrenamiento de CNN_3 con las imágenes originales y la opción de entrenamiento número 2.

Capítulo 6: Conclusiones y futuro

6.1. Conclusión del TFG

Entre las evidencias que he observado durante el desarrollo del proyecto se encuentra la del tipo de imágenes que daban mejor resultado. En todas las pruebas, el conjunto de imágenes que daba una exactitud más alta era el formado por las imágenes originales, a las que únicamente aplicábamos un filtro gaussiano. Cuanta más información reciba la red, aunque esa información no sea específicamente la lesión sino su alrededor, mejor aprende la red.

Por otro lado, queda claro que con el número de imágenes conseguido la mejor opción es usar *transfer learning*. Una red neuronal creada de cero necesita cientos de miles, o incluso millones de imágenes, para dar unos buenos resultados. En cambio, cuando se usa una red preentrenada como puede ser AlexNet es suficiente con un número mucho menor de imágenes. Aunque evidentemente, tal y como hemos podido observar en los resultados, cuanto más sean los datos mejores serán los resultados.

En cuanto a mis sensaciones y conclusión en lo que refiere al TFG en general, podría afirmar que mi principal reto ha sido recopilar las imágenes. Lo que en un primer momento parecía una tarea fácil de lograr, se convirtió en una de las fases más desesperantes del proceso.

Es evidente que el número de imágenes obtenido finalmente no ha sido el esperando al iniciar el proyecto. Apenas he llegado a la mitad del objetivo que me había planteado y aun así muchas de las imágenes obtenidas no tienen toda la calidad que perseguía. Además, al ser de distintas fuentes, las imágenes son muy diferentes entre ellas en cuanto a calidad de imagen, colores, luz, etc. Por mucho que se preprocese, se pierden datos muy valiosos para el entrenamiento de la red, y ya se perdía la posibilidad de evaluar características como el diámetro de la lesión y su evolución, como para también perder datos sobre su color o sus bordes. Ya de por sí, los melanomas y los nevos tienen muchas cosas en común que puedan

llevar a confusión. Hay melanomas que pueden parecer nevus y viceversa. Es todo un reto conseguir un 100% de exactitud.

En cuanto al entorno utilizado, no podía ser otro. Durante toda la carrera, desde el primer curso hasta el último, MatLab me ha acompañado. Estoy cómoda con el entorno y lo conozco bien, no he parado de descubrir durante estos años nuevas capacidades y posibilidades que desconocía y MatLab ofrece. En este caso, Neural Network Toolbox. Ha sido un descubrimiento puesto que no había trabajado durante mi paso por la carrera con redes neuronales. Aun así, no ha sido complicado obtener los conocimientos necesarios para manejarla en ella, MathWorks ofrece cursos online sobre Deep learning que he tenido la posibilidad de aprovechar y toda la información necesaria en su página web como para aprender a utilizar y comprender Neural Network Toolbox.

Cabe destacar que no fue posible realizar las pruebas con una GPU. Todas las pruebas se realizaron con ordenadores y servidores que contaban con una única CPU. Esto hizo tedioso el proceso de entrenamiento y evaluación de las redes. Realizar pruebas para variar y probar distintas configuraciones llevaba mucho tiempo. Aun así, como el número de imágenes no era elevado, ha sido posible realizar las pruebas.

Si se ha conseguido obtener un 87% de exactitud con las imágenes que formaban la base de datos y sin un ordenador adecuado, me pregunto qué resultados podrían obtenerse en mejores condiciones.

6.2. Futuras posibles ampliaciones del trabajo

Evidentemente, los resultados no son lo suficientemente buenos como para usarlos de forma clínica. Podría estar en juego la salud y vida de las personas, por lo que la clasificación debe ser totalmente fidedigna. Para lograrlo, mejorar la base de datos es el principal reto. Fotos con las mismas características, realizadas con un mismo dispositivo. Aumentar mucho el número de datos, que sean alrededor de los cientos de miles.

Además, una vez se obtuvieran unos buenos resultados, podría ampliarse la capacidad del algoritmo. Por ejemplo, podría utilizarse algún artilugio en forma de marco para situar sobre la lesión a la hora de realizar la fotografía, pudiéndose comparar en función de esa imagen las diferencias de tamaño. Una de las características que definen un melanoma es que mida más de 6 milímetros de diámetro, y esa información, al obtener las imágenes de internet, se pierde.

Por otro lado, una forma de mejorarlo sería realizar algoritmo que almacenara las imágenes de un mismo paciente y no solo la clasificara. Podría comparar el algoritmo entre la nueva imagen obtenida del paciente y las anteriores, pudiendo crear una alerta en caso de que hubiera sufrido una evolución brusca o fuera de lo común en un periodo de tiempo determinado. Una evolución rápida puede ser signo de melanoma.

Referencias bibliográficas

- [1] Martínez Escribano J, Piñero Madrona A, De Torre Minguela C, Ródenas López JM. *Melanoma cutáneo: Guía clínica práctica*. Consejería de Sanidad y Política Social Servicio Murciano de Salud, 2012.
- [2] Redacción (13 de junio de 2018). El cáncer de piel aumenta su incidencia en España un 10 por ciento cada año. *La Vanguardia*. Disponible online en: <https://www.lavanguardia.com/vida/20180613/4589287325/el-cancer-depiel-aumenta-su-incidencia-en-espana-un-10-por-ciento-cada-ano.html>
- [3] Evolución del melanoma. Asociación Española Contra el Cáncer. [Consulta: 3 septiembre 2018]. Disponible en: <https://www.aecc.es/es/todo-sobre-cancer/tipos-cancer/melanoma/evolucion-melanoma>
- [4] Darrell S. Rigel, Robert J. Friedman, Leonard M. Dzubow, Douglas S. Reintgen, Jean-Claude Bystry, and Robin Marks. *Cáncer de piel. MB BS MPH FRACP FACD*, ISBN: 978-84-8174-875-8
- [5] Tortora, Derrickson. Principios de Anatomía y Fisiología. Editorial médica panamericana. 11ª edición. México DF. 2006. P.147.
- [6] Li G, Satyamoorthy K, Meier F, Berking C, Bogenrieder T, Herlyn M (2003) Function and regulation of melanoma-stromal fibroblast interactions: when seeds meet soil. *Oncogene* 22:3162–3171
- [7] Tortora, Derrickson. Principios de Anatomía y Fisiología. Editorial médica panamericana. 11ª edición. México DF. 2006. P.151.
- [8] Tortora, Derrickson. Principios de Anatomía y Fisiología. Editorial médica panamericana. 11ª edición. México DF. 2006. P.152.
- [9] Tortora, Derrickson. Principios de Anatomía y Fisiología. Editorial médica panamericana. 11ª edición. México DF. 2006. P.147.
- [10] La piel normal [fotografía]. En: *¿Qué es el cáncer de piel tipo melanoma?*. American Cancer Society. [Consulta: 15 junio 2018]. Disponible en: <https://www.cancer.org/es/cancer/cancer-de-piel-tipo-melanoma/acerca/que-es-melanoma.html>
- [11] Carcinoma de células basales [fotografía]. En: *Carcinoma de células basales*. Skin Cancer Foundation. [Consulta: 15 junio 2018]. Disponible en:

- <http://cancerdepiel.org/cancer-de-piel/carcinoma-de-celulas-basales#panel1-3>
- [12] Carcinoma de células escamosas [fotografía]. En: *Carcinoma de células escamosas*. Skin Cancer Foundation. [Consulta: 15 junio 2018]. Disponible: <http://cancerdepiel.org/cancer-de-piel/carcinoma-de-celulas-escamosas#panel1-3>
- [13] Tortora, Derrickson. Principios de Anatomía y Fisiología. Editorial médica panamericana. 11ª edición. México DF. 2006. P.166.
- [14] Orlow SJ. Melanosomes are specialized members of the lysosomal lineage of organelles. *J Invest Dermatol* 1995; 105:3–7.
- [15] Melanoma en estadio 0 [fotografía]. En: Tratamiento del melanoma. Instituto Nacional del Cáncer. [Consulta: 20 junio 2018]. Disponible en: <https://www.cancer.gov/espanol/tipos/piel/paciente/tratamiento-melanoma-pdq>
- [16] Poole, Catherine M., and DuPont Guerry. *Melanoma: Prevention, Detection, and Treatment*, Yale University Press, 2005.
- [17] Melanoma de extensión superficial [fotografía]. En: Ángel López Ávila, Antonio Ramírez Andreo. *Diagnóstico y estadificación. Melanoma cutáneo: Guía clínica práctica*. Consejería de Sanidad y Política Social Servicio Murciano de Salud. 2012. Pag: 48.
- [18] Melanoma nodular [fotografía]. En: Ángel López Ávila, Antonio Ramírez Andreo. *Diagnóstico y estadificación. Melanoma cutáneo: Guía clínica práctica*. Consejería de Sanidad y Política Social Servicio Murciano de Salud. 2012. Pag: 48.
- [19] Martínez Escribano J, Piñero Madrona A, De Torre Minguela C, Ródenas López JM. *Melanoma cutáneo: Guía clínica práctica*. Consejería de Sanidad y Política Social Servicio Murciano de Salud. 2012. Pág: 45-47
- [20] ¿Qué es el cáncer de piel tipo melanoma?. American Cancer Society. [Consulta: 14 agosto 2018]. Disponible en: <https://www.cancer.org/es/cancer/cancer-de-piel-tipo-melanoma/acerca/que-es-melanoma.html>

- [21] Ángel López Ávila, Antonio Ramírez Andreo. *Diagnóstico y estadificación. Melanoma cutáneo: Guía clínica práctica*. Consejería de Sanidad y Política Social Servicio Murciano de Salud. 2012. Pág: 44-47
- [22] Mark Hudson Beale Martin T. Hagan Howard B. Demuth. *Deep Learning in Matlab. What is Deep learning?. Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. 2017.
- [23] What is the Difference Between Deep Learning and Machine Learning?. *Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [24] Mark Hudson Beale Martin T. Hagan Howard B. Demuth. *Deep Learning in Matlab. What is Deep learning?. Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. 2017.
- [25] *What makes deep learning state-of-the-art?. Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [26] *Inside a Deep neural network. Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [27] *How a Deep neural network learns. Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [28] Hubel, H. D. and Wiesel, T. N. " Receptive Fields of Single neurones in the Cat's Striate Cortex." *Journal of Physiology*. Vol 148, pp. 574-591, 1959.
- [29] Learn About Convolutional Neural Networks. MathWorks®. [Consultado: 4 julio 2018]. Disponible en:
<https://es.mathworks.com/help/nnet/ug/introduction-to-convolutional-neural-networks.html>
- [30] *About convolutional neural networks. Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [31] *About convolutional neural networks (continued). Introducing Deep learning with MATLAB*. The MathWorks, Inc. 2017.
- [32] Mark Hudson Beale Martin T. Hagan Howard B. Demuth. *Deep Learning in Matlab. What is Deep learning?. Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. 2017.
- [33] Mark Hudson Beale Martin T. Hagan Howard B. Demuth. *Neural Network Objects, Data, and Training styles. Workflow for Neural Network Design. Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. 2017.

- [34] Mark Hudson Beale Martin T. Hagan Howard B. Demuth. *Deep Learning in MATLAB. Start Deep Learning faster using transfer learning. Neural Network Toolbox™ User's Guide*. The MathWorks, Inc. 2017.
- [35] Mohammad H. Jafari, Ebrahim Nasr-Esfahani, Nader Karimi, S.M. Reza Soroushmehr, Shadrokh Samavi, Kayvan Najarian. Extraction of Skin Lesions from Non-Dermoscopic Images Using Deep Learning. 2017.
- [36] Create Simple Deep Learning Network for Classification. MathWorks ®. [Consultado: 7 julio 2018]. Disponible en:
<https://es.mathworks.com/help/deeplearning/examples/create-simple-deep-learning-network-for-classification.html;jsessionid=3428741fa1410a7fe9701543c9a7>
- [37] Get Started with Transfer Learning. MathWorks ®. [Consultado: 12 julio 2018]. Disponible en:
<https://es.mathworks.com/help/deeplearning/examples/get-started-with-transfer-learning.html;jsessionid=164612d910746a148d80fd016181>