UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques
Universitat de Barcelona

Redes completamente convolucionales en la segmentación semántica de lesiones melanocíticas

Autor: Fernando Moral Algaba

Director:	Dr. Simone Balocco
Realizado en:	Departament de Matemàtiques i Informàtica
Barcelona,	21 de junio de 2017

A Mari, Ada y Nayla.
Por el tiempo que os robé.

Agradecimientos

Quiero agradecer especialmente al Dr. Simone Balocco su ayuda, orientación y disponibilidad, su amistad y su eterna paciencia para conseguir que no me perdiese en elucubraciones. Sin su ayuda este proyecto no sería el mismo y aún debo decir más, tampoco yo lo sería.

Gracias también a todos los profesores que han invertido su tiempo y voluntad en formarme. Me es imposible agradecerles personalmente a todos y cada uno su ayuda, pues la lista es larga.

Gracias asimismo a mis compañeros de clase, Paul Quispe, Carlos Cortés, Dani Aparisi, Alicia Morales, Guillem Pascual, Juan Marín y todos aquellos que me han ayudado y de quienes tanto aprendí.

Abstract

Skin cancer is the more common type of cancer. Melanoma, that begins at melanocytes, is the most aggressive type of skin cancer and responsible of about 90 % of total deaths caused by this disease. Early diagnosis is the best way to defeat melanoma and can increase survival rate to near 100 %.

Studies on Automated image detection of skin lesion has evolved achieving high rates of accuracy on melanoma detection and classification.

Deep learning and Fully Convolutional Networks has become and useful tool on image analysis. This project explores the application of FCNs on semantic segmentation over combinations of two major datasets, images from dermatologic databases and skin mole images captured by cellular phone camera.

Trained nets has been tested over another two datasets of unseen images of skin moles and dermatologic images. Data generated at this study evidence high accuracy, precision, sensitivity and specificity rates despite the small database size, which is composed by only a few hundreds images.

Resum

El tipus de càncer mes comú es el de pell. El melanoma, que comença als melanocits, es el tipus de càncer de pell més agressiu y responsable d'aproximadament el 90 % del total de morts causades per aquesta malaltia. El diagnostic precoç es la millor manera de derrotar el melanoma i pot augmentar la taxa de supervivència fins prop del 100 %.

Els estudis de detecció automàtica en imatges de lesions de pell han evolucionat arribant a alts índexs d'exactitud a la detecció i classificació del melanoma.

L'aprenentage profund i les xarxes completament convolucionals han esdevingut una eina molt útil per a l'anàlisi d'imatges. Aquest projecte explora l'aplicació de FCNs a la segmentació semàntica de combinacions de dos conjunts de dades, imatges de bases de dades dermatològiques y imatges de pigues de la pell capturades amb la càmara d'un telèfon mòbil.

Les xarxes entrenades han estat provades amb dos conjunts de dades de noves imatges de pigues i dermatològiques.

Les dades generades a aquest estudi evidencien elevades taxes d'exactitut, precissió, sensibilitat i especificitat tot i el petit tamany de la base de dades, composada només per uns quants centenars d'imatges.

Resumen

El tipo de cáncer más común es el de piel. El melanoma, que comienza en los melanocitos, es el tipo de càncer de piel más agresivo y responsable de aproximadamente el 90 % del total de muertes causadas por esta enfermedad. El diagnóstico precoz es la mejor manera de derrotar el melanoma y puede aumentar la tasa de supervivencia hasta casi el 100 %.

Los estudios de detección automática en imágenes de lesiones de la piel han alcanzado altos índices de exactitud en la detección y la clasificación del melanoma.

El aprendizaje profundo y las redes completamente convolucionales se han convertido en una herramienta muy útil para el análisis de imágenes. Este proyecto explora la aplicación de FCNs en la segmentación semántica de combinaciones de dos conjuntos de datos, imágenes de bases de datos dermatológicas y imágenes de lunares de la piel capturados con la cámara de un teléfono móvil.

Las redes entrenadas han sido probadas sobre otros dos conjuntos de datos de nuevas imágenes de lunares de piel e imágenes dermatológicas. Los datos generados en este estudio evidencian elevadas tasas de exactitud, precisión, sensibilidad y especificidad a pesar del pequeño tamaño de la base de datos, que está compuesta por tan sólo unos pocos cientos de imágenes.

Glosario

α	Factor de aprendizaje
AUC	Área bajo la curva
BoVW	Bag Of Visual Words (Bolsa de palabras visuales)
CCB	Carcinoma de células basales
CCE	Carcinoma de células escamosas
CONV	Capa convolucional
FC	Fully connected, capa totalmente conectada
FCN	Redes completamente convolucionales, Fully Convolutional Networks
FN	Falso negativo
FP	Falso positivo
ILSVRC	Imagenet Large Scale Visual Recognition Challenge
ISIC	International Skin Image Collaboration
ReLU	Rectified Layer Unit (Unidad de capa rectificada).
ROI	Region of Interest (Región de interés)
tanh	Función de tangente hiperbólica
UVA	Radiación ultravioleta en el rango 320-400 nanómetros
UVB	Radiación ultravioleta en el rango 290-320 nanómetros
VN	Verdadero negativo
VP	Verdadero positivo

Índice de figuras

Índice de tablas

Tabla de contenidos

1. Introducción

Las acumulaciones de células pigmentarias o lunares, pueden evolucionar hacia un cáncer de piel, debe controlarse y consultar con el dermatólogo cualquier alteración en su forma o tamaño, especialmente si se produce alguna sensación pruriginosa.

La detección, diagnóstico y control de la evolución de estas lesiones puede resultar compleja. Por esta razón se ha desarrollado un método automatizado de análisis de imágenes que podría reportar múltiples beneficios, ya que si se diagnostica precozmente, la curación del melanoma, que es la forma más agresiva del cáncer de piel, es cercana al 100 % [5].

1.1. Objetivo del TFG

Nuestro objetivo principal consiste en explorar las posibilidades de las redes neuronales profundas en el análisis de imágenes dermatológicas. En ese sentido, nos interesa obtener datos preliminares sobre las bases de datos existentes y la metodología empleada por otros investigadores.

Por otro lado, queremos comprobar los resultados que podemos obtener entrenando una red completamente convolucional con tres conjuntos de datos distintos:

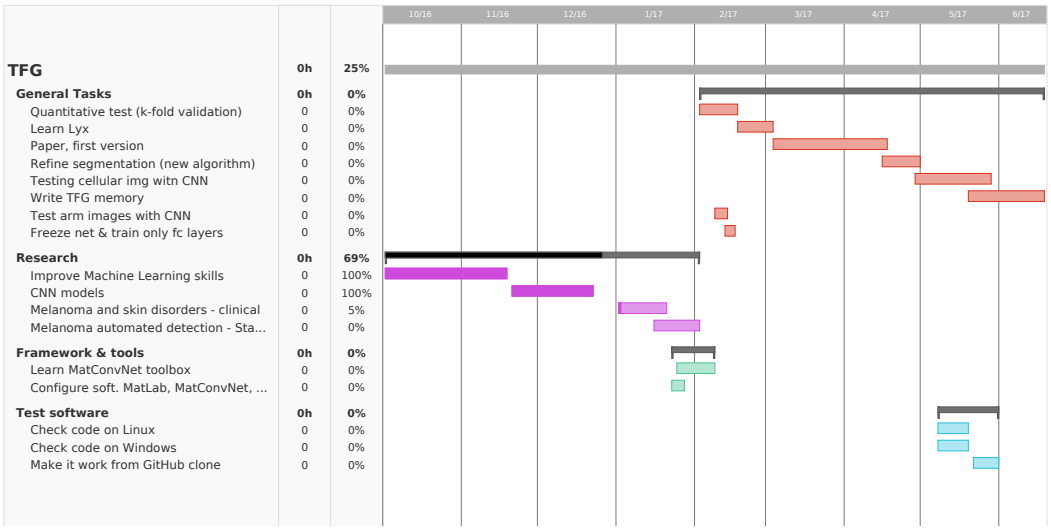
- Base de datos de melanomas
- Base de datos de acumulaciones pigmentarias, recortadas de una secuencia de imágenes de un brazo, en distintas condiciones de iluminación y adquiridas con la cámara de un teléfono móvil.
- Base de datos resultante de la reunión de las dos anteriores.

Una vez entrenadas estas redes, evaluaremos dos conjunto de validación de imágenes ajenas al conjunto de entrenamiento.

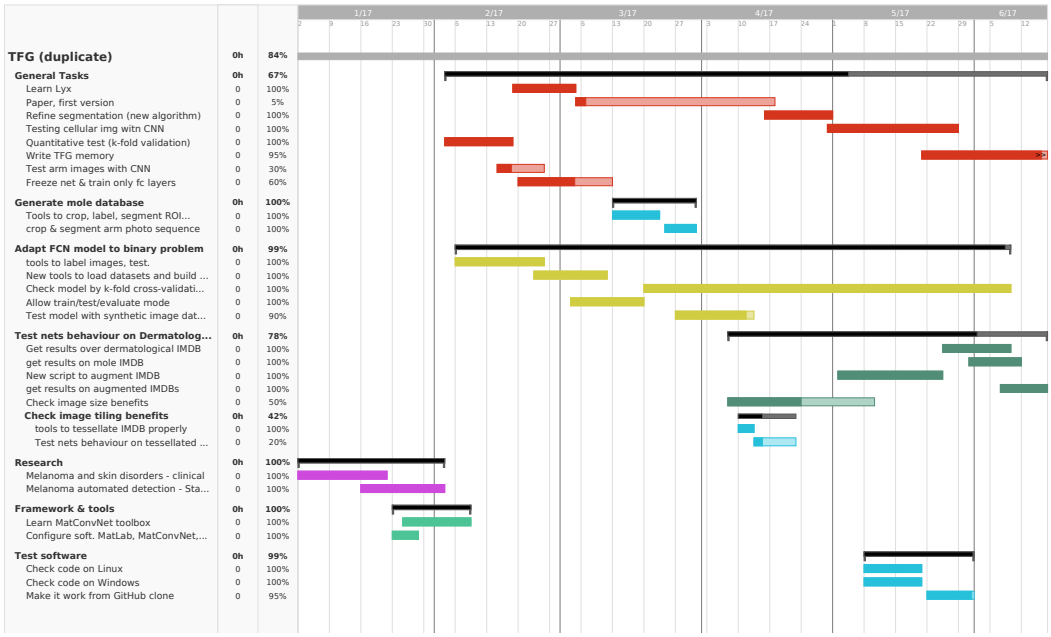
El primer conjunto se compone de 49 imágenes de melanomas y el segundo de 49 imágenes de lunares.

1.2. Planning TFG

Se han dedicado dos semestres a este trabajo con una dedicación estimada de 6 horas diarias. La imagen de la figura 1a corresponde al diagrama de Gantt en fecha 2 de Enero de 2017. Mientras que la de la figura 1b se trata de un esquema resumido del periodo comprendido entre el 2 de Enero y el 16 de Junio de 2017.



(a) Proyecto desde el 3 de octubre, Instantanea del diagrama el día 2 de enero de 2017



(b) 2 Enero al 16 Junio 2017

Figura 1: Diagramas de Gantt

Como se puede ver en los gráficos de la figura 1 el diagrama ha evolucionado desde una versión más ingenua a una mucho más elaborada. Esto ha respondido a las necesidades que surgían al realizar el proyecto y al hecho de que el nuevo conocimiento obtenido, con frecuencia, planteaba nuevas preguntas. La linea de tiempo de algunos puntos ha cambiado mucho, ha sido imposible respetar los plazos prefijados, puesto que algunas tareas resultaron ser más pesadas de lo que en un principio se pensó. La puesta a punto del modelo ha supuesto un esfuerzo continuo casi hasta el último instante del proyecto. Dicho modelo es un prototipo experimental desarrollado como ejemplo y que hace uso intensivo de la GPU dependiendo de diferentes herramientas de software que no siempre funcionan del modo esperado o están configuradas adecuadamente. Hemos hallado numerosas dificultades y "BUGs"que hemos resuelto y algunos que no se resolvieron sino que fue preciso sortear desactivando o sustituyendo alguna funcionalidad.

2. Problema clínico

2.1. Cáncer y cáncer de piel.

Cáncer es el nombre que reciben un grupo de enfermedades, caracterizadas por un aumento inusual en la actividad de la división celular. Esta actividad anómala, suele producirse por alteraciones en el material genético de las células. Las causas pueden ser diversas, exposición a ciertas radiaciones o a determinados compuestos químicos, causas hereditarias, infecciones víricas e incluso de modo natural, al producirse un fallo en el proceso de replicación del material genético. Este crecimiento celular anómalo puede estar localizado o diseminarse a otros tejidos del cuerpo, provocando lo que se conoce como metástasis, que es la expansión del cáncer a otros órganos distintos a aquel en que se originó.

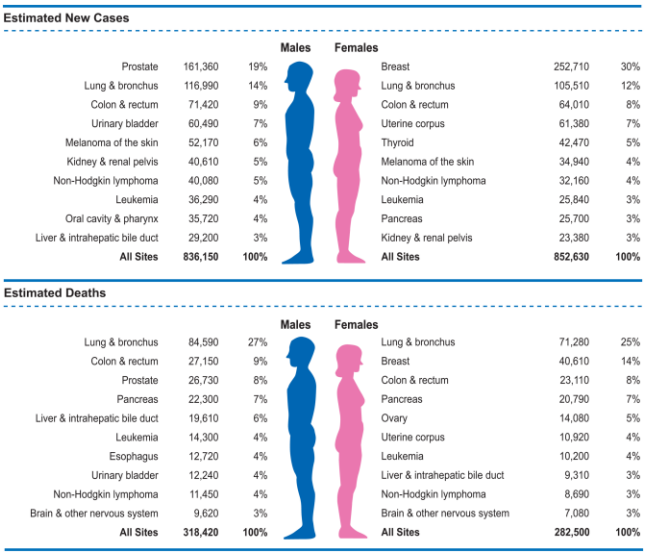
De todos los tipos de cáncer existentes el más habitual es el de piel. Según estimaciones de la American Cáncer Society [19][21] en 2017, en estados unidos, se producirán 5.400.000 nuevos cánceres de piel de tipo no-melanoma y casi 90.000 de tipo melanoma , frente a 1.600.000 de los de cualquier otra clase (ver figura: 2).

El principal desencadenante de esta enfermedad es la exposición a la luz solar, en concreto las radiaciones de tipo UVA y UVB.

La incidencia del melanoma, el más virulento de los tres, aun siendo mucho menor que la del Carcinoma de células basales (CCB) y Carcinoma de células escamosas (CCE), se ha incrementado en las últimas décadas en la población de piel más clara [2].

Aunque existen otros tipos, se suele hacer una clasificación para subdividir el cáncer de piel en tres categorías principales, CCB, CCE y melanoma.

CCB y CCE se definen como cánceres de tipo no melanoma. Esta distinción es relevante, ya que el de tipo melanoma es el de peor pronóstico, responde peor al tratamiento y se extiende a otros tejidos rápidamente, siendo la causa de la mayoría de muertes relacionadas con el cáncer de piel [2].



Estimated New Cases

			Males	Females			
Prostate	161,360	19%			Breast	252,710	30%
Lung & bronchus	116,990	14%			Lung & bronchus	105,510	12%
Colon & rectum	71,420	9%			Colon & rectum	64,010	8%
Urinary bladder	60,490	7%			Uterine corpus	61,380	7%
Melanoma of the skin	52,170	6%			Thyroid	42,470	5%
Kidney & renal pelvis	40,610	5%			Melanoma of the skin	34,940	4%
Non-Hodgkin lymphoma	40,080	5%			Non-Hodgkin lymphoma	32,160	4%
Leukemia	36,290	4%			Leukemia	25,840	3%
Oral cavity & pharynx	35,720	4%			Pancreas	25,700	3%
Liver & intrahepatic bile duct	29,200	3%			Kidney & renal pelvis	23,380	3%
All Sites	**836,150**	**100%**			**All Sites**	**852,630**	**100%**

Estimated Deaths

			Males	Females			
Lung & bronchus	84,590	27%			Lung & bronchus	71,280	25%
Colon & rectum	27,150	9%			Breast	40,610	14%
Prostate	26,730	8%			Colon & rectum	23,110	8%
Pancreas	22,300	7%			Pancreas	20,790	7%
Liver & intrahepatic bile duct	19,610	6%			Ovary	14,080	5%
Leukemia	14,300	4%			Uterine corpus	10,920	4%
Esophagus	12,720	4%			Leukemia	10,200	4%
Urinary bladder	12,240	4%			Liver & intrahepatic bile duct	9,310	3%
Non-Hodgkin lymphoma	11,450	4%			Non-Hodgkin lymphoma	8,690	3%
Brain & other nervous system	9,620	3%			Brain & other nervous system	7,080	3%
All Sites	**318,420**	**100%**			**All Sites**	**282,500**	**100%**

Figura 2: Estadísticas del cáncer sin considerar CCB o CCE [19]

La exploración habitual de los lunares de la piel, es el método más recomendado para detectar los melanomas en las primeras fases de su evolución.

En la actualidad, la detección de estas lesiones se realiza por inspección visual y estudio de las imágenes adquiridas con un dermatoscopio. La inspección manual de los lunares se puede realizar analizando 5 características, método que recibe el nombre de regla ABCDE.

A Asimetría en alguno de sus ejes.

B Bordes irregulares.

C Color no uniforme.

D Diámetro mayor de 6mm

E Evolución. Cambios en forma, color o tamaño.

Se trata de un procedimiento útil, aunque impreciso. Por poner un ejemplo diremos que existen autores, que postulan la conveniencia de suprimir la regla del diámetro, ya que es habitual hallar melanomas inferiores a los 6 e incluso 4 milímetros de diametro [6].

La más importante de las reglas anteriores es probablemente la E de evolución.

La detección de melanomas por inspección visual no resulta trivial y requiere de la pericia de un profesional bien entrenado. Debido a esto y a la gran importancia de un diagnóstico temprano, se suele biopsiar cualquier lesión sospechosa de ser un melanoma.

Podemos comprender la dificultad que implica el diagnóstico de estas lesiones y las molestias que conlleva al paciente, más aún, si pensamos en que para controlar

la evolución de los lunares sospechosos, se hace necesaria la visita continuada al dermatólogo y el control exhaustivo, por parte de este, de los lunares de la piel.

Por todo lo expuesto, una automatización en el proceso de detección y control de la evolución de estas lesiones, conllevaría beneficios evidentes para el paciente y el propio dermatólogo.

3. Estado del arte

La detección automatizada del cáncer de piel, especialmente el melanoma, sigue siendo un importante desafío para muchos investigadores.

Si bien algunos autores reportan datos bastante prometedores, también es cierto que a menudo resulta muy complejo reproducir los resultados que obtuvieron.

Masood y Ali Al-Jumaily [14] proporcionan la única revisión amplia sobre el estado del arte de la detección automática del melanoma, incluyendo 31 trabajos de 1993 a 2012. Proponen establecer un marco general para evaluar los modelos de diagnóstico y exponen la necesidad de observar ciertos criterios de calidad para las distintas fases del proceso de análisis de las imágenes: calibración, preprocesamiento, segmentación, extracción y selección de características, separación de conjunto de entrenamiento y conjunto de validación entre otros.

En la misma línea Fornaciali et al. [4] insisten en la dificultad, con frecuencia imposibilidad, de reproducir la mayoría de los estudios realizados. Aseguran que el potencial del diagnóstico automatizado del melanoma se halla limitado y que esto es debido al énfasis que hace la literatura actual en modelos de visión artificial anticuados. Reimplementan una técnica básica, conocida como bolsa de palabras, a la que podemos referirnos en el ámbito de la visión artificial como bolsa de palabra visuales o «Bag of Visual Words» (BoVW), obteniendo un área bajo la curva (AUC) de 81.2 %. Frente a este modelo implementan dos modelos más, una Red Neuronal profunda y una BoVW avanzada, mejorando ambas el resultado de la técnica básica con una AUC del 84.6 % y 89.3 % respectivamente.

Recientemente se ha publicado un estudio [3] que enfrenta el problema en condiciones más favorables.

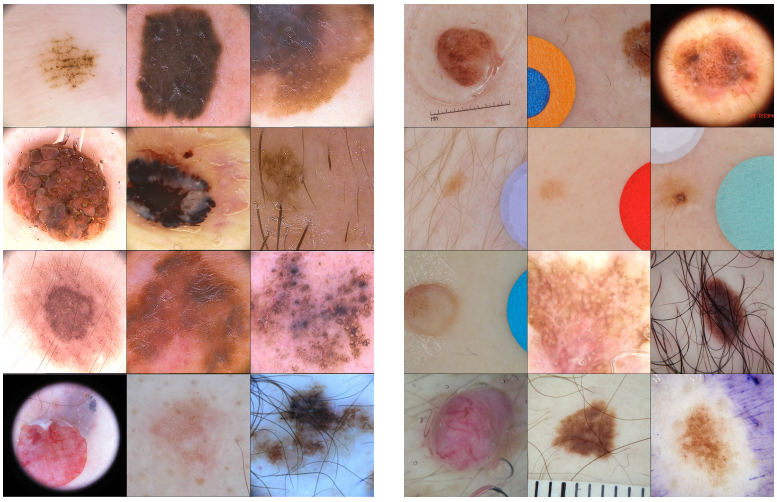
Reuniendo diversas bases de datos han conseguido un total de 129.500 imágenes, lo cual supera otros estudios al menos en dos ordenes de magnitud.

Esta gran base de datos ha permitido establecer una jerarquía entre grupos o categorías de lesiones, CCB, CCE, Melanomas, nevus y queratosis seborreica.

Cabe aclarar que las tres primeras categorías son de tipo maligno y ya las habíamos comentado anteriormente, mientras que nevus y queratosis seborreica son lesiones de tipo benigno.

La abundancia de información, ha permitido a los autores diversificar aún más estas clases generales en subclases y entrenar con ellas una red convolucional.

Para obtener la probabilidad de una clase general, por ejemplo melanoma,



(a) Heterogeneidad (b) Artefactos

Figura 3: Imágenes de ejemplo de lesiones dérmicas. (3a) Diferencias de aspecto. (3b) Algunos de los artefactos presentes.

sumaron las probabilidades de las subclases que pertenecían a esta.

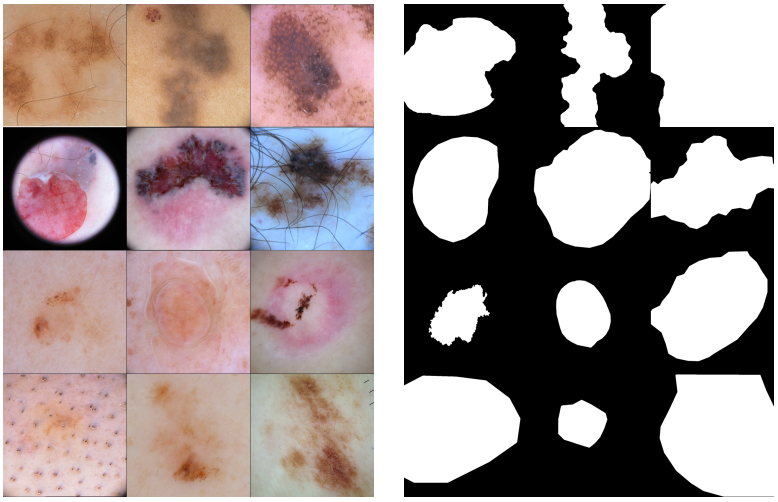
En dichas condiciones, la red convolucional obtiene una exactitud del 72,1±0,9 % , frente a la obtenida por dos dermatólogos en un subconjunto del conjunto de validación, que establecen el — sorprendentemente coincidente — porcentaje del 65,56 *y* 66,0 %.

4. Bases de datos disponibles

La principal limitación es intrínseca al problema mismo, las imágenes de lesiones dermatológicas no tienen una forma, color o textura común, sino que su aspecto difiere de lesión a lesión, véase como ejemplo la figura 3a donde se muestran algunas de las imágenes que se pretenden clasificar.

Dada la magnitud del problema, es muy importante disponer de una base de datos, de suficiente tamaño, con imágenes que delimiten perfectamente la lesión. Pero las bases de datos existentes presentan varios problemas [4] —como podemos apreciar en el ejemplo de la figura 3b —.

La mayoría de los artefactos que se observan en las imágenes, aunque no son los únicos presentes, se enumeran a continuación:



(a) Lesiones dérmicas (b) Segmentaciones manuales

Figura 4: Dificultad en la segmentación manual. Conjunto de imágenes y segmentaciones de una base de datos ¿Que zonas pertenecen realmente a la lesión? ¿Todo el tejido marcado está efectivamente lesionado?

- Marcas de medida o reglas para obtener una idea del tamaño de la lesión.
- Parches de colores para conseguir una referencia del color de la lesión.
- Vendajes, pelo, hebras.
- Burbujas de aceite debidas a la técnica dermatoscópica utilizada.
- Marcos circulares que rodean la zona fotografiada.
- Anotaciones con la fecha.
- Tinciones

Otro problema común, que puede complicar el proceso de aprendizaje automático, es que las lesiones han de ser segmentadas por especialistas. Estos han de determinar la zona de la imagen que efectivamente corresponde a la lesión con la máxima exactitud posible.

La pericia de estos especialistas es un factor determinante para un diagnóstico certero, siendo necesaria una dilatada experiencia para obtener precisiones superiores al 80 % [17].

Consideremos además que el problema que nos ocupa presenta mayor dificultad. No se trata de determinar si una lesión debe ser clasificada o no como melanoma, sino de establecer que zona de la piel corresponde efectivamente a la lesión y evitar marcar como afectadas, aquellas regiones que correspondan a tejido sano. Esto es algo muy difícil, véase la figura 4 con algunas segmentaciones comprometidas.

	Entrenamiento						**T**est	
	Dermatológico		**L**unares	**M**ix			**D**erm.	**L**unares
Código	**ED/ED+**		**EL/EL+**	**EM**			**TD**	**TL**
Procedencia	pH2	ISBI	UB	PH2	ISBI	UB	ISBI	UB
Número	200	350	279	200	350	279	49	49
Porcentaje (%)	36.4	63.6	100	24.1	30.2	33.7	100	100
Total	550 ED		279 EL	829			49	49
Aumentado	4400 ED+		2232 EL+	No			No	No

Tabla 1: Resumen de los conjuntos de imágenes

Por todo ello los investigadores suelen hacer un cribaje de las bases de datos, eliminando aquellas imágenes que, por un motivo u otro, no son adecuadas o pueden distorsionar los resultados que genera el método de análisis.

Esto provoca que los resultados obtenidos dependan fuertemente de la elección de las imágenes de estudio.

Así pues, se trata de que podamos reproducir adecuadamente la metodología empleada por los distintos autores, que tengamos acceso al código que se utilizó y que la base de datos empleada esté claramente especificada y accesible. No basta con saber que base de datos se usó, es necesario conocer que imágenes, de esas bases de datos, fueron las que participaron en la evaluación del método desarrollado.

Aunque el cribaje de imágenes que difieren mucho del resto es una opción frecuente, la solución más razonable pasa por incrementar el número de imágenes de estudio, pudiéndose de este modo realizar un aprendizaje mucho más robusto. Esto no resulta sencillo, ya que la mayoría de las bases de datos de imágenes dermatológicas, pertenecen a instituciones privadas u hospitales y no son libremente accesibles.

Algunas de las que hemos podido obtener presentan dificultades añadidas. Al no existir un método de recopilación estandarizado, las lesiones se han clasificado de diferentes modos: atendiendo a su malignidad, la clase de lesión a la que pertenecen (CCB, CCE, melanoma, nevus, queratitis y otras muchas), la presencia o ausencia de estudio histológico de biopsia de la lesión, su asimetría categorizada en niveles (1, 2, 3), la presencia de velo azul-blanquecino, el color y otros parámetros.

En nuestro estudio exploratorio hemos dividido nuestros datos en dos categorías principales: Imágenes dermatológicas y lunares.

- Imágenes dermatológicas (ED): Empleamos la base de datos «$PH2$ dataset» [16] disponible públicamente de modo gratuito, aunque compuesta tan solo por 200 imágenes. A esta base de datos le añadimos algunas de las imágenes que se ofrecían en el «challenge» ISBI 2016, provenientes de la base de datos del «International Skin Image Collaboration» (ISIC), que cumpliesen con las condiciones de disponer de segmentación y tener un diagnóstico confirmado por histología del tejido biopsiado. Reuniendo ambas bases de datos obtenemos 550 imágenes de estudio, etiquetamos esta base de datos con el código (ED).

- Lunares (EL): Las imágenes de acumulaciones melanocíticas estudiadas, se obtuvieron recortando manualmente los lunares de una secuencia de fotografías. Dicha secuencia, se capturó con un teléfono móvil y corresponde a un brazo en diferentes posiciones y condiciones de iluminación. Tanto la zona rectangular que incluye la lesión, como la segmentación de esta, se obtuvieron de modo manual. Disponemos de un total de 279 imágenes.

Nuestra base de datos, aún siendo muy limitada, puede ayudarnos a explorar el uso de redes completamente convolucionales o «fully convolutional networks» (FCN) en la segmentación semántica de las regiones de interés (ROI) .

Para comprobar el desempeño de las redes generamos dos conjuntos de imágenes de test:

- Test dermatológico (TD): 49 imágenes dermatológicas provenientes de la base de datos de ISIC independientes del conjunto de entrenamiento.

- Test lunares (TL). 49 imágenes de lunares generadas en la Universitat de Barcelona (UB) que no forman parte de la base de datos de lunares.

Para el desarrollo experimental se utilizaron diferentes combinaciones de los conjuntos de entrenamiento que se hallan resumidas en la tabla 1. A los conjuntos marcados con el símbolo «+» se les aplicó un procedimiento de aumento de datos, según se establece en el subapartado 7.4.

5. Aprendizaje automático, Redes neuronales y análisis de imágenes.

Las diferentes técnicas de aprendizaje automático, han permitido obtener mejoras en algunas tareas fundamentales para el tratamiento de imágenes, como son: La clasificación, detección, localización o segmentación de regiones de interés.

Haremos una pequeña introducción a los conceptos básicos del aprendizaje automático y las redes neuronales. La mayoría de los conceptos que comentaremos merecen una explicación mucho más exhaustiva que la que podemos incluir en los límites de este trabajo. Se recomienda al lector interesado la consulta del excelente material disponible a través de Internet [1][7].

5.1. El perceptrón.

Las redes neuronales, inspiradas inicialmente en el funcionamiento del cerebro, son modelos matemáticos compuestos por nodos (neuronas) interconectados entre si y organizados habitualmente en capas.

El perceptrón, desarrollado por Frank Rosenblatt a partir de un trabajo anterior de Warren McCulloch y Walter Pitts [15], es uno de los modelos más sencillos de neurona artificial.

En la figura 5 podemos ver un ejemplo del funcionamiento de estas neuronas con solo tres entradas. Cada entrada está asociada a un factor multiplicativo llamado peso, representado en el esquema por w_1, w_2 y w_3. A la suma ponderada de las entradas se les aplica una función matemática conocida como función de activación. Si el valor resultante supera un umbral preestablecido, la salida sera 1 y en caso contrario será 0.

Si definimos un sesgo equivalente al valor umbral cambiado de signo y lo denotamos por el valor b podremos expresar este proceso con la ecuación 5.1 .

$$Salida = \begin{cases} 0 & si\ f(\sum x \cdot w) - b \leq 0 \\ 1 & si\ f(\sum x \cdot w) - b > 1 \end{cases} \tag{5.1}$$

Este tipo de neuronas resulta útil para modelar problemas linealmente separables, por ejemplo puertas lógica AND, OR o NOT. Si necesitamos modelar características más complejas ,como una puerta XNOR o cualquier otro problema no linealmente separable, como el caso de la figura 6 , tendremos que combinar varios perceptrones. Hablamos entonces de perceptrones multicapa.

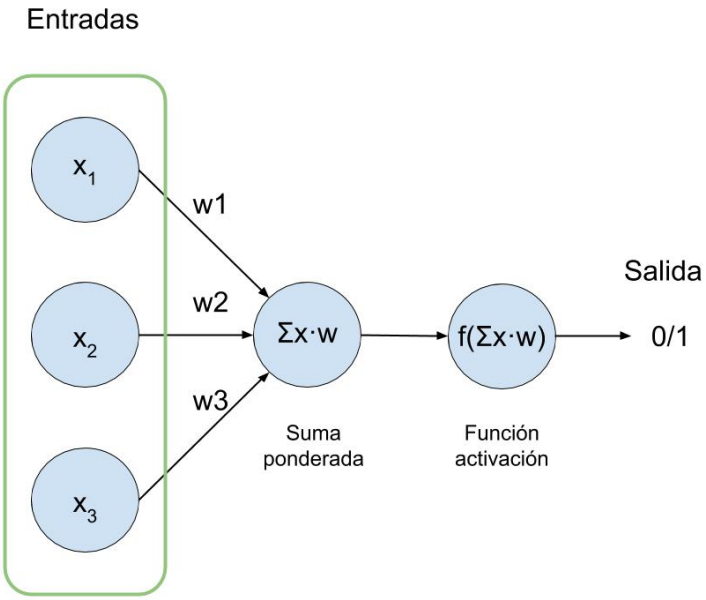


Figura 5: Neurona artificial o perceptrón.

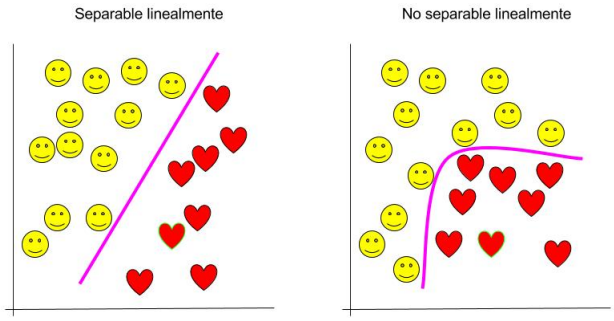


Figura 6: Ejemplo linealmente separable y linealmente no separable.

5.2. Funciones de activación

En el ejemplo del perceptrón de la figura 5, se calculaba la suma ponderada de las entradas y, al valor así obtenido, se le aplicaba la función de la ecuación 5.1. Esta función de activación, que recibe el nombre de función escalón o de Heaviside, es útil en el caso del perceptrón para problemas linealmente separables.

Para modelos más complejos, que deban resolver problemas no linealmente separables, no resulta adecuada. Esto es así porque en el proceso de entrenamiento de la red se utiliza un procedimiento llamado retropropagación, que precisa de una función de activación diferenciable. Para actualizar los pesos de la red, la retropropagación emplea un algoritmo conocido como descenso de gradiente, que usa la derivada de la función de activación. Como la función escalón no es diferenciable para $x = 0$ y además su derivada es 0 para cualquier otro valor, el algoritmo de retropropagación no funcionará.

Otras funciones de activación que no adolecen de este problema, por ejemplo la función sigmoide, presentan otra serie de inconvenientes, algunos de los cuales comentaremos más adelante.

La figura 7 muestra algunas de las funciones de activación más usadas. Existen otras como Leaky ReLU, maxout y softmax cuya elección dependerá de nuestro problema.

5.2.1. Función Sigmoide

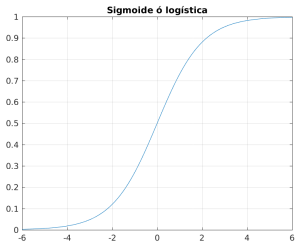
La función sigmoide es una función no lineal que genera un valor de salida entre 0 y 1. Los números próximos al extremo negativo tienden a cero cuanto menores son mientras que los próximos al extremo positivo tienden a 1 cuanto mayor es su valor.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5.2}$$

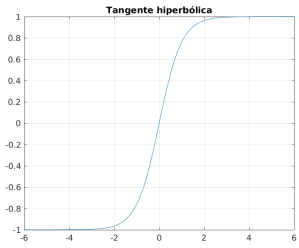
La función sigmoide presenta algunos inconvenientes:

- Al aproximarse a cero o a uno el valor de salida de la función, la pendiente en estas zonas es casi cero. En el cálculo de la retropropagación, necesario para el proceso de aprendizaje, se multiplica este valor para el cálculo del gradiente de descenso, lo cual lo hace inadecuado, pues casi anula el gradiente.
- La salida media de esta función no es cero sino 0.5. Esto también tiene efectos no deseables en el cálculo del gradiente de descenso.

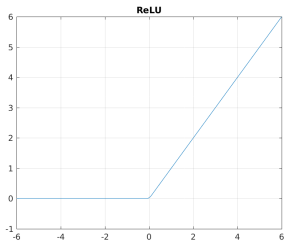
Estos inconvenientes han hecho que en la actualidad esta función se use poco.



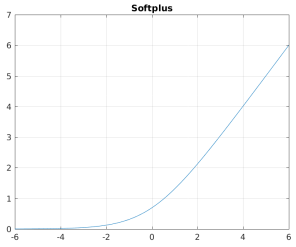
(a) Sigmoide



(b) Tangente Hiperbólica



(c) ReLU



(d) Softplus

Figura 7: Funciones de activación

5.2.2. Función de tangente hiperbólica (tanh)

La función tanh es el cociente entre el seno y el coseno hiperbólico de un valor. Es una función no lineal que genera un valor de salida entre -1 y 1.

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5.3}$$

Presenta el mismo inconveniente que vimos en la función sigmoide, la pendiente en los extremos es casi cero. Pero a diferencia de ella, sus valores de salida si que están centrados en torno al valor cero y es por ello por lo que se suele preferir esta función a la sigmoide.

5.2.3. ReLU

La función ReLU genera una salida de cero para cualquier valor $x < 0$ y una salida lineal con pendiente uno para el resto de valores de entrada.

$$f(x) = max(0, x) \tag{5.4}$$

Esta función, acelera la convergencia en el cálculo del gradiente de descenso [13] respecto a la sigmoide o tanh. Además, su cálculo es relativamente sencillo, pues no se necesitan exponenciales, lo cual acelera el cómputo. A pesar de estas ventajas también presenta algún inconveniente, como el hecho de que en ocasiones, los pesos de algunas neuronas tomen valores que hagan que estas neuronas no se activen

nunca. Se trata además, de una función no diferenciable cuya derivada es cero en la parte negativa.

Se han propuesto alternativas basadas en este modelo como las *noisy-ReLUs*, *PReLUs*, *leaky ReLUs* o *ELUs* que no entraremos a analizar en este trabajo.

5.2.4. Softplus

La función *softplus* es una aproximación a las *ReLUs*.

$$f(x) = \ln(1 + e^x) \tag{5.5}$$

Su derivada es la función logística.

$$f\prime(x) = \frac{1}{1 + e^{-x}} \tag{5.6}$$

Si bien esto soluciona el problema de las *ReLUs*, consiguiendo que la función sea diferenciable y con derivada positiva en todo el rango de valores, también elimina su principal ventaja, que es agilizar el computo, puesto que nuevamente precisamos calcular exponenciales.

5.2.5. Maxout

Se han propuesto otras soluciones que aplican la no linealidad al producto de los valores de entrada y los pesos. Un ejemplo seria *maxout* [8], que es una versión general de las *ReLUs*.

5.2.6. Softmax

La función *softmax* es una generalización de la función logística, toma como entrada un vector x n-dimensional y ofrece como salida otro vector y n-dimensional, cuyos valores se hallan comprendidos entre 0 y 1.

5.3. Redes neuronales, funcionamiento

En una red neuronal el diseño habitual se hace por capas, denotándose la primera como de entrada, la última como de salida y las intermedias como capas ocultas. La salida de cada una de las neuronas de una capa, esta interconectada a la entrada de todas las de la siguiente y ponderadas por un peso, tal como vimos en el diseño del perceptrón de la figura 5.

En la figura 8 se muestra un posible diseño de red neuronal. Podemos ver que tenemos una capa de entrada, dos capas ocultas y una capa se salida con dos posibles clases.

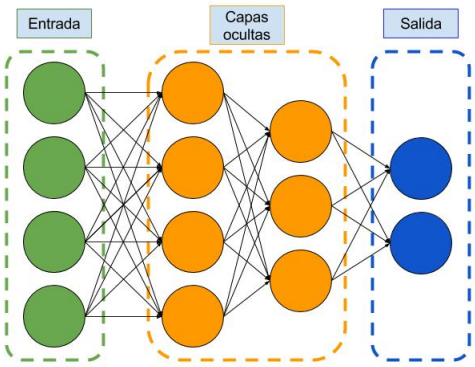


Figura 8: Ejemplo de red neuronal

5.3.1. Aprendizaje

Las redes neuronales se entrenan mediante aprendizaje supervisado, esto quiere decir que conocemos, de antemano, el resultado que debe producir la red para un conjunto de datos de entrada, llamado conjunto de entrenamiento.

Si inicializamos la red con pesos y sesgos aleatorios y suministramos los datos de entrada del conjunto de entrenamiento, obtendremos una hipótesis por parte de la red. Dado que conocemos el valor real que queremos que la red produzca, podemos calcular el error cometido mediante algún algoritmo y modificar esos pesos, de modo que en la siguiente iteración el error cometido sea menor.

5.3.2. Función de coste

Una manera sencilla sería calcular y minimizar el error cuadrático medio.

$$J(x) = \frac{1}{2}(h(x) - y)^2 \tag{5.7}$$

Donde y corresponde el valor esperado y $h(x)$ a la hipótesis generada por la red. Esta ecuación recibe varios nombres, error cuadrático medio, función de coste o función objetivo. Una vez tenemos un modo de calcular el error, o coste, necesitaremos un algoritmo que permita minimizarlo.

5.3.3. Descenso por gradiente

El descenso por gradiente es un algoritmo, sirve para minimizar funciones que pueden depender de muchos parámetros. Para simplificar el proceso y obtener una intuición de como funciona el método, imaginemos que simplemente tenemos que ajustar dos pesos a los que denominaremos θ_0 y θ_1. Inicializaremos los pesos de manera aleatoria y seguidamente ejecutaremos el siguiente algoritmo hasta que converja al coste que queremos:

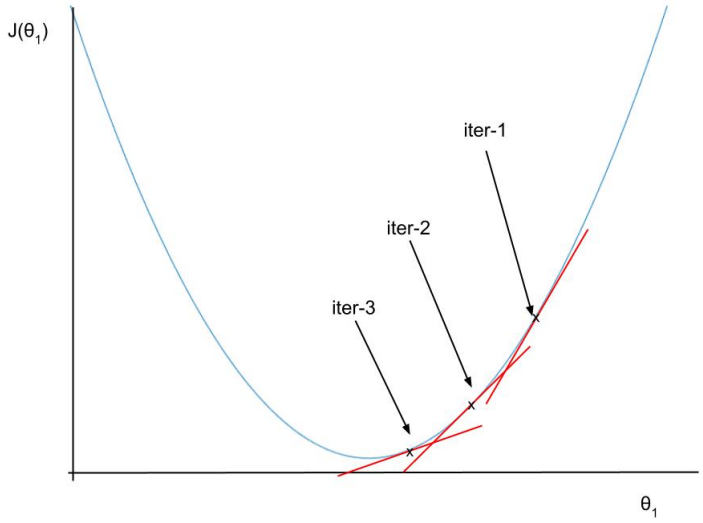


Figura 9: Coste $j(\theta_1)$ frente a valor de θ_1

$$\theta_{j1} \leftarrow \theta_{j0} - \alpha \frac{\partial}{\partial \theta_j} j(\theta_{0,}\theta_1) \tag{5.8}$$

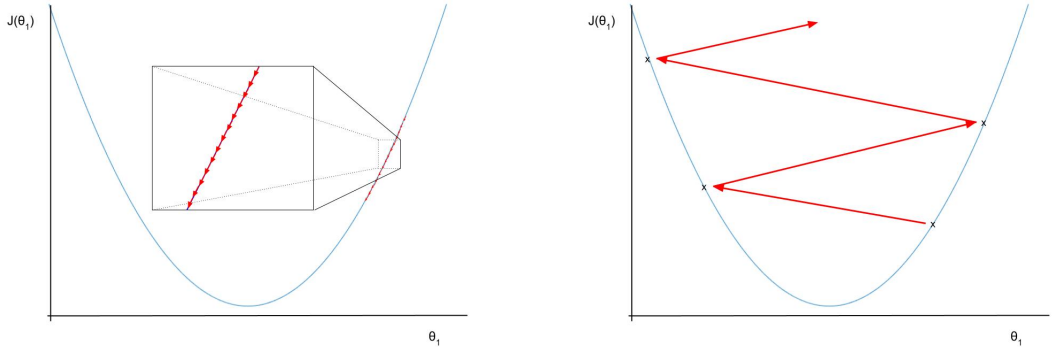
Donde $\frac{\partial}{\partial \theta_j} j(\theta_{0,}\theta_1)$ es la derivada parcial de la función de coste para θ_0 y θ_1, α es el factor de aprendizaje [ver: 5.3.4], θ_{j0} es el valor actual de los pesos y θ_{j1} el nuevo valor calculado .

Al calcular y restar el gradiente, lo que hacemos es ir en la dirección en la que este decrece. Si hacemos un gráfico del valor del peso frente al coste, obtenemos algo como lo que se observa en la figura 9

Cada iteración nos acercará un poco más al valor mínimo del coste, aunque debemos recordar que este mínimo puede ser local y que el factor de aprendizaje α debe estar bien escogido.

5.3.4. Factor de aprendizaje α

α es un factor que multiplica el valor resultante de la derivada parcial. A mayor valor del factor de aprendizaje, mayor será el "salto" que daremos entre una iteración y otra para calcular el nuevo valor de $J(x)$. Un valor demasiado bajo puede hacer que el gradiente de descenso converja demasiado lentamente. Un valor excesivamente alto puede hacer que no encontremos el mínimo, que no halla convergencia o incluso que diverja. Este comportamiento se ilustra en la figura 10



(a) α pequeña, la convergencia es lenta.

(b) α grande, és posible la divergencia.

Figura 10: Ejemplos de α inadecuada

5.3.5. Propagación del error - retropropagación

Hemos visto que para obtener una hipótesis por parte de la red neuronal, debemos suministrarle unos valores de entrada, que se propagaran a través de ella hasta producir una salida. Este proceso se conoce como propagación hacia adelante.

También hemos comentado que para el proceso de aprendizaje, es necesario calcular el error cometido y que este se puede calcular a partir de la diferencia entre el valor real y el predicho. Este error es el cometido por la red en la propagación hacia adelante de los valores de entrada, necesitamos conocer el error que comete cada neurona y su contribución a este error global para poder actualizar los pesos de la red adecuadamente.

La propagación hacia adelante, no es más que una composición de funciones, en la que al valor de entrada se le aplica sucesivamente la función de activación. Si combinamos esta idea con el calculo del descenso de gradiente, donde calculábamos la derivada parcial del coste y hacemos uso de la regla de la cadena, es posible retropropagar el error a las neuronas de la capa anterior.

5.4. Construcción de redes convolucionales y Análisis de imágenes

Según hemos visto hasta ahora, las redes neuronales se componen de una capa de entrada, una de salida y un número variable de capas intermedias llamadas capas ocultas. Las variables de entrada se ponderan mediante un factor multiplicativo llamado peso y se transfieren a todas las neuronas de la primera capa, cada neurona realiza la suma ponderada de sus entradas y aplica una función de activación, produciendo un valor de salida que transfiere a todas las neuronas de la siguiente capa.

En el análisis de imágenes mediante redes neuronales, debemos considerar algunas características inherentes al problema. Tomemos como ejemplo una pequeña

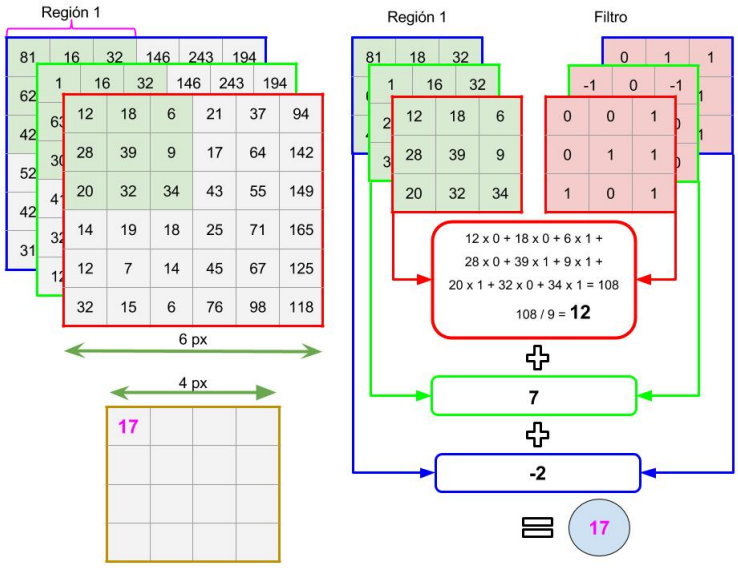


Figura 11: Ejemplo de convolución

imagen RGB de tamaño 256×256. Como cada uno de los píxeles es, en si, una variable de entrada a la red, obtendremos un total de $256 \times 256 \times 3$ píxeles o, lo que es lo mismo, $\sim$ 200,000 variables. Esto significa que cada una de las neuronas de la primera capa oculta tendría asociados $\sim$ 200,000 pesos. Se da un problema añadido debido a que un alto número de variables de entrada suele producir sobreajuste de la red.

Una solución mucho más adecuada para el análisis de imágenes mediante redes neuronales, consiste en el uso de redes neuronales convolucionales. Estas redes, se construyen apilando una serie de capas, que van transformando la imagen de entrada, extrayendo características y modificando la dimensionalidad, hasta obtener una salida con las puntuaciones de cada una de las clases que hemos etiquetado y suministrado a la red como conjunto de entrenamiento.

5.4.1. Capa convolucional - CONV

Estas capas permiten convolucionar nuestra imagen con un filtro, cuyos pesos podemos inicializar aleatoriamente. Estos pesos serán los que se actualicen durante el proceso de aprendizaje.

En la figura 11 se ilustra la operación de convolución entre una imagen de tres canales de 6x6 píxeles y un filtro de dimensiones 3x3x3.

La operación de convolución sigue el mismo principio que vimos para el caso de los perceptrones. Cada neurona de la primera capa ejecutará un producto entre el valor de entrada (píxel) y un peso asociado (cada valor del filtro), sumará estos valores y posteriormente aplicará la función de activación. La diferencia más evidente es que, ahora cada neurona de la primera capa convolucional estará asociada solamente a una región de la imagen de entrada, esta región recibe el nombre de campo receptivo.

En nuestro ejemplo de la figura 11 aplicamos una convolución a nuestra imagen con un solo filtro, también llamado kernel. Lo que obtenemos tras este proceso es lo que se conoce como mapa de características, las regiones en que el filtro coincide mejor con la zona de la imagen convolucionada tendrán una respuesta mayor. El tamaño del kernel es relevante, puesto que dependiendo de el se podrán extraer características de mayor o menor tamaño.

5.4.2. Salto - Stride

La convolución se realiza en toda la imagen desplazando el filtro por ella. Podemos desplazar el filtro un solo píxel o hacer saltos de 2 o más píxeles. Este valor se conoce con el nombre de salto o stride. Cuanto mayor sea el stride menor será el volumen de salida.

5.4.3. Profundidad - Depth

En nuestro ejemplo de la figura 11 convolucionabamos nuestra imagen con un solo filtro, también llamado kernel. Nada nos impide apilar varios filtros y convolucionarlos con nuestra imagen, al número de filtros apilados se le da el nombre de profundidad o depth.

5.4.4. Relleno de ceros - zero padding

Vimos en nuestro ejemplo de la figura 11 que, al convolucionar una imagen con un filtro, las dimensiones de altura y anchura de la salida resultante, eran menores que las de la imagen original. Este efecto de la convolución puede corregirse — si estamos interesados en recuperar la información del borde de la imagen — sobredimensionando dicha imagen con un marco relleno de ceros, de modo que las dimensiones de altura y anchura resulten ser las de la imagen inicial.

5.4.5. Época

Se define como época, cada una de las etapas del proceso iterativo de aprendizaje en que todas las imágenes de entrenamiento han sido propagadas por la red. Emplear aquí el término «iteraciones» — sin más explicación — puede llevar a confusión. Consideremos que el conjunto de entrenamiento se suele dividir en lotes, que se propagan a través de la red en cada iteración. Así pues, una época corresponde a todas las iteraciones necesarias para hace pasar todos los lotes, que componen el conjunto de entrenamiento, a través de la red.

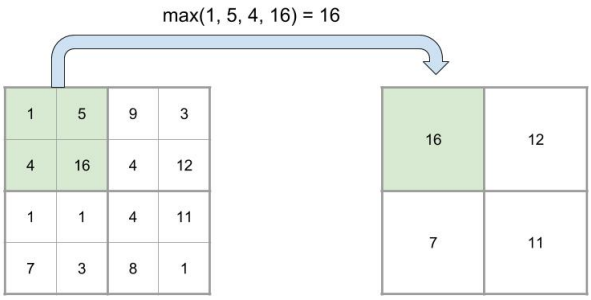


Figura 12: Agrupación por máximo, max-pool

5.4.6. Capa ReLU

Con el uso de la capa convolucional, hemos realizado la primera parte del cálculo que nos interesaba, es decir, la suma ponderada del producto de entradas y pesos. Pero aún necesitamos aplicar nuestra función de activación no lineal.

Un modo bastante efectivo y computacionalmente poco costoso de hacerlo, es colocar una capa ReLU después de cada capa convolucional, esta capa, que aplicará la función que vimos en la figura 7c, convertirá todos los valores negativos en cero.

Podríamos utilizar alguna de las funciones de activación que vimos en la figura 7 , como *tanh* o *sigmoide* pero ReLU da buenos resultados y resulta muy sencilla de calcular.

5.4.7. Capa de agrupación - Pool layer

Tras la capa convolucional y la ReLU se acostumbra a usar un proceso llamado *pooling*. Este proceso, que reduce la dimensionalidad del mapa de características, retiene la mayor parte de la información relevante.

La capa de agrupación, subdivide el mapa de características en regiones, sobre las que aplica una operación matemática de la que se obtiene un único valor. Las operaciones más habituales son la suma, la media y el máximo. Siendo la del máximo la opción más popular y que se ilustra en la figura 12

El proceso de agrupación reporta varios beneficios en el proceso:

- Reduce el numero de parámetros, lo cual permite agilizar el cómputo.
- Consigue que la red sea invariante ante pequeñas transformaciones de traslación o rotación, debido a que tomamos el máximo de una región.
- También al reducir el número de parámetros disminuye la posibilidad de sobreajuste.

5.4.8. Capa de descarte - Drop layer

El proceso de descarte o «dropout»[9][22] es una técnica de regularización. Es utilizada frecuentemente en el entrenamiento de redes neuronales. Permite reducir la probabilidad de sobreajuste, impidiendo que un porcentaje de las neuronas de una capa se activen en un ciclo del proceso de aprendizaje. Como este procedimiento se efectúa aleatoriamente en cada ciclo, las neuronas desactivadas no serán siempre las mismas. Se consigue así que la red aprenda patrones de un modo más general y no dependa, rígidamente, de la activación de determinadas neuronas.

Esta técnica, que resulta útil para redes completamente conectadas, no es muy usada en las redes convolucionales. Su elección depende del problema al que nos enfrentemos y en numerosas situaciones la escasa mejora obtenida no justifica su empleo.

5.4.9. Normalización por lotes - Batch normalization

La normalización por lotes [10] es otra técnica de regularización ampliamente empleada. Durante el proceso de aprendizaje, la media y varianza de los valores de entrada de una capa varían, precisamente porque dependen de los parámetros aprendidos de las capas previas. Este efecto, que los autores del procedimiento de normalización por lotes llaman «covariate shift», ralentiza el proceso de aprendizaje, hace que los parámetros de inicialización deban ser cuidadosamente elegidos y dificulta el buen funcionamiento del gradiente de descenso.

Para evitar estos efectos indeseables, el algoritmo consigue dotar a todas las capas de la red de unas entradas de media cero y varianza uno. Esto, no solo mejora los efectos citados sino que además, logra evitar el sobreajuste, haciendo prácticamente innecesario el uso de capas de descarte.

5.4.10. Capa totalmente conectada - Fully connected layer (FC)

Las capas totalmente conectadas, se corresponden con lo que vimos en el caso de los perceptrones multicapa y con el ejemplo de la figura 8. Cada una de las neuronas de esta capa, está conectada a todas las salidas de la capa anterior. Estas capas, que suelen situarse al final de una red neuronal convolucional, permiten el cálculo de las puntuaciones obtenidas para cada una de las clases del problema original.

5.4.11. Conversión entre capas totalmente conectadas y capas convolucionales

La única diferencia entre una capa convolucional y una totalmente conectada, reside en las conexiones de las neuronas. Mientras que en las CONV las conexiones se hacen con una región de la entrada, en las FC se hace con toda ella.

Por lo demás su función es idéntica, puesto que calculan el producto de pesos y valores de entrada del mismo modo.

Es posible convertir una capa FC en una capa CONV y viceversa.

Para convertir una capa CONV en una FC, bastará con implementar una matriz de pesos de gran tamaño, que en su mayoría estará compuesta por ceros, para reproducir la conectividad de la capa convolucional. Conseguiremos así que apliquen la misma función de propagación hacia adelante de los valores de entrada.

La conversión de una capa FC en una capa CONV requiere un proceso algo más complejo. Como ejemplo diremos que una capa FC de 4096 pesos que tenga una entrada de $7 \times 7 \times 512$ se puede expresar del mismo modo mediante una capa CONV con 4096 filtros de dimensiones 7×7, estableciendo un salto o stride de 1 y sin rellenar con ceros [12].

5.5. Redes completamente convolucionales - FCN

Las redes que son totalmente convolucionales, es decir aquellas en las que no hay capas FC, presentan algunas ventajas. Podemos tratar la red entera como un enorme filtro convolucional, lo cual permite aplicar toda la red en una única propagación hacia adelante, que es computacionalmente mucho más eficiente. Además, permiten tomar como entrada una imagen de medida arbitraria y producir una salida de la medida adecuada. Esta técnica ha sido aplicada con mucho éxito en la segmentación semántica de imágenes [18].

5.6. Transferencia de aprendizaje - Métodos

Entrenar adecuadamente una red convolucional implica utilizar un enorme número de imágenes, por ejemplo «Imagenet» es una base de datos que dispone de más de un millón de imágenes de mil categorías diferentes.

No es sencillo obtener una base de datos semejante que sea adecuada a nuestro problema.

Existen métodos que permiten aprovechar el aprendizaje de otras redes, entrenadas con grandes bases de datos, transfiriendo este aprendizaje de modo que podamos finalizar el entrenamiento con nuestro conjunto de datos.

5.6.1. Red convolucional como extractor de características

El primero de los métodos de transferencia de aprendizaje que comentaremos, consiste en utilizar una red pre-entrenada y eliminar la última capa completamente conectada. Trataremos entonces la red resultante como un extractor de características.

Una vez extraídas las características entrenaremos un clasificador lineal, como «SoftMax» o una «Support Vector Machine» con nuestro conjunto de datos.

5.6.2. Ajuste fino de la red - Fine tunning

El segundo procedimiento utiliza la red pre-entrenada y efectúa el ajuste fino de los pesos entrenando con nuestro conjunto de datos.

Se pueden entrenar todas las capas de la red o congelar las primeras y entrenar solamente las últimas.

Esta posibilidad es debida a que las redes convolucionales aprenden en sus primeras capas las características más gruesas, como forma o bordes y que suelen ser generalizables a cualquier conjunto de imágenes.

Pueden ser así aprovechadas en el entrenamiento de nuestro conjunto de datos sin necesidad de modificaciones. De este modo podremos centrar nuestro aprendizaje en las características más finas, como pequeños cambios de textura o color, específicas de nuestras imágenes.

Se pueden usar también técnicas mixtas, concediendo un peso distinto a cada capa, mayor cuanto más se acerque la capa al final de la red.

La elección de la técnica adecuada depende de cada problema. Se suele decidir en función del tamaño de nuestra base de datos y de la similitud entre nuestras clases y aquellas con las que se entreno la red original. Las opciones más recomendadas son las siguientes:

- **Pequeña base de datos, conjunto de datos similar al original**: Normalmente da mejores resultados utilizar la red como extractor de características, aplicando un clasificador lineal que entrenaremos con nuestros datos.
- **Base de datos grande, datos similares a los originales:** Usar el ajuste fino de los pesos de toda la red.
- **Pequeña base de datos muy diferente de los datos originales:** Como en el primer caso, es mejor entrenar un clasificador lineal y usar la red como extractor de características. Como nuestros datos son muy diferentes, en lugar de usar toda la red, seria preferible utilizar las primeras capas como extractor de características y entrenar las últimas con nuestros datos.
- **Base de datos grande y datos muy distintos a los originales:** Aunque podemos plantearnos entrenar toda la red desde cero, resulta útil inicializar la red con los pesos de una red pre-entrenada y continuar el entrenamiento con nuestros datos.

La práctica más habitual es truncar la última capa (softmax) de la red pre-entrenada y sustituirla con una capa softmax, adecuada al número de clases de nuestro problema. Inicializamos a cero los pesos de esta última capa y hacemos el ajuste fino de la red con nuestros datos.

Es también recomendable usar un factor de aprendizaje α pequeño. Utilizar un α demasiado grande podría distorsionar los pesos de partida demasiado rápido, perdiendo los beneficios obtenidos con el pre-entrenamiento.

Otra técnica habitual consiste en congelar los pesos de las primeras capas para no alterar las características generales aprendidas por la red original.

6. Condiciones experimentales

6.1. Framework

Para realizar nuestras pruebas hemos utilizado el software VLFeat - MatConvNet [24].

Difícilmente explicaremos lo que es MatConvNet mejor que sus autores. Por ello mostramos una traducción propia del abstract que acompaña al manual de este software.

> «MatConvNet es una implementación de redes neuronales convolucionales (CNNs) para MATLAB.
>
> »La herramienta está diseñada enfatizando la simplicidad y flexibilidad. Ofrece las piezas de construcción de las CNNs como sencillas funciones de MATLAB, proveyendo rutinas para el cálculo de convoluciones lineales con bancos de filtros, agrupación de características y mucho más.
>
> »De este modo, MatConvNet permite el rápido prototipado de las arquitecturas de las nuevas CNNs.
>
> »Al mismo tiempo, soporta una computación eficiente sobre CPU y GPU, permitiendo el entrenamiento de modelos complejos con grandes conjuntos de datos, como los de Imagenet Large Scale Visual Recognition Challenge (ILSVRC).»

6.2. Modelo de red

La red VGGnet, desarrollada por karen Simonyan y Andrew Zisserman para el ILSVRC de 2014 [20], demostró que la profundidad de una red convolucional influye en su exactitud.

Esta red, que mejoraba significativamente los modelos usados hasta ese momento, se distribuyó en dos versiones, de 16 y 19 capas CONV/FC, compatibles con el framework de *Caffe* [11].

Cuenta con una arquitectura muy homogénea, que solo realiza convoluciones de 3×3 con un $Salto = 1$ y agrupaciones por máximo de 2×2.

Su principal inconveniente es que la evaluación es más costosa y que emplea gran cantidad de memoria y parámetros.

Su arquitectura, excluyendo capas de agrupación, entrada y clasificador softmax final, esta compuesta por 13 capas CONV y 3 FC en el modelo de 16 capas y de 16 CONV y 3 FC en el modelo de 19 capas.

Una red de este tipo puede mejorar en mucho su rendimiento si la convertimos en una red FCN, como demostraron J. Long, E. Shellhamer y T. Darrell [18] evaluando VGGnet [20], Alexnet [13] y GoogLeNet [23]. Las redes FCN tienen además la ventaja de que permiten utilizar entradas de medida arbitraria y producir la salida correspondiente. Estas redes resultan especialmente adecuadas para tareas de predicción espacialmente densa, como la segmentación semántica.

6.2.1. Segmentación semántica

A diferencia de la clasificación o localización de objetos, la segmentación semántica etiqueta cada imagen a nivel de píxel, por ello el resultado no será una «bounding box» rodeando el objeto perteneciente a cada clase, sino que tendrá una forma completamente arbitraria compuesta por los píxeles pertenecientes a cada clase.

6.2.2. MatConvNet - FCN

A partir del modelo comentado anteriormente de FCN para la segmentación semántica [18], S. Ehrhardt y A. Vedaldi desarrollan una versión en MatConvNet disponible en GitHub. Este es el modelo que — convenientemente modificado — emplearemos en nuestro trabajo.

Aunque el modelo se desarrolló para la segmentación de 21 clases (20 más el fondo), hemos adaptado el código para reducirlo a un problema de clasificación binaria.

La adaptación del modelo ha resultado bastante compleja. El modelo de partida no forma parte propiamente del «toolbox» VLFeat-MatConvNet, sino que es un ejemplo de utilización de este.

Por este motivo, el código no está bien documentado y se precisa de conocimientos avanzados y una gran experiencia en técnicas de aprendizaje profundo para comprender las operaciones y algoritmos utilizados en este paquete de software.

Tampoco ayuda el hecho de que el código presente un número indeterminado de errores, algunos evidentes y otros no tanto, que impiden su correcto funcionamiento. Recomendaríamos a quien desee introducirse en el uso de este software, que empezase implementando ejemplos sencillos de redes neuronales con las herramientas que ofrece MatConvNet y que, solo después de haber adquirido experiencia en su uso y contando con una formación sólida en aprendizaje profundo, intentase explorar el ejemplo de FCN si desea adaptarlo a sus necesidades.

6.3. Adaptación del ejemplo de FCN

El proceso de creación y comprobación de los modelos ha sido complejo debido a múltiples causas, expondremos algunas que creemos merecen ser comentadas.

La ejecución del modelo de FCN-MatConvNet en MatLab instalado sobre los dos sistemas operativos que hemos utilizado, Windows 10 y Ubuntu 16.04 requiere ser especialmente cuidadoso. Hay que tener en cuenta algunas consideraciones al intentar ejecutar dicho modelo y en consecuencia nuestra propia versión del mismo :

- La carga de ficheros, dataset, modelos y otros se debe hacer respetando las diferencias propias de cada sistema operativo, concretamente las rutas no se escriben igual en Windows que en Linux. Esto puede solventarse sustituyendo la barra «/» por «//» allí donde se precise o — de modo mucho más simple — empleando la función «fullfile» de MatLab.

- Debemos tener en cuenta el compilador de «C» que necesitamos para compilar MatConvNet en Matlab. En el caso de Linux, esto pasa por instalar y utilizar, solo para la compilación de VLFeat-MatConvNet, un compilador compatible con nuestra versión de MatLab ya que el «GCC», disponible en nuestro sistema, es mucho más reciente y no está soportado por la aplicación.

- Dependiendo de la tarjeta que utilicemos podemos tener dificultades añadidas en el empleo de CUDA. Las tarjetas basadas en Pascal pueden trabajar con CUDA 7.5, pero no están completamente soportadas, esta combinación suele ser fuente de errores. CUDA 8.0 si que soporta totalmente las tarjetas pascal, el problema es que las versiones de MatLab anteriores a R2017A eran incompatibles con CUDA 8.0, admitiendo como mucho CUDA 7.5 .

- Otra fuente de problemas habitual es el uso de la función VL_IMREADJPG de VLFeat. Esta función no suele dar problemas en entornos Windows, pero en Linux no funciona adecuadamente para imágenes que no sean JPG. Esto puede solucionarse, en tiempo de ejecución, consultando el sistema operativo usado y el tipo de imagen a leer con los comandos pertinentes de MatLab. Las imágenes que dan error se pueden leer con el comando de MatLab IMREAD en lugar de utilizar VL_IMREADJPEG. El problema tiene su origen en el hecho de que MatConvNet implementa VL_IMREADJPEG usando la librería del sistema operativo libjpeg, que no lee algunos formatos de imagen.

- Por supuesto hay que disponer de controladores actualizados para nuestro procesador gráfico y seguir las instrucciones de instalación y compilación de MatConvNet después de haber instalado adecuadamente CUDA y CUDNN.

6.4. Equipo

Los experimentos realizados se ejecutaron sobre un equipo con 16 GB de RAM, con un procesador AMD FX-6300 y una sola GPU Geforce GTX 1060 con 6 GB de

memoria.

Este modesto equipo es adecuado para experimentos exploratorios, si se desea trabajar con bases de datos mayores o imágenes de tamaño superior es recomendable emplear equipos más potentes.

El cuello de botella en el proceso de aprendizaje suele ser la cantidad de memoria disponible, configuraciones con menos de 6 GB de memoria en la GPU son poco recomendables.

6.5. Sistemas operativos y «toolboxes»

- Windows 10
- Ubuntu 16.04
- MatConvNet 1.0-beta24
- VLFeat 0.9.20
- CUDA 8.0
- MatLab R2017A (Linux y Windows)
- CUDNN v6
- Driver Nvidia 381.22 (código abierto) para Linux

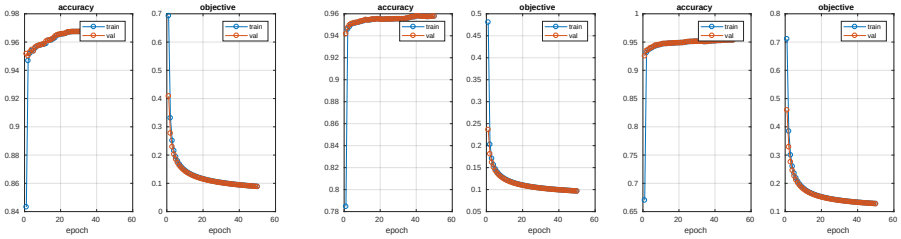
6.6. Modificaciones al modelo original

Se han implementado numerosos cambios y funciones auxiliares en el diseño original, manteniendo el núcleo del cómputo del aprendizaje de la red neuronal.

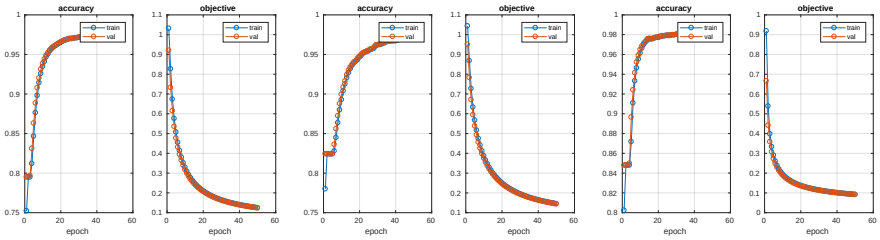
El «pipeline» se ha simplificado tanto como ha sido posible.

Para ejecutar la aplicación debe lanzarse la función *initprocess*. Esta función admite múltiples parámetros que permiten establecer las rutas necesarias: al modelo de red empleado, a las imágenes y sus segmentaciones, al directorio donde se generan los resultados, al conjunto de validación y un largo etcétera. Puede también indicarse el factor de aprendizaje, el numero de «folds» en un análisis «k-fold validation», las épocas que deseemos, el modo de uso: train, test o evaluación, el uso de la CPU o la GPU y otros.

Algunos de los parámetros, más propios del algoritmo de aprendizaje, deben ser cambiados en las funciones pertinentes, habiéndose centralizado en la función *initprocess* solo los más generales.



(a) Dermatológicas $i = 1$ (b) Dermatológicas $i = 2$ (c) Dermatológicas $i = 3$



(d) Lunares $i = 1$ (e) Lunares $i = 2$ (f) Lunares $i = 3$

Figura 13: Curvas de aprendizaje: Exactitud y error para el conjunto de imágenes dermatológicas (arriba) y el de lunares (abajo). 3 iteraciones.

7. Pruebas realizadas

Para analizar la robustez de los resultados se utilizó un esquema de validación cruzada (3 fold cross-validation).

Entrenamos la red con cinco conjuntos de datos distintos que se hallan resumidos en la tabla 1

Cada uno de estos conjuntos de datos se dividió, en el proceso de entrenamiento de la red, en tres particiones con el mismo número de elementos que denotaremos como A, B y C.

Para cada una de las pruebas que se realizaron, se entrenaron tres redes con los siguientes conjuntos de datos:

- $Entrenamiento = \{\{B\} \cup \{C\}\}$; $Validación = \{\{A\}\}$
- $Entrenamiento = \{\{A\} \cup \{C\}\}$; $Validación = \{\{B\}\}$
- $Entrenamiento = \{\{A\} \cup \{B\}\}$; $Validación = \{\{C\}\}$

Previo al entrenamiento, las posiciones de cada una de las imágenes de un grupo se permutan de manera aleatoria. Presentaremos en cada apartado la media de los valores obtenidos y su desviación estandar.

En la figura 13 podemos ver un ejemplo de las curvas de aprendizaje y error cometido en el entrenamiento de los conjuntos «ED» y «EL» (ver tabla 1) para cada una de las particiones del conjunto de datos.

En todos los casos se usaron imágenes reescaladas a unas dimensiones de 64×64 píxeles. El objetivo perseguido con este reescalado es doble, por una parte aceleramos el cómputo, ya que una imagen mayor tardaría mucho más en procesarse y, por otro lado, al disminuir el número de parámetros de entrada, pretendemos minimizar en lo posible el sobreajuste.

Los píxeles que la red predice como pertenecientes a la lesión se contabilizarán como verdaderos positivos (VP) si efectivamente pertenecen a ella y como falsos positivos (FP) si no es así. En cuanto a los píxeles predichos por la red como no pertenecientes a la lesión, distinguiremos entre los que efectivamente no pertenecen contabilizándolos como verdaderos negativos (VN) y a los que pertenecen aún siendo predichos como ajenos a la lesión lo haremos como falsos negativos (FN).

A partir de estos valores calcularemos algunos parámetros:

$$Sensibilidad = \frac{VP}{VP + FN}$$

$$Especificidad = \frac{VN}{VN + FP}$$

$$Precisión = \frac{VP}{VP + FP}$$

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN}$$

$$I.\,Jaccard = \frac{VP}{VP + FP + FN}$$

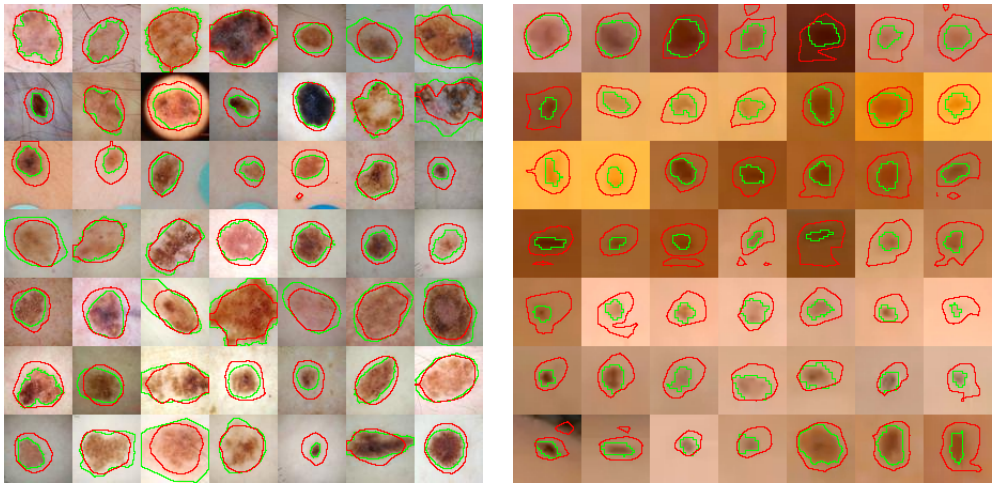
$$Dice\ Sorensen = \frac{2 \times VP}{(2 \times VP) + FP + FN}$$

Calculamos también el contraste de la lesión, expresado como la diferencia absoluta entre el valor medio de los píxeles de la ROI y el fondo. Expresamos esta diferencia en unidades de intensidad, siendo 0 la coincidencia absoluta y 255 la máxima diferencia posible.

En todos los casos hemos empleado 50 épocas de entrenamiento, cada una de las cuales se dividió en 10 subconjuntos o lotes de procesamiento. Utilizaremos el modelo de red «8s» que es el que aplica un salto de 8 píxeles.

Ajustaremos el factor de aprendizaje para cada una de las pruebas pero conservaremos inalterados otros parámetros del modelo como el momento el gradiente de descenso (0.9) o la probabilidad de las capas de descarte (0.5).

Los resultados obtenidos en la evaluación de los conjuntos de test con los distintos conjuntos de entrenamiento se aprecian en la tabla 2.



(a) Test con imágenes dermatológicas (TD).

(b) Test con imágenes de lunares (TL).

Figura 14: Segmentaciones obtenidas entrenando con el conjunto ED. En color verde la segmentación manual o «ground-truth». En color rojo la predicción de la red.

7.1. Entrenamiento con imágenes dermatológicas

Establecemos el índice de aprendizaje de la red en $\alpha = 1 \times 10^{-7}$.

Las redes resultantes de esta prueba son capaces de segmentar adecuadamente tanto imágenes dermatológicas como imágenes de lunares según puede observarse en la figura 14.

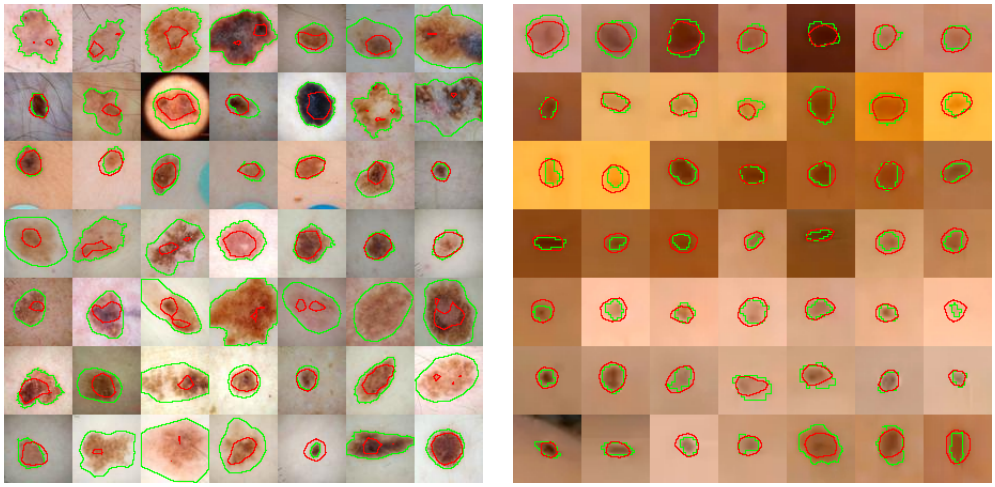
Es especialmente relevante el hecho de que obtengamos una elevada sensibilidad. La segmentación obtenida tiende a ser conservadora marcando — en la mayoría de los casos — un área ligeramente superior a la de referencia.

Considerando las características del problema, es preferible —desde el punto de vista clínico — detectar todas las lesiones aunque ello signifique un incremento en la tasa de falsos positivos.

A pesar de estos excelentes resultados debemos ser cautos debido a los siguientes factores:

- El reducido número de imágenes de entrenamiento de que dispone nuestra base de datos.
- El sobreajuste que se puede producir, en parte, debido al punto anterior.
- La heterogeneidad entre las imágenes dermatológicas.
- La generación aleatoria de los conjuntos de entrenamiento, podría arrojar valores diferentes dependiendo del orden en que las imágenes se hacen pasar por la red.

Estos resultados animan a continuar la investigación en esta linea.



(a) Test con imágenes dermatológicas (TD).

(b) Test con imágenes de lunares (TL).

Figura 15: Segmentaciones obtenidas entrenando la red con el conjunto EL. En color verde la segmentación manual o «ground-truth». En color rojo la predicción de la red.

7.2. Entrenamiento con lunares

Al igual que en el caso anterior fijamos $\alpha = 1 \times 10^{-7}$.

Podemos apreciar que en este caso el comportamiento de las redes es algo distinto. En la evaluación del conjunto de test de imágenes dermatológicas, el valor de la sensibilidad es muy inferior al del conjunto de test de lunares. Sucede igual con la exactitud. El valor correspondiente a la especificidad es muy alto debido a la práctica ausencia de falsos positivos.

En la figura 15 podemos observar como las segmentaciones de imágenes dermatológicas son peores que las obtenidas en las pruebas del apartado 7.1.

Debemos tener en cuenta varias consideraciones:

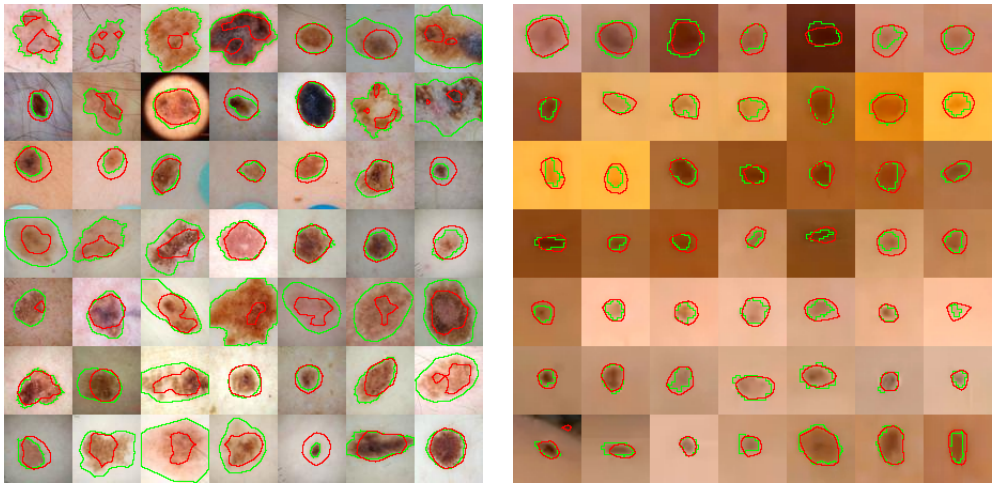
- Las imágenes de lunares se obtuvieron a partir de secuencias de fotografías de un brazo de las que se recortaron manualmente los lunares observados. El tamaño de las imágenes así obtenidas es muy pequeño, en ocasiones inferiores a 32×32 píxeles. Estas imágenes se reescalaron hasta el tamaño requerido en nuestra prueba, es decir 64×64 píxeles.
- Se trata de imágenes con un bajo contraste entre la ROI y el fondo.
- En cambio, las imágenes de origen dermatológico se obtuvieron mediante un dermatoscopio, bajo condiciones de iluminación más favorables que las de los lunares y en todos los casos eran de dimensiones muy superiores a los 64×64 píxeles. Estas imágenes también se reescalaron.
- Las Imágenes de la base de datos de lesiones dermatológicas son mucho más heterogéneas que las de la base de datos de lunares.

- El poco contraste y resolución de las las imágenes de lunares hace que en ocasiones las segmentaciones manuales sean muy inexactas.
- Aunque ambas bases de datos son pequeñas, la dermatológica esta compuesta por 550 imágenes, mientras que la de lunares solo de 279. Esto significa que la primera prácticamente dobla a la segunda.

En el caso de la predicción sobre el conjunto TL de lunares el ajuste de la red es muy bueno. No obstante, debemos tener en cuenta que las imágenes disponibles se obtuvieron de una secuencia de imágenes del mismo brazo. Por ello, aun cuando la iluminación y la orientación sea diferente, algunos imágenes corresponderán al mismo lunar.

Todos estos factores podrían provocar sobreajuste en este caso de test de lunares.

7.3. Entrenamiento con base de datos «Mix»



(a) Test con imágenes dermatológicas (TD).

(b) Test con lunares (TL).

Figura 16: Segmentaciones obtenidas entrenando la red con el conjunto TM. En color verde la segmentación manual o «ground-truth». En color rojo la predicción de la red.

Entrenaremos la red con la base de datos resultante de la unión de las empleadas en los apartados 7.1 y 7.2, denominaremos «Mix» a este conjunto.

Mantenemos $\alpha = 1 \times 10^{-7}$.

Los resultados obtenidos pueden consultarse en la tabla 2.

Vemos que parecen mejorar ligeramente los que obtuvimos entrenando con lunares en el apartado 7.2. En el caso de la sensibilidad no podemos extraer conclusiones válidas debido a la alta variabilidad.

Sobre el conjunto de test dermatológico (TD) mejoran sensiblemente especificidad y precisión pero, nuevamente debido a la variabilidad, no podemos extraer conclusiones firmes del indice de Jaccard o el Sorensen Dice.

Sobre el conjunto de test TL los resultados son similares a los obtenidos al entrenar la red con el conjunto EL de lunares.

En la figura 16 podemos ver las segmentaciones predichas para las dos iteraciones más cercanas a la media obtenida.

7.4. Data augmentation

Un modo de aumentar el número de imágenes de las que disponemos es efectuar transformaciones sobre ellas.

Pueden usarse distintos métodos para transformar las imágenes, rotaciones perpendiculares al plano que define la imagen ($altura \times anchura$), rotaciones sobre un eje contenido en el plano definido por la imagen (mirroring o reflejo), deformaciones, contracciones, dilataciones y otros.

En nuestro caso hemos aplicado las cuatro posibles rotaciones de 90 grados sobre el eje central perpendicular al plano de la imagen original y a su reflejo. Esto nos da un total de 8 imágenes por cada una de las originales.

Hemos obtenido así dos nuevas bases de datos, ver tabla 1 :

- I. dermatológicas aumentada (TD+). Compuesta por un total de 4400 imágenes, .
- I. de lunares aumentada (TL+). Compuesta por 2232 imágenes.

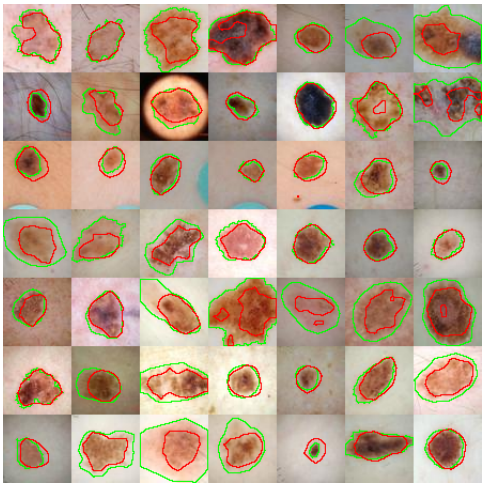
Mantenemos $\alpha = 1 \times 10^{-7}$

Los resultados obtenidos con el conjunto aumentado de imágenes dermatológicas se pueden observar en la tabla 2.

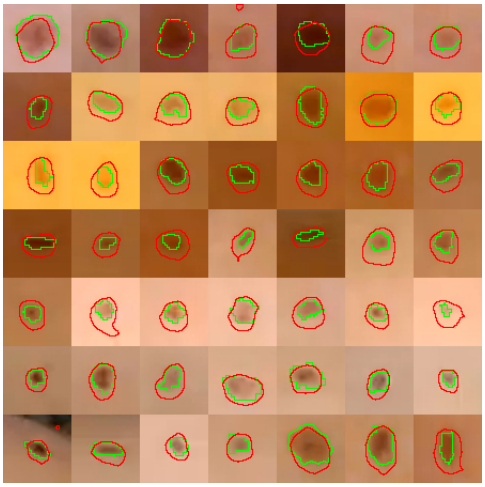
Estos resultados muestran una excelente sensibilidad, especificidad y exactitud sobre ambos conjuntos de evaluación siendo la precisión adecuada sobre el conjunto de test TD y algo menor sobre el TL.

Los resultados son muy similares a los obtenidos con la base de datos sin aumentar (TD) y creemos que al incrementar en un factor de 8 el número de imágenes de entrenamiento obtenemos una red más robusta, con menor tendencia al sobreajuste.

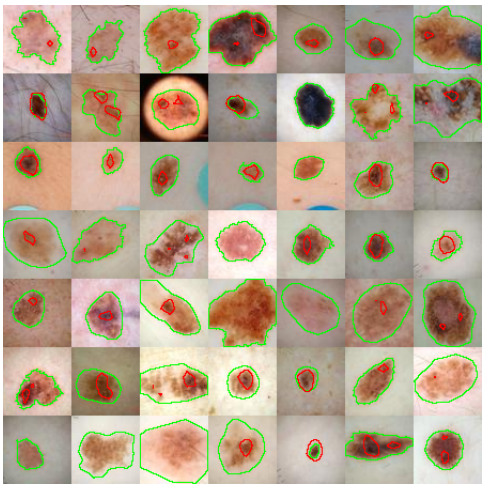
Los resultados obtenidos por la base de datos de lunares aumentada también son similares a los que conseguimos sin el «data augmentation», pueden observarse las segmentaciones obtenidas en la figura 17.



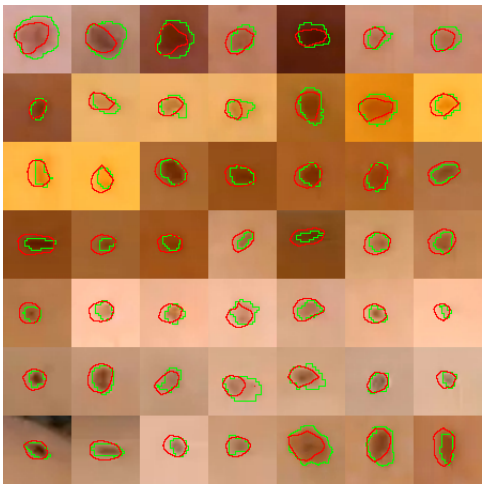
(a) Entrenamiento con TD+, test con



(b) Entrenamiento con TD+, test con TL



(c) Entrenamiento con TL+, test con TD.



(d) Entrenamiento con TL+, test con TL.

Figura 17: Segmentaciones obtenidas entrenando la red con los conjuntos TD+ y TL+. En color verde la segmentación manual o «ground-truth». En color rojo la predicción de la red.

	Entrenamiento						
	ED	EL	EM	ED+	EL+	Conjuntos	
	$(\bar{x}) \pm \sigma$					TD	Test
Sensibilidad	$0{,}92 \pm 0{,}10$	$0{,}37 \pm 0{,}28$	$0{,}58 \pm 0{,}26$	$0{,}74 \pm 0{,}20$	$0{,}22 \pm 0{,}26$		
Especificidad	$0{,}86 \pm 0{,}10$	$1{,}00 \pm 0{,}01$	$0{,}98 \pm 0{,}02$	$0{,}97 \pm 0{,}03$	$1{,}00 \pm 0{,}01$		
Precisión	$0{,}71 \pm 0{,}24$	$0{,}96 \pm 0{,}09$	$0{,}91 \pm 0{,}15$	$0{,}87 \pm 0{,}16$	$0{,}96 \pm 0{,}11$		
Exactitud	$0{,}85 \pm 0{,}08$	$0{,}75 \pm 0{,}21$	$0{,}81 \pm 0{,}17$	$0{,}86 \pm 0{,}12$	$0{,}71 \pm 0{,}21$		
Jaccard	$0{,}64 \pm 0{,}19$	$0{,}34 \pm 0{,}25$	$0{,}51 \pm 0{,}21$	$0{,}64 \pm 0{,}15$	$0{,}20 \pm 0{,}21$		
Sorensen Dice	$0{,}76 \pm 0{,}16$	$0{,}45 \pm 0{,}27$	$0{,}63 \pm 0{,}20$	$0{,}77 \pm 0{,}12$	$0{,}28 \pm 0{,}25$		
Contraste	50 ± 26	50 ± 26	50 ± 26	50 ± 26	50 ± 26		
	$(\bar{x}) \pm \sigma$						
Sensibilidad	$1{,}00 \pm 0{,}01$	$0{,}82 \pm 0{,}14$	$0{,}91 \pm 0{,}11$	$0{,}98 \pm 0{,}05$	$0{,}70 \pm 0{,}23$	TL	
Especificidad	$0{,}82 \pm 0{,}06$	$0{,}97 \pm 0{,}02$	$0{,}94 \pm 0{,}03$	$0{,}90 \pm 0{,}04$	$0{,}98 \pm 0{,}01$		
Precisión	$0{,}39 \pm 0{,}16$	$0{,}78 \pm 0{,}17$	$0{,}65 \pm 0{,}18$	$0{,}51 \pm 0{,}18$	$0{,}82 \pm 0{,}15$		
Exactitud	$0{,}83 \pm 0{,}06$	$0{,}95 \pm 0{,}03$	$0{,}93 \pm 0{,}03$	$0{,}90 \pm 0{,}03$	$0{,}94 \pm 0{,}05$		
Jaccard	$0{,}39 \pm 0{,}16$	$0{,}62 \pm 0{,}12$	$0{,}57 \pm 0{,}15$	$0{,}49 \pm 0{,}16$	$0{,}67 \pm 0{,}12$		
Sorensen Dice	$0{,}53 \pm 0{,}16$	$0{,}76 \pm 0{,}09$	$0{,}71 \pm 0{,}13$	$0{,}64 \pm 0{,}14$	$0{,}69 \pm 0{,}15$		
Contraste	28 ± 6	28 ± 6	28 ± 6	28 ± 6	28 ± 6		

Tabla 2: Resumen de resultados

8. Conclusiones

La utilización de redes completamente convolucionales para la segmentación semántica de lesiones melanocíticas, es un método adecuado.

Entrenando la red con imágenes provenientes de bases de datos dermatológicas, de alta calidad y elevado contraste, se obtiene una exactitud del $86{,}0 \pm 2\,\%$, una especificidad del $97 \pm 1\,\%$, una precisión del $71 \pm 24\,\%$ y una sensibilidad del $74 \pm 5\,\%$ para el conjunto de evaluación de imágenes dermatológicas.

En el caso de evaluar un conjunto de lunares de bajo contraste y poca resolución, obtenidas a partir de pequeños recortes de imágenes adquiridas con la cámara de un móvil, los resultados también fueron excelentes. La exactitud fue del $90 \pm 3\,\%$, la especificidad del $90 \pm 3\,\%$ y la sensibilidad del $98 \pm 1\,\%$. La precisión disminuyó hasta un $39 \pm 16\,\%$ debido a que la red segmenta un área mayor a la marcada manualmente.

En el caso de entrenar la red con imágenes de baja resolución de los lunares, se obtuvieron resultados sensiblemente inferiores.

Comprobamos que, aunque predicen bien el conjunto de test del mismo tipo de imágenes, no generalizan adecuadamente sobre el conjunto de test dermatológico. Incluso utilizando un conjunto aumentado se obtuvo una sensibilidad de tan solo el $22 \pm 9\,\%$.

Este comportamiento se observa tanto en la base de datos de lunares original como en la aumentada.

El entrenamiento con bases de datos aumentadas parecen mejorar ligeramente los resultados en el caso de las imágenes de origen dermatológico. No se aprecian diferencias importantes en el caso de la base de datos de lunares aumentada .

Los mejores resultados, tanto visual como numéricamente, se obtienen con la base de datos dermatológica aumentada. Sobre el conjunto de test de imágenes dermatológicas la sensibilidad fue de un $74 \pm 20\,\%$ la especificidad de un $97 \pm 3\,\%$ la precisión de un $87 \pm 16\,\%$ y la exactitud de un $86 \pm 12\,\%$. Sobre el conjunto de test de lunares la sensibilidad fue de $70 \pm 23\,\%$, la especificidad del $98 \pm 1\,\%$, la precisión del $51 \pm 18\,\%$ y la exactitud del $90 \pm 3\,\%$.

Los resultados no resultan fácilmente comparables sin la ayuda visual de las imágenes de las segmentaciones obtenidas. Creemos que una de las métricas que mayor información aporta sobre la bondad de la segmentación, es el índice o coeficiente de Jaccard, que mide el grado de similitud de dos conjuntos. Efectivamente, si consideramos ambos conjuntos de test, se obtienen valores más elevados entrenando la red con ED+ que con los demás conjuntos de entrenamiento.

9. Trabajo futuro

Mencionaremos algunas de las cuestiones que surgieron durante la elaboración del presente trabajo y que merecen, a nuestro parecer, un estudio más detallado:

Factor de aprendizaje variable Las redes convolucionales aprenden las características más generales en las primeras capas de entrenamiento.

En este estudio se mantuvo constante el factor α en todas las capas. Una posibilidad a explorar seria la de establecer un α menor en las primeras e irlo aumentando progresivamente con la profundidad de la capa. De este modo no alteraríamos significativamente las características más generales — aprendidas en el pre-entrenamiento de la red— para conceder, progresivamente con la profundidad de la capa, mayor importancia a las más específicas de nuestro conjunto de datos.

Superresolución Las imágenes de lunares se capturaron de fotografías hechas con la cámara de un móvil en condiciones de iluminación poco favorables. Estas imágenes no ofrecen una resolución y contraste adecuados. Una opción a explorar seria la de aplicar algoritmos de superresolución a las imágenes obtenidas, evaluando el comportamiento de la red ante estas mejoras.

Tamaño de las imágenes Las imágenes estudiadas en nuestras bases de datos se reescalaron a unas dimensiones de $64 \times 64 \times 3$. Tamaños superiores ofrecen mayor cantidad de información a la red con lo cual pueden mejorar su funcionamiento. Tamaños inferiores podrían afectar negativamente a su desempeño.

Las dimensiones elegidas en nuestro estudio, se deben a que queríamos comprobar el comportamiento de ambas bases de datos en condiciones similares. El tamaño de las imágenes de lunares, como comentamos anteriormente, era inferior pero próximo a $64 \times 64 \times 3$, de ahí nuestra elección.

Una posibilidad que surgió durante la elaboración de este proyecto, fue la de dividir las imágenes dermatológicas de gran resolución en teselas de menor tamaño. Esto no es preciso en redes completamente convolucionales, puesto que admiten tamaños arbitrarios de las imágenes de entrada. En nuestro caso, el tamaño de las imágenes disponibles no era homogéneo y como la red las procesa por lotes para aprovechar las capacidades de paralelización de la GPU, no era viable enviarlas directamente. Dividir una imagen en teselas de tamaño regular podría solucionar este problema, aunque se precisaría comprobar como influye esto en el aprendizaje.

Distancia de Hausdorff La distancia de Hausdorff mide como de lejos están uno de otro dos subconjuntos compactos de un espacio métrico. Este parámetro podría ayudarnos a valorar la bondad de la predicción de la red, comparándola con la segmentación manual.

Sin embargo, el algoritmo de que disponíamos no era adecuado, puesto que de la predicción de la red no se obtiene necesariamente una superficie cerrada, sino una nube de puntos. Necesitamos ajustar el algoritmo para estas condiciones.

Modelos de red, otras arquitecturas. Hemos estudiado el comportamiento del modelo FCN-MatConvNet con nuestras bases de datos. Este modelo se entrenó previamente con la base de datos PASCAL VOC 2011 . Otras arquitecturas entrenadas con distintas bases de datos podrían arrojar resultados distintos.

Clasificación El modelo de red empleado se ha utilizado para la segmentación semántica de las imágenes de prueba. Este modelo puede ser utilizado para establecer, al mismo tiempo, una clasificación en categorías de las imágenes evaluadas, lo cual aportaría información interesante que actualmente no tenemos en cuenta.

Bibliografía

[1] deeplearning.net. Deep learning. http://deeplearning.net/reading-list/. visitado el 31 de mayo de 2017.

[2] T. L. Diepgen and V. Mahler. The epidemiology of skin cancer. *British Journal of Dermatology*, 146(s61):1–6, 2002.

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, Feb. 2017.

[4] M. Fornaciali, M. Carvalho, F. V. Bittencourt, S. Avila, and E. Valle. Towards automated melanoma screening: Proper computer vision & reliable results. Apr. 2016.

[5] R. J. Friedman, D. S. Rigel, and A. W. Kopf. Early detection of malignant melanoma: The role of physician examination and self-examination of the skin. *CA: A Cancer Journal for Clinicians*, 35(3):130–151, 1985.

[6] S. Goldsmith and A. Solomon. A series of melanomas smaller than 4 mm and implications for the abcde rule. *Journal of the European Academy of Dermatology and Venereology*, 21(7):929–934, Aug 2007.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *JMLR WCP*, 28::1319–1327,2013, Feb. 2013.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Feb. 2015.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[12] A. Karpathy. Cs231n: Convolutional neural networks for visual recognition., 2017. Accedido el 7/06/2017.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] A. Masood and A. Ali Al-Jumaily. Computer aided diagnostic support system for skin cancer: A review of techniques and algorithms. *International Journal of Biomedical Imaging*, 2013:22, 2013.

[15] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[16] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira. Ph 2-a dermoscopic image database for research and benchmarking. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 5437–5440. IEEE, 2013.

[17] Morton and Mackie. Clinical accuracy of the diagnosis of cutaneous malignant melanoma. *British Journal of Dermatology*, 138(2):283–287, 1998.

[18] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. May 2016.

[19] R. L. Siegel, K. D. Miller, and A. Jemal. Cancer statistics, 2017. *CA: A Cancer Journal for Clinicians*, 67(1):7–30, 2017.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. Sept. 2014.

[21] A. C. Society. Cancer facts & figures 2017. Atlanta:American Cancer Society, 2017.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. Sept. 2014.

[24] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.