

Clasificación de melanomas a través de redes neuronales residuales

Universidad de Córdoba

Escuela Politécnica Superior de Córdoba

GRADO EN INGENIERÍA INFORMÁTICA MENCIÓN
EN COMPUTACIÓN

Petición de Tema

Juan José Méndez Torrero

Directores:

D. Pedro Antonio Gutiérrez Peña

D. Javier Sánchez Monedero

27 de mayo de 2019

Índice general

1. Datos del proyecto	1
2. Introducción	3
3. Objetivos	6
4. Antecedentes	8
4.1. Conjunto de datos ISIC	8
4.2. Redes neuronales residuales (ResNet)	9
4.3. OpenCV	9
4.4. Keras	10
5. Fases de desarrollo del proyecto	11
5.1. Análisis y estudio del problema	11
5.2. Diseño	11
5.3. Codificación	12
5.4. Pruebas	12
5.5. Documentación	12
6. Recursos	13
6.1. Recursos humanos	13
6.2. Recursos materiales	13
6.2.1. Recursos software	13
6.2.2. Recursos hardware	14
7. Planificación temporal del proyecto	15

Índice de tablas

7.1. Distribución temporal del proyecto	15
---	----

Capítulo 1

Datos del proyecto

Título del proyecto: Clasificación de melanomas a través de redes neuronales residuales.

Autor:

- **Nombre:** Juan José Méndez Torrero.
- **Documento Nacional de Identidad:** 31018712-S.
- **Correo Electrónico:** i42metoj@uco.es.
- **Titulación:** Grado en ingeniería informática.
- **Especialidad:** Mención en Computación.

Directores del proyecto:

- D. Pedro Antonio Gutiérrez Peña.
- D. Javier Sánchez Monedero

Firma del autor de director del proyecto.

■ **Autor:**

Fdo. Juan José Méndez Torrero

■ **Directores**

Fdo. Pedro Antonio Gutiérrez Peña

A handwritten signature in blue ink, appearing to read 'Javier Sánchez Monedero', is written over a horizontal line. The signature is stylized with a large 'J' and a circular flourish.

Fdo. Javier Sánchez Monedero

Capítulo 2

Introducción

Desde los años 50, Alan Mathison Turing, un matemático y científico de la computación nacido en Reino Unido, decidió mostrar a la gente común lo que una máquina podría ser capaz de hacer. Este matemático escribió un artículo titulado *Computing Machinery and Intelligence* [1] en el que proponía una serie de juegos en los que, una máquina no lógica podría llegar a aprender. Pues bien, desde aquel artículo han surgido muchos términos para definir la acción de pensar para una máquina. Los más conocidos son *Machine Learning* [2] y *Deep Learning* [3], que son técnicas de aprendizaje en las que la máquina es capaz de simular el pensamiento humano.

Machine Learning [2] y *Deep Learning* [3] son técnicas de aprendizaje parecidas, aunque su principal diferencia es que la segunda eleva el nivel de aprendizaje a uno más detallado que el primero. Por esto, se dice que la técnica *Deep Learning* [3] es un subconjunto dentro del campo de la técnica *Machine Learning* [2]. Esta técnica consiste en la creación de una lista de reglas para solventar un problema dado. En el caso de la técnica *Deep Learning* [3] consiste en la creación de un modelo que simule el funcionamiento de un cerebro humano, haciendo que simulen cuándo una neurona se activa o desactiva.

Estas técnicas han avanzado tanto que ya han llegado a límites en los que sus creadores no hubieran imaginado, desde la creación de coches automáticos hasta máquinas que son capaces de reconocer la voz humana e interpretarla. Además de para el uso recreativo, las técnicas de *Machine Learning* [2] y *Deep Learning* [3] se utilizan para ayudar a

otras áreas de estudio, como la medicina. Gracias a estas técnicas, una máquina ya es capaz de reconocer los síntomas de un paciente y realizar un diagnóstico según varios factores.

Además, se ha conseguido aligerar muchas tareas que antes tenían que realizarse por una persona, tales como, realizar un reconocimiento de personas en una imagen o la captura del movimiento dentro de una escena. De esta forma, trabajos muy complicados para ser realizados por el ser humano, se han convertido en tareas en las que sólo hace falta la presencia humana para interpretar o tomar acciones con los resultados obtenidos por la máquina que aprende gracias a las técnicas comentadas anteriormente.

La visión por computador es una de las técnicas que más han avanzado gracias al *Deep Learning* [3], lo que ha impulsado aplicaciones como la detección de cáncer en la piel de diversos tipos. Un melanoma, según la fundación de cáncer en la piel *Skin Cancer Foundation* [4], es la forma de cáncer de piel más peligrosa. Esta fundación nos enseña cómo poder detectar precozmente si estamos ante una lesión de tipo benigno o si se trata de un melanoma (tipo de lesión maligna). Para ello hace uso de las primeras letras del alfabeto, siendo *A* por asimetría, *B* por el borde de la lesión, *C* por el color, *D* por el diámetro y *E* por la evolución de la lesión. Por ejemplo, en el caso de tener una lesión en la piel en la que su forma y borde no son uniformes, el color de ésta es oscuro, su diámetro es mayor de 6mm y vemos que durante un tiempo su forma ha cambiado y que tenemos nuevos síntomas, entonces, según estas reglas, estaríamos ante un tipo de lesión maligna, es decir, ante un melanoma.

Para este proyecto, haremos uso de modelos de *Machine Learning* [2] y *Deep Learning* [3], para la clasificación de lesiones en la piel a través de imágenes. Utilizaremos las imágenes incluidas el conjunto de datos *ISIC* [5], en el cual encontramos multitud de imágenes etiquetadas por expertos según el tipo de lesión en cuestión. En principio, estas imágenes pueden ser agrupadas según muchos atributos, tales como, la edad aproximada, la clase de lesión o incluso según la longitud de su diámetro. Para este estudio clasificaremos las imágenes según si son un

tipo de lesión benigna o maligna.

Capítulo 3

Objetivos

El principal objetivo de este proyecto es el diseño de una herramienta capaz de clasificar correctamente imágenes de lesiones en la piel en función del tipo de lesión. Esta clasificación se realizará con técnicas de *Machine Learning* [2], más concretamente técnicas basadas en el aprendizaje profundo (*Deep Learning* [3]). El conjunto de imágenes usadas para este proyecto será el de la base de datos de *ISIC* [5].

Con respecto al modelo a usar para la clasificación, utilizaremos una red neuronal artificial residual, la cuál se creará con una biblioteca del lenguaje *Python* [6] llamada *Keras* [7], ya que nos permite la creación de modelos de redes neuronales de una manera sencilla e intuitiva. Elegimos este tipo de red con la idea de reproducir los resultados recientemente publicados en el artículo *Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks* [8].

Junto con estos objetivos, podremos encontrar los siguientes:

- El primer objetivo a solventar será el procesamiento de los datos, es decir, el procesamiento de las imágenes. Para ello, haremos uso de otra biblioteca en *Python* [6] llamada *OpenCV* [9], la cual nos permitirá realizar operaciones sobre las imágenes de manera más fácil. Nuestra idea es usar esta biblioteca para obtener la máscara de la imagen haciendo uso de operadores de segmentación.

Una vez conseguida esa máscara, se la aplicaremos a la imagen original para quedarnos sólo con el contorno del melanoma en cuestión.

- El segundo objetivo, como hemos comentado anteriormente, será la

creación de una red neuronal artificial residual. Ésta se encargará de clasificar las imágenes entre lesión benigna o lesión maligna. Esta fase es la más tediosa, ya que tendremos que evaluar el modelo con diferentes hiper-parámetros para el algoritmo de entrenamiento, con el objetivo de encontrar los valores óptimos para este modelo.

- Finalmente, una vez encontrado el modelo óptimo, se realizará un análisis de los resultados obtenidos, viendo si nuestro modelo es capaz de clasificar correctamente las imágenes de lesiones suministradas en la fase de test.

Capítulo 4

Antecedentes

4.1. Conjunto de datos ISIC

El conjunto de datos ISIC [5] fue creado por una asociación académica para facilitar la obtención de imágenes digitales de la piel con el objetivo final de reducir la mortalidad por melanoma. Este conjunto de casos clínicos se elabora de manera colaborativa entre varias industrias internacionales y cuenta con un total de 23.906 imágenes de lesiones en la piel, las cuales son clasificadas como benignas, malignas, intermedias o desconocidas. Para este estudio sólo se recogerán imágenes de lesiones benignas y malignas.

Desde el año 1979, la cantidad de personas que son diagnosticadas con un melanoma ha aumentando muchísimo y, aunque esta enfermedad es más frecuente después de los cuarenta años, la incidencia en jóvenes ha aumentado debido, en parte, al mayor uso de dispositivos de bronceados artificial. En 2019, los cálculos de la *Sociedad Americana Contra el Cáncer* [10] estima que se producirán 96.480 nuevos casos de melanoma que serán diagnosticados y que aproximadamente 7.230 personas morirán a causa de un melanoma.

Por consiguiente, esta institución ha recopilado un conjunto de imágenes para su uso en medicina preventiva. Su objetivo es que el porcentaje de personas fallecidas disminuya gracias a un diagnóstico precoz del melanoma, ya que un melanoma es más fácil de eliminar si se ha diagnosticado con antelación.

4.2. Redes neuronales residuales (ResNet)

Las redes neuronales residuales o ResNet [11] (*Residual Networks*) son un tipo de modelo de red neuronal artificial que se usan para el reconocimiento de imágenes. Este modelo surgió en la empresa *Microsoft*, y sus creadores [12] la usaron para competir en la famosa competición de reconocimiento de imágenes *ImageNet* [13]. Su intención era que este modelo simulara las células piramidales en la corteza cerebral, creando una serie de saltos o *atajos* para que la información producida por una capa, pudiera saltar la siguiente y enviar su información a capas posteriores.

Igualmente, su objetivo es también el de intentar solventar el problema de la desaparición de los gradientes. Esto se consigue gracias a la reutilización de las activaciones de una capa anterior hasta que la capa adyacente aprenda su peso. Gracias a lo comentado hasta ahora, la red *ResNet* [11] permite crear un modelo con una gran cantidad de capas, como por ejemplo la usada en la competición *ImageNet* [13] de 2015 (152 capas).

4.3. OpenCV

OpenCV [9] (*Open Source Computer Vision Library*) es una biblioteca de código abierto que se usa para apoyar el desarrollo de sistemas inteligentes en visión y aprendizaje automático. Esta biblioteca cuenta con más de 2.500 algoritmos optimizados para realizar las tareas anteriormente comentadas. Tiene interfaces con los lenguajes de programación *C++* [16], *Python* [6], *Java* [17] y *Matlab* [18] y es compatible con los sistemas operativos *Windows*, *Linux*, *Android* y *Mac OS*. Inicialmente, esta biblioteca está escrita de forma nativa en *C++*.

Una de sus ventajas es que permite realizar tareas de procesamiento sobre imágenes y vídeos de manera muy sencilla, facilitando tareas como la aplicación de un tipo de filtro a una imagen dada. Otra tarea que también puede realizar es la creación de redes neuronales sencillas para clasificación de imágenes. Por ejemplo, se podría usar para que re-

conociera las siluetas de una serie de personas y, posteriormente, crear un modelo que fuera capaz de aprender esas siluetas.

4.4. Keras

Keras [7] es una biblioteca de software libre que se utiliza para la creación de redes neuronales artificiales profundas. Esta biblioteca está escrita en *Python* [6] y es capaz de ejecutarse sobre otras bibliotecas como *Theano* [19], *TensorFlow* [20] o *Microsoft Cognitive Toolkit* [21]. Fue creada por el desarrollador *François Chollet* el 27 de marzo de 2015.

Permite al usuario común poder crear modelos de redes neuronales artificiales profundas de manera muy sencilla y rápida. Cabe destacar que la comunidad que hay detrás de esta biblioteca es muy grande, con lo que el código que implementa tendrá más funciones y permitirá en un futuro que el usuario pueda crear cualquier tipo de red neuronal artificial. Además, esta biblioteca permite generar modelos de *Deep Learning* [3] en dispositivos con *iOS* o *Android*.

Capítulo 5

Fases de desarrollo del proyecto

5.1. Análisis y estudio del problema

En la primera fase proyecto, se estudiará el problema desde un punto de vista teórico, centrándose en cómo procesar la base de datos elegida y cómo funciona una red neuronal artificial residual. También se estudiará el funcionamiento de las GPUs, de la biblioteca *Keras* [7] y el lenguaje de programación *Python* [6].

Los requisitos del sistema son los correspondientes objetivos explicados en el capítulo 3. Este problema se puede dividir en dos:

- Problema real: generar un clasificador capaz de distinguir entre lesiones malignas y lesiones benignas, utilizando la base de datos *ISIC* [5].
- Problema técnico: crear una herramienta software utilizando el lenguaje de programación *Python* [6] y las bibliotecas *Keras* [7] y *OpenCV* [9] que entrene un modelo de red neuronal residual para la clasificación de imágenes de lesiones en la piel.

5.2. Diseño

Una vez terminado el análisis y estudio de los requisitos, se pasará al diseño del software para posteriormente poder codificarlo. El diseño intentará satisfacer de la mejor forma posible los objetivos y requisitos anteriormente comentados. Además, se analizarán los datos de entrada, haciendo un procesamiento de los mismos para obtener los mejores re-

sultados. Finalmente, se analizarán los parámetros que más influyen en la clasificación para configurarlo de una manera óptima.

5.3. Codificación

El procesamiento de las imágenes y la creación del modelo se realizarán haciendo uso del lenguaje de programación *Python* [6] junto con las bibliotecas *Keras* [7] y *OpenCV* [9]. La primera biblioteca se utilizará para la creación del modelo de red neuronal y la segunda para el procesamiento previo de la base de datos *ISIC* [5].

5.4. Pruebas

El software resultante se someterá a una serie de pruebas para solucionar los posibles errores que se produzcan. Finalmente, cuando nuestro modelo haya superado las pruebas anteriores, se someterá a nuevas pruebas para analizar que los objetivos y requisitos comentados anteriormente quedan satisfechos.

5.5. Documentación

Por último, este software irá acompañado de un manual técnico en el que se explicarán los métodos utilizados para la creación de mismo, junto con un manual de usuario en el que se recogerán los pasos necesarios para poder hacer uso del software creado.

Capítulo 6

Recursos

Para la realización de este proyecto se dispondrá de los siguientes recursos:

6.1. Recursos humanos

- **Autor:** Juan José Méndez Torrero.

Alumno de cuarto curso del Grado en Ingeniería Informática en la mención de computación.

- **Directores:**

- D. Pedro Antonio Gutiérrez Peña.

Profesor Titular de Universidad del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba y miembro investigador del grupo AYRNA [22].

- D. Javier Sánchez Monedero.

Investigador de la Universidad de Cardiff y miembro investigador del grupo AYRNA [22].

6.2. Recursos materiales

6.2.1. Recursos software

- Mac OS Mojave Versión 10.14.4 y Ubuntu 18.04 como Sistemas Operativos.
- *Visual Studio Code* como entorno de desarrollo.

- *Python* como lenguaje de programación.
- *OpenCV* como herramienta para el procesamiento de las imágenes.
- *Keras* como herramienta para la creación de los modelos neuronales haciendo uso como *backend* de *TensorFlow*.
- *LaTeX* para la creación de la documentación.

6.2.2. Recursos hardware

Para la realización de este proyecto se hará uso de un equipo personal con las siguiente características:

- MacBook Pro (Retina 13 pulgadas, año 2017)
- Procesador: 2,3 GHz Intel Core i5.
- Memoria: 8GB 2133 MHz LPDDR3.
- Gráficos: Intel Iris Plus Graphics 640 1536 MB.
- 128 GB almacenamiento flash PCIe.

Para la realización de las pruebas se hará uso de los recursos hardware del grupo de investigación AYRNA[22], los cuales consisten en un clúster con las siguientes características:

- Procesador: 8 procesadores Intel Xeon 3.2 GHz (32 bits y 2 núcleos de procesos) y 8 procesadores Intel Xeon 2 GHz E5405 CPUs (64 bits y 4 núcleos de procesos).
- Memoria: 1GB de RAM por núcleo de proceso.
- Capacidad: Disco duro de 2 TB.

Capítulo 7

Planificación temporal del proyecto

Para la realización de las fases de desarrollo de este proyecto, especificadas en el capítulo 5, se hará una distribución que se encuentra especificada en la Tabla 7.1

Actividad	Junio	Julio	Agosto	Septiembre
Obtención de conocimientos	20	0	0	0
Análisis de requisitos	20	0	0	0
Diseño	40	30	0	0
Codificación	0	70	60	0
Pruebas	0	0	30	10
Documentación	0	0	0	20

Tabla 7.1: Distribución temporal del proyecto

Todas las fases del proyecto suman 300 horas.

Bibliografía

- [1] Alan Mathison Turing. *Computing Machinery and Intelligence*. Mind, Octubre 1950.
- [2] Ethem Alpaydin. *Machine Learning: The New AI*. MIT Press, 2016.
- [3] Ian Goodfellow and Yoshua Bengio. *Deep Learning, Adaptive Computation and Machine Learning series*. MIT Press, 2017.
- [4] Skin Cancer Foundation, <https://cancerdepiel.org/cancer-de-piel/melanoma>, última consulta: 23-05-2019.
- [5] International Skin Imaging Collaboration, *Dataset ISIC* <https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery>, última consulta: 22-05-2019.
- [6] Python <https://www.python.org/>, última consulta: 22-05-2019.
- [7] François Chollett, Keras <https://keras.io>, última consulta: 22-05-2019.
- [8] Lequan Yu, Hao Chen, Qi Dou, Jing Qin y Pheng-Ann Heng, *Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks*, IEEE Transactions on medical imaging, Vol. 36, 4 de abril de 2017.
- [9] OpenCV, <https://docs.opencv.org/3.4.5/>, última consulta: 22-05-2019.
- [10] American Cancer Society *¿Cuáles son las estadísticas principales del cáncer de piel tipo melanoma?*, <https://www.cancer.org/es/cancer/cancer-de-piel-tipo-melanoma/acerca/estadisticas-clave.html>, última consulta: 26-05-2019.

- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Deep Residual Learning Image Recognition*, sección 4.1, Microsoft Research, 2016.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Deep Residual Learning Image Recognition*, Microsoft Research, 2016.
- [13] ImageNet, *Large Scale Visual Recognition Challenge 2016* <http://www.image-net.org/>, última consulta: 22-05-2019.
- [14] Ivor Armstrong Richards, *Systema: connecting Matter, Life, Culture and Technology*, 2015.
- [15] Henry J. Kelley, *Gradient theory of optimal flight paths*, 1960.
- [16] Bjarne Stroustrup, Dinamarca, 1979.
- [17] Sun Microsystems, Oracle Corporation, 1996. <https://www.oracle.com/sun/>, última consulta: 22-05-2019.
- [18] Cleve Moler y MathWorks, Utah, Estados Unidos, 1984.
- [19] Montreal Institute for Learning Algorithms, 2007, <https://mila.quebec/en/>, última consulta: 22-05-2019.
- [20] Google Brain, 2015, <https://ai.google/research/teams/brain/>, última consulta: 22-05-2019.
- [21] Microsoft Research, 2016, <https://www.microsoft.com/en-us/research/>, última consulta: 22-05-2019.
- [22] Pedro Antonio Gutiérrez Peña and Juan Carlos Fernández Caballero. Grupo de investigación y redes neuronales artificiales ayna, <https://www.uco.es/ayrna/>, última consulta: 22-05-2019.