MÁSTER EN DEEP LEARNING

2018/2019

DETECCIÓN DE CÁNCER MEDIANTE
REDES NEURONALES CONVOLUCIONALES

Otto Francisco Wagner Navarro

Proyecto dirigido por:
Dr. Fernando Corbacho
Dr. Manuel Sánchez-Montañés

MÁSTER EN DEEP LEARNING

2018/2019

DETECCIÓN DE CÁNCER MEDIANTE REDES NEURONALES CONVOLUCIONALES

Otto Francisco Wagner Navarro
(ofwagner@ottofwagner.com)

Proyecto dirigido por:
Dr. Fernando Corbacho
Dr. Manuel Sánchez-Montañés

Abstract

El cáncer es una de las enfermedades que más preocupa a la sociedad. Anualmente provoca millones de muertes a la vez que se producen millones de casos nuevos. En los diagnósticos radiológicos, en muchas ocasiones, no se detectan los indicios de cáncer, bien por falta de personal o bien por falta de especialización. En este trabajo trataremos de crear una prueba de concepto de un modelo que nos ayude a predecir, mediante el análisis digital de radiografías de tórax, hallazgos patológicos que puedan indicar la presencia de cáncer. Creemos que el uso de la Inteligencia Artificial, en este caso el *Deep Learning*, puede ser un gran apoyo a la labor del facultativo radiólogo.

Agradecimientos

Cursar un máster es un largo camino que dura aproximadamente un año. Un recorrido apasionante, lleno de senderos maravillosos, pero también, lleno de curvas, montañas, baches, etc. El final de este camino llega con la realización y posterior defensa del Proyecto Fin de Máster. Afortunadamente en este camino no se está sólo y se cuenta con innumerables compañeros de viaje.

En primer lugar, me gustaría dar las gracias a mis compañeros, por todo lo vivido y compartido: las aportaciones en clase, los consejos a la hora de afrontar los trabajos, las experiencias compartidas, las risas en las comidas de "menos de una hora", etc.

Agradecer a mis tutores, Fernando y Manuel, por su tiempo, conocimientos, experiencia y dedicación a la hora de dirigirme y orientarme en el proyecto final.

Gracias a mis profesores por haber intentado hacer fácil lo difícil, grandes profesionales, pero mejores personas aún. En este apartado quiero recordar no sólo a los profesores del máster, sino también a los profesores que me han acompañado desde mis primeros años del colegio, pues ellos me enseñaron "el camino".

Quisiera, también, dar las gracias a mi familia y a mis amigos por el apoyo que me han dado, y el haber entendido mis ausencias por mi dedicación al máster. En este apartado tengo que mencionar a mi padre quien me inculcó las matemáticas desde que tengo "uso de razón" y especialmente a mi madre, quien siempre me da alas y me "enseñó a volar".

No quisiera olvidar a dos personas que ya no están, mis Yayos. Hace 25 años que el cáncer se los llevó, pero sé que desde el Cielo me han guiado para hacer este trabajo, y, por tanto, este trabajo es tanto de ellos como mío.

Índice:

1 Planteamiento del problema y revisión de la literatura

1.1 Contexto

Según el estudio titulado "Projections in Breast and Lung Cancer Mortality among Women: A Bayesian Analysis of 52 Countries Worldwide" (Martín-Sánchez *et al.*, 2018) y *Global Cancer Observatory* (2018), el cáncer es una de las principales causas de morbilidad (número de personas que enferman en una población y período determinados) y mortalidad mundial. Los nuevos datos sobre el cáncer sugieren que el censo mundial de esta enfermedad ha aumentado a 18,1 millones de casos y causa 9,6 millones de muertes al año (*New Global Cancer Data: GLOBOCAN 2018 | UICC*, 2018). Este mismo informe sostiene que "hay algunos indicios de que la intensificación de los esfuerzos de prevención está empezando a reducir las tasas de incidencia del cáncer, por ejemplo, una menor incidencia del cáncer de pulmón en los hombres de Europa septentrional y América del Norte, o en el cáncer de cuello uterino en la mayoría de las regiones, excepto en el África subsahariana, en comparación con 2012".

En la siguiente infografía podemos observar cómo los cánceres de pulmón y pecho ocupan un 23,2 % de los cánceres diagnosticados. En concreto, el cáncer de pulmón supone un 18,4 % de todas las muertes producidas por cáncer.

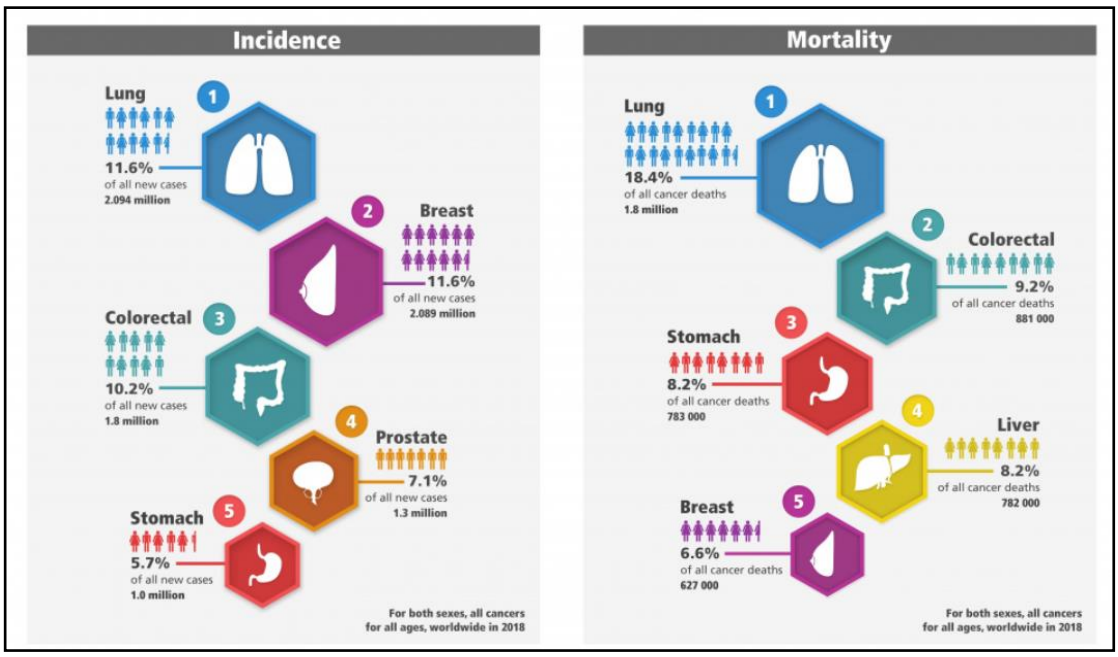


Figura 1: Porcentaje de nuevos casos de cáncer y muertes en 2018
(*New Global Cancer Data: GLOBOCAN 2018* | *UICC*, 2018; *Global Cancer Observatory*, 2018)

Los cánceres de pecho y pulmón se localizan en la caja torácica y, por lo tanto, una primera exploración suele estar acompañada por radiografías de tórax (Rogers, 2010). Esta tarea consume mucho tiempo, ya que, requiere que radiólogos expertos estudien las imágenes. Esto conlleva dos problemas que pueden provocar un error de diagnóstico: fatiga del profesional y falta de experiencia diagnóstica en áreas del mundo donde los radiólogos no están disponibles (Rajpurkar *et al.*, 2018). Por todo ello, parece interesante la ayuda que puede ofrecer la Inteligencia Artificial al diagnóstico precoz de estas enfermedades.

En medicina, la Inteligencia Artificial tiene un impacto a tres niveles (Topol, 2019): 1) interpretación de imágenes; 2) mejora del flujo de trabajo y reducción de errores; 3) facilidad para que los pacientes obtengan sus datos. Litjens *et al.* (2017) hacen una revisión de los principales usos del Deep Learning en la medicina, dentro de los que podemos destacar:

- Clasificación
    - Clasificación de imágenes en general: radiografías, imágenes de resonancias, etc.
    - Clasificación de objetos en imágenes: como pueden ser hallazgos en radiografías
- Detección
    - Órganos, regiones, etc.
    - Objeto y/o lesiones
- Segmentación
    - Órganos y subestructuras
    - Tipos de lesiones
- Registros de imágenes y combinación con informes

También los autores comentan las múltiples áreas de aplicación de estas técnicas digitales como por ejemplo cerebro, ojos, tórax, patología y microscopia, mamas, corazón, abdomen, y musculoesquelético.

1.2 Revisión de la literatura más actual y definición del problema

En este epígrafe repasaremos muy brevemente la tipología de hallazgos de imágenes patológicas que podemos encontrar en las radiografías de tórax. A continuación, se describirán las dos principales bases de datos públicas para el entrenamiento del modelo y finalmente repasaremos las principales arquitecturas de redes neuronales convolucionales profundas. Para ello nos basaremos en las publicaciones científicas más recientes.

1.2.1 Definición de hallazgos patológicos en la radiografía de tórax:

Para la definición de los conceptos acudiremos al *Diccionario Médico* (*Clínica Universidad de Navarra,* 2019) y *Diccionario de la lengua española* (*Edición del Tricentenario,* 2019):

- **Atelectasia:** “Colapso pulmonar”.
- **Cardiomegalia:** “Aumento del tamaño cardiaco por hipertrofia o dilatación”.
- **Condensación:** “Aumento de la densidad o la capacidad de atenuación de los rayos X en un órgano al acumularse líquido en zonas donde normalmente existe densidad de aire”.
- **Derrame:** “Vertido de un líquido o de una exudación”.
- **Edema:** “Aumento patológico del líquido intersticial. Produce hinchazón localizada o difusa, resultante del acúmulo del componente extravascular del líquido extracelular en un determinado órgano o tejido”.
- **Enfisema:** “Enfermedad pulmonar obstructiva crónica con distensión de los espacios aéreos distales a los bronquiolos terminales y destrucción de los tabiques alveolares, lo que implica una pérdida de elasticidad pulmonar, de tal forma que el aire queda atrapado al final de la espiración”.
- **Engrosamiento Pleural:** “Engrosamiento de la membrana que recubre los pulmones”.
- **Fibrosis:** “Aumento patológico del tejido conjuntivo en algún órgano o tejido”.
- **Hernia:** “Protrusión o salida de parte de un órgano de la estructura anatómica que normalmente la fija”.
- **Infiltración:** “Difusión o acumulación en un tejido, en particular el tejido conjuntivo, de alguna substancia o estructura celular que le es extraña o en cantidades excesivas con respecto a lo normal”.
- **Masa:** “Genéricamente, hinchazón bulto o tumefacción”.
- **Neumonía:** “Inflamación aguda del parénquima pulmonar en la que los alveolos y bronquiolos se taponan por el acúmulo de un exudado fibrinoso”.
- **Neumotórax:** “Presencia de aire dentro de la cavidad pleural, que provoca un colapso del pulmón”.
- **Nódulo:** “Pequeña eminencia o vegetación, nudosidad”.

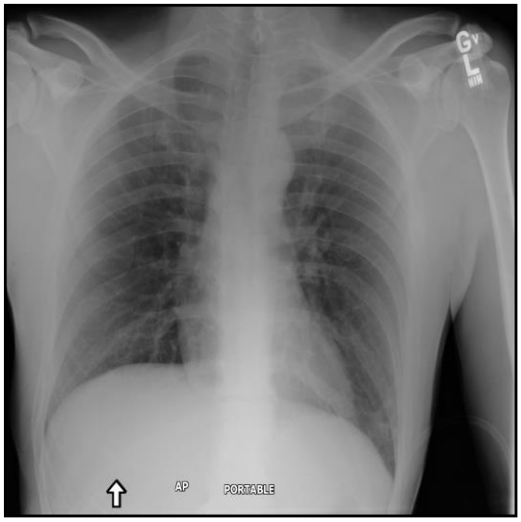


Figura 2: Ejemplo de radiografía de tórax con una infiltración.
(*ChestXray-NIHCC*, 2018)

La figura 2 muestra un ejemplo de radiografía de un paciente con una de las patologías antes mencionadas.

1.2.2 Bases de datos de *National Institute of Health*

La segunda base de datos de radiografías de tórax (antero-posterior y postero-ranterior) más grande de la comunidad científica es la proporcionada por el *National Institute of Health* (*NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community | National Institutes of Health (NIH)*, 2017). Esta base de datos contiene aproximadamente 112,000 radiografías y 30,000 pacientes. Una radiografía antero-posterior es aquella en la que el paciente está situado de frente a la parte generadora o a la fuente de energía, en el momento de la adquisición de una imagen; una radiografía postero-anterior es aquella en la que el paciente está situado de espalda a la parte generadora o fuente de energía (*Diccionario médico. Clínica Universidad de Navarra*, 2019). Existen dos versiones (ChestNet 8 y ChestNet 14) y dependiendo de la fecha de publicación de los artículos se usa una u otra. La principal diferencia entre estas bases de datos es el número diferente de hallazgos patológicos etiquetados.

Como veremos más adelante nosotros utilizaremos una muestra de la ChestNet 14 (*NIH Chest X-ray Dataset Sample*, 2017). Esto se debe a que el modelo se realizará en un equipo sin GPUs y, por tanto, hay que limitar el tamaño de los datos.

1.2.2.1 Métodos de etiquetado de las radiografías y problemas frecuentes

Las radiografías de las bases de datos del punto anterior van acompañadas de su informe médico. Para poder etiquetarlas los autores escanearon dichos informes utilizando algoritmos de procesamiento de lenguaje natural (NLP) (Wang *et al.*, 2017, 2018) etiquetando cada radiografía con los correspondientes nombres de hallazgos patológicos (o si no hay ningún hallazgo patológico, una etiqueta que indique la ausencia de ellos). Hay que tener en cuenta que en una misma imagen puede haber varios hallazgos patológicos diferentes, por lo que una misma imagen puede tener varias etiquetas (imágenes multietiquetadas).

En la imagen siguiente podemos observar un diagrama que muestra las proporciones de las etiquetas.

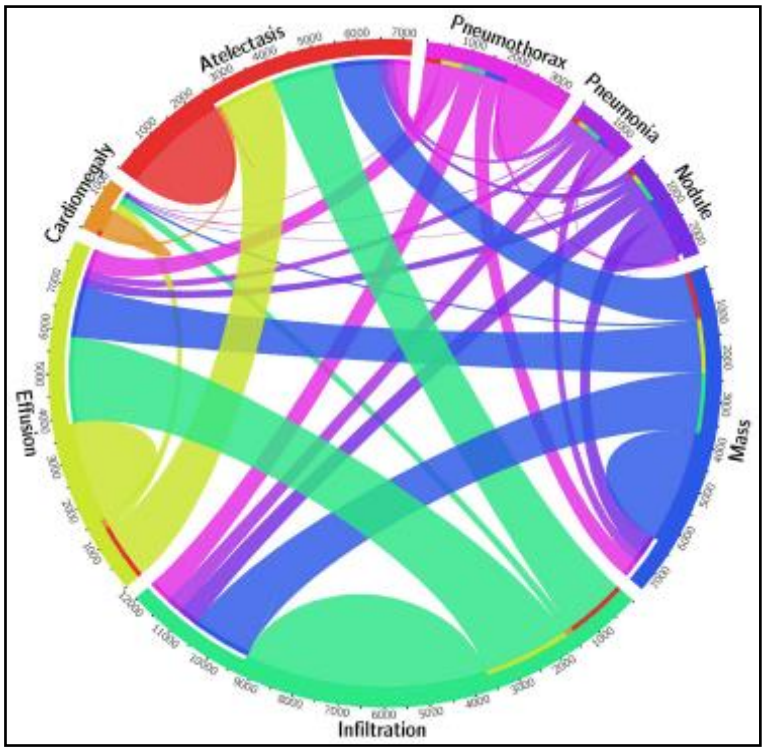


Figura 3: Proporciones de los hallazgos patológicos en las imágenes (Wang *et al.*, 2017)

El proceso de etiquetado realizado por Wang y sus compañeros no fue trivial, ya que, primero tuvieron que normalizar el vocabulario médico y después utilizaron los algoritmos MetaMap (Aronson y Lang, 2010) y DNorm (Leaman, Khare y Lu, 2015), que son comúnmente usados en estos problemas de medicina. DNorm en un algoritmo de *machine learning* para el reconocimiento y la normalización de enfermedades. MetaMap detecta

bioconceptos del corpus de un texto biomédico. Los autores usaron una combinación de ambos mejorada.

En la siguiente tabla se muestra una comparativa de los resultados de etiquetados usando MetaMap y el método mejorado.

	MetaMap			**MetaMap + Dnorm**		
Patología	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Atelectasia	0.95	0.95	0.95	0.99	0.85	0.91
Cardiomegalia	0.99	0.83	0.90	1.00	0.79	0.88
Derrame	0.74	0.90	0.81	0.93	0.82	0.87
Infiltración	0.25	0.98	0.39	0.74	0.87	0.80
Masa	0.59	0.67	0.62	0.75	0.40	0.52
Nódulo	0.95	0.65	0.77	0.96	0.62	0.75
Normal	0.93	0.90	0.91	0.87	0.99	0.93
Neumonía	0.58	0.93	0.71	0.66	0.93	0.77
Neumotórax	0.32	0.82	0.46	0.90	0.82	0.86
Total	0.84	0.88	0.86	0.90	0.91	0.90

Tabla 1: Comparativa de resultados de etiquetado (Wang *et al.*, 2017)

Como vemos en la tabla anterior, el método combinado mejora sustancialmente la precisión del etiquetado automático de la mayoría de los hallazgos, especialmente hallazgos como masa, derrame, neumotórax, infiltración y neumonía. En cambio, si se miran las métricas de *Recall* y *F1-Score* vemos como empeoran los resultados. Esto último puede ser un problema en el caso que los datos no estuvieran balanceados ya que el modelo que etiqueta las imágenes estaría discriminando peor en el modelo mejorado que en MetaMap.

Ahora bien, este etiquetado puede contener errores, ya que, no dejan de ser diagnósticos de profesionales médicos y, por tanto, debido al error humano las conclusiones médicas pueden no ser perfectas, a diferencia con lo que pasaría por ejemplo con el etiquetado de otro tipo de imágenes como pueden ser el de animales u objetos cotidianos (Oakden-Rayner, 2017).

1.2.2.2 Pre-procesado de imágenes y *Bounding Box*

Las imágenes comunes de radiografía suelen tener una resolución de 3000 x 2000 píxeles, pero debido a que con esta resolución sería necesaria una capacidad de computación muy elevada, las imágenes se reducen a 1024 x 1024 píxeles con la consiguiente pérdida de detalle.

Además, estas bases de datos también contienen una tabla donde están las coordenadas del área dentro de las cuales encuentran los hallazgos en cada imagen. Estos *bounding box* existen solo para algunas radiografías (en las que el informe incluye la anotación de la posición de la patología).

1.2.3 Arquitecturas más utilizadas

El artículo titulado *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases* (Wang *et al.*, 2017) hace una revisión de las arquitecturas de redes neuronales convolucionales más comunes. Para ello utilizan las siguientes arquitecturas pre-entrenadas con *Imagenet* (Deng *et al.*, 2009): AlexNet (Krizhevsky, Sutskever y Hinton, 2012), GoogLeNet (Szegedy *et al.*, 2018), VGGNet-16 (Simonyan y Zisserman, 2014) y ResNet-50 (He *et al.*, 2015). Para conectar estas arquitecturas a la etapa *Fully Connected Network* se usa una capa de *Global Pooling* (Lin, Chen y Yan, 2014).

En las siguientes figuras se muestran las arquitecturas antes mencionadas.

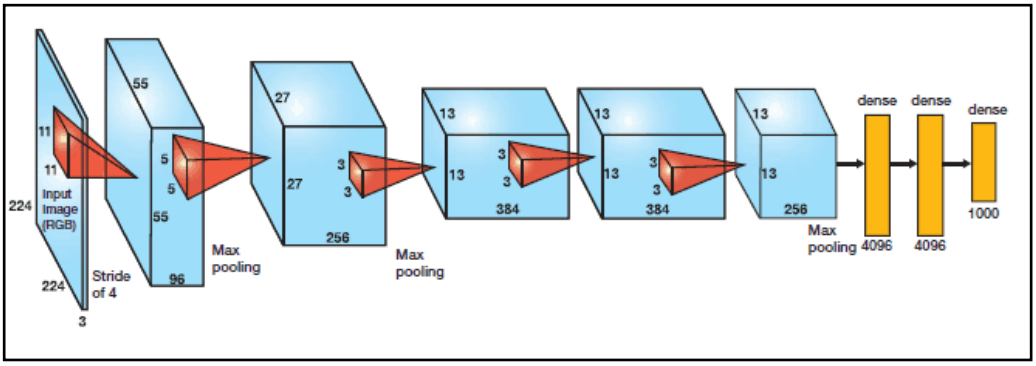


Figura 4: AlexNet (*ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks – CV-Tricks.com*, 2018)

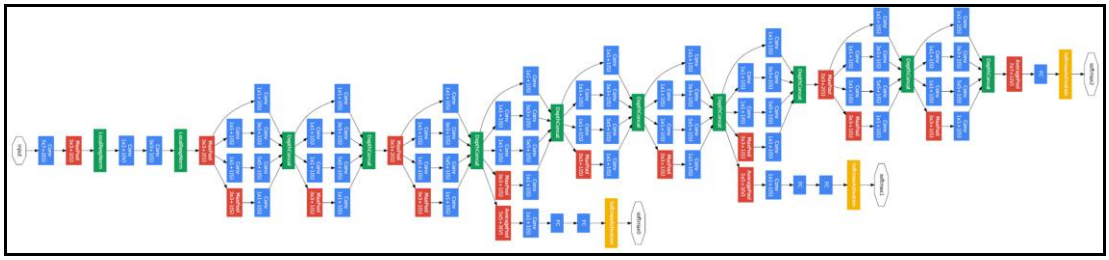


Figura 5: GoogLeNet (*Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 Image Classification*, 2018)

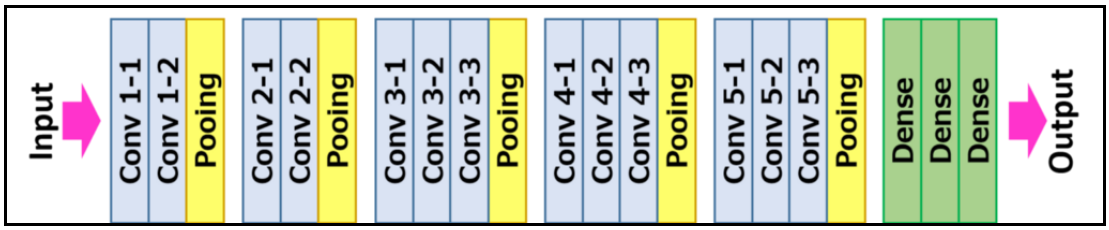


Figura 6: VGGNet 16 (*CNN Architectures — LeNet, AlexNet, VGG, GoogLeNet and ResNet – mc.ai*, 2018)

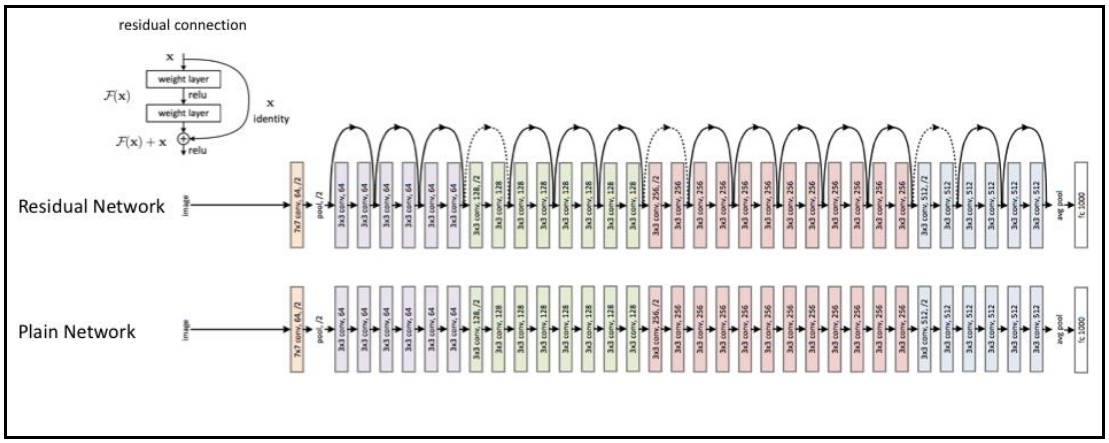


Figura 7: ResNet 50 (*Common architectures in convolutional neural networks*, 2018)

Cada una de estas arquitecturas presentaba una serie de avances en el momento de su creación. En AlexNet podemos destacar el uso de la función de activación ReLU (Unidad Lineal Rectificada) en vez de las funciones más tradicionales como Tangente Hiperbólica o Sigmoidal. Por su parte, VGGNet es una mejora de AlexNet. Debido a los avances informáticos, como por ejemplo el uso intensivo de las GPU, ha habido grandes avances desde estas arquitecturas, tanto en profundidad de la red como en complejidad de la arquitectura. Las redes Inception (en sus distintas versiones) y ResNet son un claro ejemplo (*ResNet*, *AlexNet*,

VGGNet, Inception: Understanding various architectures of Convolutional Networks – CV-Tricks.com, 2018).

Como se muestra en la tabla siguiente, en general GoogLeNet y ResNet 50 obtienen los mejores resultados en la mayoría de los casos:

	AlexNet	**GoogLeNet**	**VGGNet-16**	**ResNet-50**
Atelectasia	0.65	0.63	0.63	**0.71**
Cardiomegalia	0.69	0.71	0.71	**0.81**
Derrame	0.66	0.69	0.65	**0.74**
Infiltración	0.60	**0.61**	0.59	**0.61**
Masa	**0.56**	0.54	0.51	**0.56**
Nódulo	0.65	0.56	0.66	**0.72**
Neumonía	0.55	0.60	0.51	**0.63**
Neumotórax	0.74	0.78	0.75	**0.79**

Tabla 2: Comparativa de AUC de arquitecturas preentrenadas. (Wang *et al.*, 2017)

1.2.4 Definición del problema y motivación

Al tratarse de un Proyecto de Fin de Máster, limitaremos el estudio a la detección de los hallazgos patológicos de masas y nódulos. El objetivo será crear un modelo de red neuronal que detecte positivos. Los positivos serán aquellas radiografías que puedan contener una masa y/o un nódulo. Para la selección de estas patologías hemos contado con el asesoramiento del Jefe de Servicio de Neumología de un hospital español de reconocido prestigio (no se revela su nombre por cuestiones de confidencialidad).

La principal motivación, para la realización de este proyecto, es la necesidad de mejorar la detección temprana de este tipo de enfermedades, ya que, si el cáncer es detectado con prontitud, los tratamientos tienen una mayor efectividad (Wang, 2017). Además, este procedimiento podría alertar y evitar errores humanos debido a la fatiga del radiólogo y la falta de especialización y experiencia en algunos países en vías de desarrollo. Algoritmos de este tipo pueden ser un gran apoyo al facultativo, y de ningún modo pretenden ser un sustituto de éste.

1.2.5 Objetivos

- Probar varias arquitecturas de redes neuronales convolucionales usando *transfer learning*.
- Intentar encontrar los mejores hiper-parámetros que configuren las redes.
- Realizar un benchmarking de los modelos de redes neuronales frente a otros modelos de Machine Learning evaluando el poder discriminante mediante la curva ROC y el área bajo ésta (AUC).
- Seleccionar un modelo Deep Learning que constituya nuestra prueba de concepto.

2 Descripción de la técnica y resultados

2.1 Los datos del NIH

2.1.1 Breve descripción de los datos

Como ya comentamos en el primer epígrafe, vamos a utilizar una muestra aleatoria de 5,606 observaciones escogidas aleatoriamente de ChestNet 14 (*NIH Chest X-ray Dataset Sample*, 2017). De esa muestra, 553 imágenes presentan hallazgo de nódulo y/o masa, lo que supone un 9.86% de nuestras observaciones, es decir, nuestros datos no están balanceados. Si a esto sumamos que tenemos un tamaño de muestra pequeño, será difícil obtener unos buenos resultados en nuestros modelos.

Aprovechamos este punto para recordar al lector que nuestro objetivo es encontrar un modelo que servirá como prueba de concepto y permitirá conocer mejor las particularidades de este tipo de datos. Además, se intentará encontrar la mejor arquitectura de redes convolucionales para resolver nuestro problema. Como comentaremos al final de nuestro trabajo (y ahora adelantamos), en un futuro se contará con equipos más potentes para poder usar la base de datos completa y, además, trataremos de ser más ambiciosos y construir

modelos que predigan todos los diferentes hallazgos patológicos que contiene esta base de datos.

A continuación, mostramos dos gráficos que nos permiten conocer mejor cómo es nuestra base de datos.

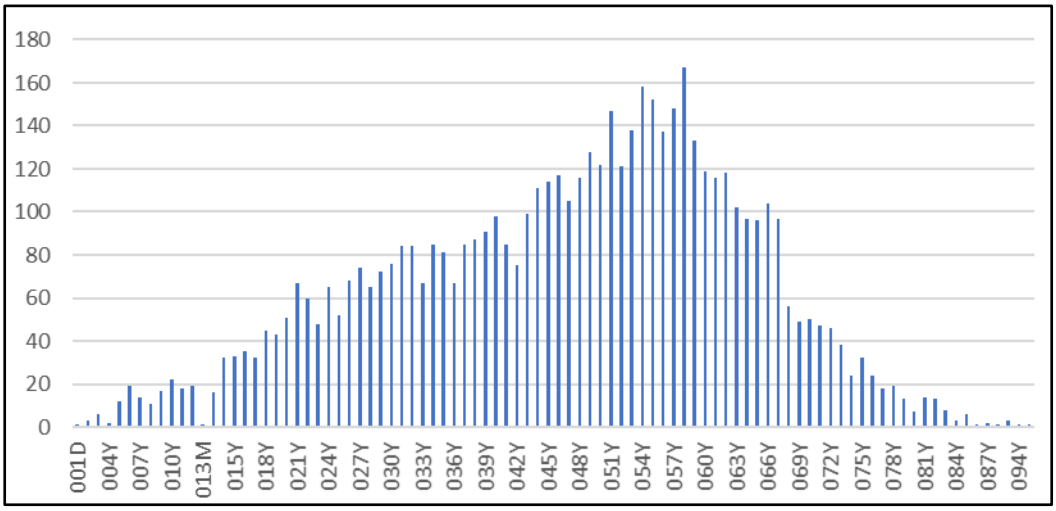


Figura 8: Distribución de radiografías por edades

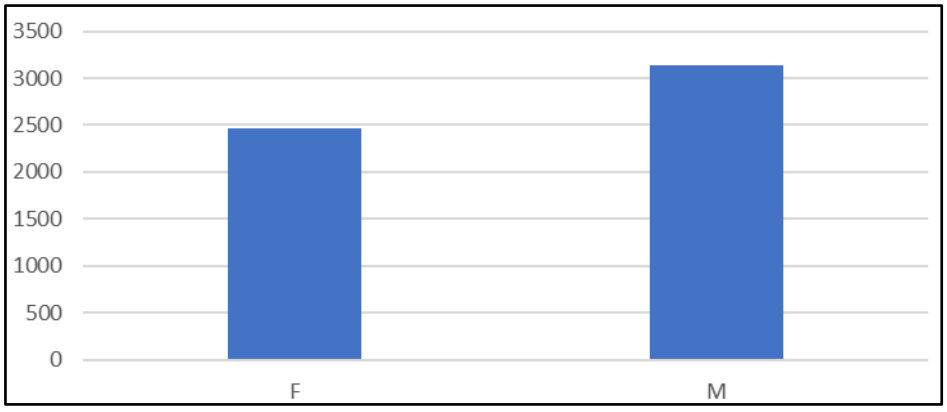


Figura 9: Distribución de radiografías por sexo

Edad Media	46.7 años
Edad Mínima	1 día
Edad Máxima	94 años
Mujeres	44%
Hombres	56%
Imágenes Antero-posterior	39%
Imágenes Postero-anterior	61%
Nº Raidiografias media por paciente	2.7
Tiempo medio entre radiografías	1.7 años

Tabla 3: Características de las imágenes

Como se puede observar, tanto en las dos gráficas anteriores como en la tabla, tenemos una gran amplitud de pacientes: desde un neonato hasta un paciente de 94 años, aunque la mayor parte se encuentran entre los 40 y 70 años. Además, el sexo de los pacientes está bastante equilibrado.

Para realizar nuestro modelo, los datos se dividirán en tres conjuntos: entrenamiento, validación y test. Además, como los datos están ordenados por paciente, se construirán los tres conjuntos evitando que un mismo paciente esté en más de uno.

2.1.2 Tratamiento del desbalanceo de clases

Debido al gran desbalanceo en las clases, pueden existir problemas en el entrenamiento, ya que, el modelo asignará más peso a las etiquetas mayoritarias y, por lo tanto, tenderá a ignorar a las minoritarias. Aunque Keras tiene la opción de ajustar los pesos de las clases, para que todas tengan la misma importancia independientemente de sus proporciones, hemos preferido balancear las clases utilizando la técnica del *oversampling* por repetición. Esto es: remuestrear las observaciones que poseen hallazgos de la clase minoritaria (masa/nódulo en nuestro caso). Otra opción hubiera sido eliminar observaciones del caso mayoritario, pero al tener un conjunto pequeño de datos hemos preferido la primera opción.

Con el fin de no contaminar los diferentes conjuntos, tanto el conjunto de entrenamiento como el de validación se remuestrean de manera independiente. No se remuestrea el conjunto de test.

2.2 Breve Introducción a las Redes Convolucionales y al *Transfer Learning*

2.2.1 Redes convolucionales

Las redes convolucionales o convolutivas están basadas en la operación matemática de convolución (Berzal, 2018):

$$(f * g) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)d\tau$$

En el caso discreto sería:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$

En el análisis de imágenes, una convolución funciona moviendo ventanas de tamaño $n \times n \times 3$ (tensor de pesos) sobre la entrada en 3D (por ejemplo, los 3 canales RGB). En el caso que nosotros contemplamos las imágenes son en 2D, ya que, solo tenemos un canal de color (escala de grises) y, por tanto, las ventanas son de tamaño $n \times n \times 1$.

La ventana se mueve una distancia *d* en cada paso, y va extrayendo características de los datos. Cada tensor se transforma, a través de un producto tensorial por el tensor de pesos (núcleo de convolución), en un vector 1D. La longitud del vector 1D conforma el número de filtros (profundidad de salida). Todos estos vectores son luego unidos espacialmente en una capa de salida en 3D, de dimensiones altura × ancho × profundidad de salida (Chollet, 2017). La siguiente figura muestra de una forma esquemática, y muy intuitiva, la operación de convolución.

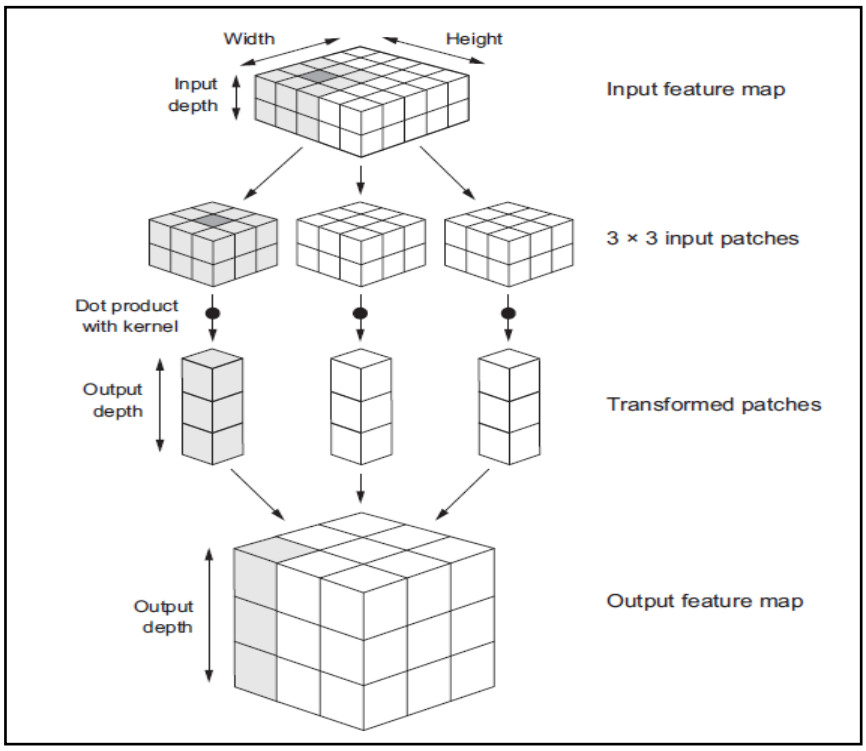


Figura 10: Esquema de operación de convolución (Chollet, 2017)

Una de las características que diferencian la capa convolucional de otras capas es que no todas las entradas están conectadas con todas las salidas. Según Berzal (2018):

> *Cada neurona individual de una capa convolutiva opera sobre un subconjunto de las entradas (un fragmento de la señal de voz o una región de la imagen). Por tanto, cada neurona utilizada detecta una característica local de la señal de entrada. En el caso de las imágenes, cada neurona detecta una característica en una región diferente de la imagen. Ahora bien, si una de las neuronas consigue detectar una característica útil en una región particular de la imagen, tiene sentido que se aproveche esa característica aprendida para que la red sea capaz de detectar la misma característica en otras regiones de la imagen. Por este motivo, se utilizan detectores replicados para que la red sea capaz de detectar una característica particular en cualquier posición de la imagen. Es decir, se utilizan los mismos pesos (la misma máscara de convolución) en neuronas diferentes cuyas entradas están conectadas a diferentes regiones de la imagen.*

Una capa convolucional tiene los siguientes hiper-parámetros:

- Profundidad: número de filtros que tiene la capa

- Conectividad local: tamaño del filtro (e.g. 3x3, 5x5 y 7x7)
- Paso (*stride*): cuántas celdas "saltan" cada filtro
- Relleno con ceros (*zero padding*): se rellenan los bordes con cero de la capa de salida para preservar su tamaño

Generalmente, en las redes neuronales convolucionales se suelen utilizar varias capas donde se realiza esta operación, que extrae las características, y después suele haber una o varias capas densas completamente conectadas (*Fully Connected Network*) antes de la capa de salida.

A las capas convolucionales se le añaden capas de *pooling*. Una capa de *pooling* consiste en "reducir el tamaño de su entrada para que las capas posteriores puedan trabajar con datos de entrada reducidos" (Berzal, 2018). De esta forma se consigue una menor carga computacional en el entrenamiento y una pequeña invarianza frente a translaciones de la entrada. Este tipo de capas suelen ir entre las capas convolucionales.

A continuación, vemos una arquitectura muy simple de red convolucional

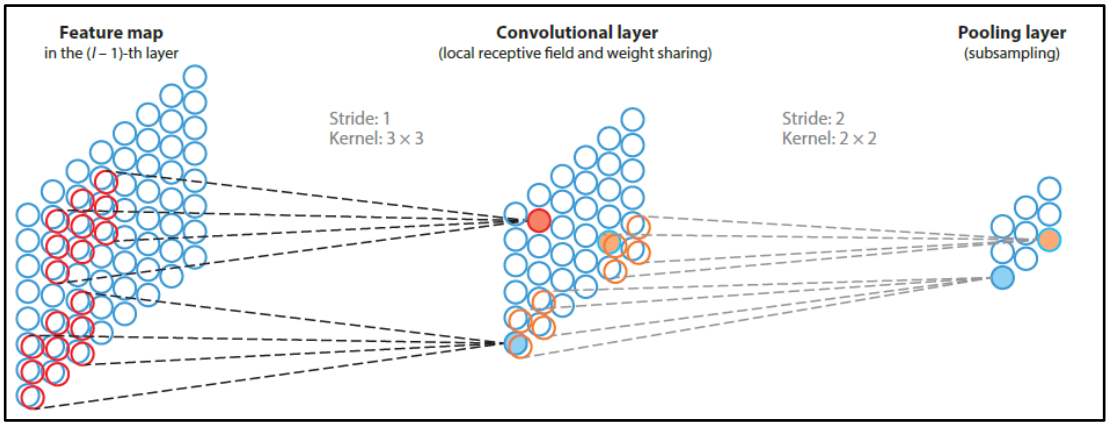


Figura 11: Esquema de red neuronal convolucional (Shen, Wu y Suk, 2017)

Además de estas dos capas características de las redes convolucionales, se les añade capas de *dropout* y *batch normalization*. La capa de *dropout* (Srivastava *et al.*, 2014) se utiliza para evitar el *overfitting*, es un método de regularización. Consiste en apagar un porcentaje aleatorio de las unidades en cada paso del entrenamiento. Los pesos de las unidades apagadas no se actualizan en ese paso. La capa de *batch normalization* (Ioffe y Szegedy, 2015) consiste en normalizar las salidas de las capas ocultas para evitar el *covariate shift*.

Existen estructuras muy complejas donde la arquitectura en vez de ser lineal tiene varias ramificaciones, como por ejemplo GoogLeNet (Szegedy *et al.*, 2018).

2.2.2 El *transfer learning*

Según Berzal (2018), el *transfer learning* se define como "reutilización de características extraídas para resolver un problema en la resolución de un problema diferente ".

El uso del *transfer learning* dependerá del tamaño del conjunto de datos a entrenar y/o de la similitud entre los conjuntos de datos originales y los de destino (Karim, Sewak y Pujari, 2018).

Además, los autores sostienen que hay cuatro casos de uso principales:

- **Caso 1:** El nuevo conjunto de datos (destino) es pequeño y es similar al original conjunto de datos de entrenamiento
- **Caso 2:** El nuevo conjunto de datos (destino) es pequeño pero diferente del original conjunto de datos de entrenamiento
- **Caso 3:** El nuevo conjunto de datos (de destino) es grande y es similar al original conjunto de datos de entrenamiento
- **Caso 4:** El nuevo conjunto de datos (destino) es grande y diferente del original conjunto de datos de entrenamiento

Nuestro caso se correspondería con el caso 2 y por tanto procede a realizar, como los autores indican:

- Separar la mayoría de las capas iniciales de la red
- Añadir a las capas pre-entrenadas una nueva capa completamente conectada que coincide con el número de clases del conjunto de datos de destino
- Aleatorizar los pesos de la nueva capa completamente conectada y congelar todos los pesos de la red pre-entrenada
- Entrenar la red para actualizar solamente los pesos de la nueva capa completamente conectada

Imagenet (Deng *et al.*, 2009; Krizhevsky, Sutskever y Hinton, 2012) es una de las bases de imágenes más utilizadas para entrenar redes, que luego se reutilizarán usando *transfer learning* .

Chollet (2017) propone dos métodos para utilizar *transfer learning*:

- El primero de ellos consiste en calcular la predicción de la red convolucional utilizando los pesos del pre-entrenamiento . Las predicciones de esta red serán los valores de entrada de la red densa que más adelante construiremos.
- El otro método consiste en construir un modelo donde algunas de las capas se usen los pesos pre-entrenados y en otras los pesos se ajustarán fruto del entrenamiento de la red.

El *transfer learning* es muy útil porque nos permite usar modelos (o parte de ellos) que han sido entrenados en máquinas más potentes que las tradicionales, con arquitecturas más complejas y con conjuntos de entrenamientos muy grandes.

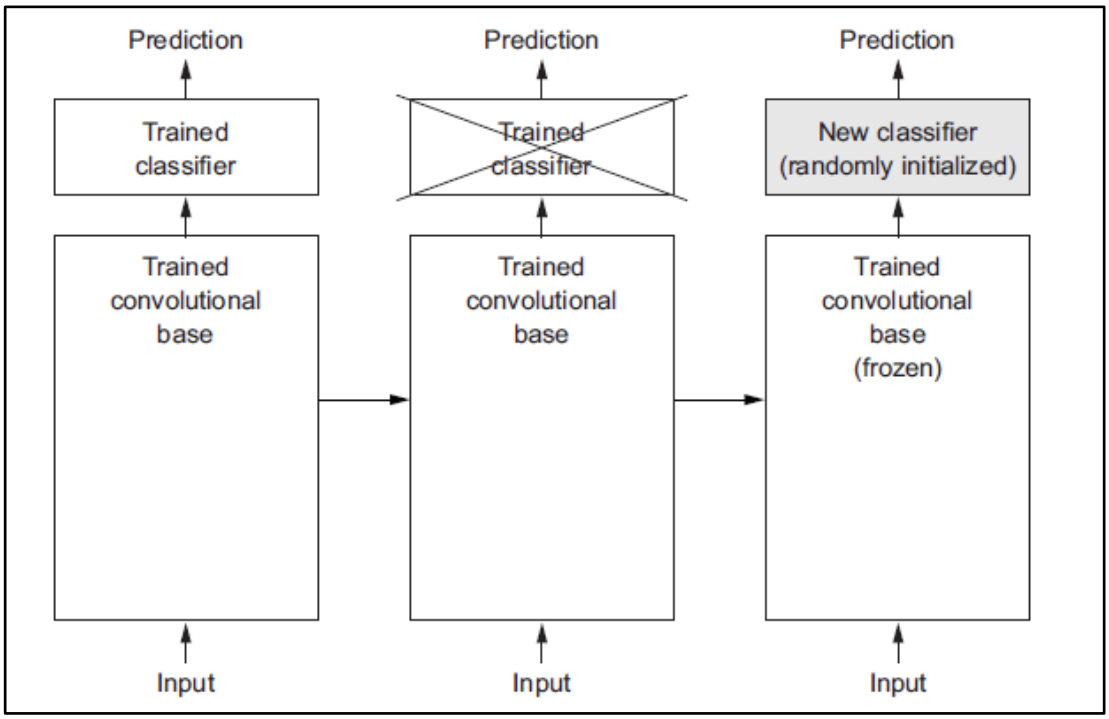


Figura 12: *Transfer Learning* para extracción de características (Chollet, 2017)

Como veremos más adelante, usaremos esta técnica, y, por lo tanto, sólo haremos un entrenamiento parcial, es decir, solo entrenaremos las capas finales: las densas y la de salida.

Hemos optado por esta técnica debido a que estamos usando un ordenador portátil que no dispone de GPUs y, por tanto, entrenar una red desde cero no es viable. Además, aunque pudiéramos entrenar la red desde cero, la extracción de características aprendidas por la red seguramente no sería muy buena debido al pequeño tamaño de la muestra. Por tanto, consideramos que es mejor usar la extracción de características generales que proporciona un modelo entrenado con *Imagenet*. Por otra parte, como hemos visto, nuestros datos son altamente desbalanceados (las imágenes que presentan hallazgos con masas y/o nódulos son aproximadamente el 10% de la muestra), lo que dificulta aún más la extracción de características y por tanto justifica más el uso de una técnica como el *transfer learning*.

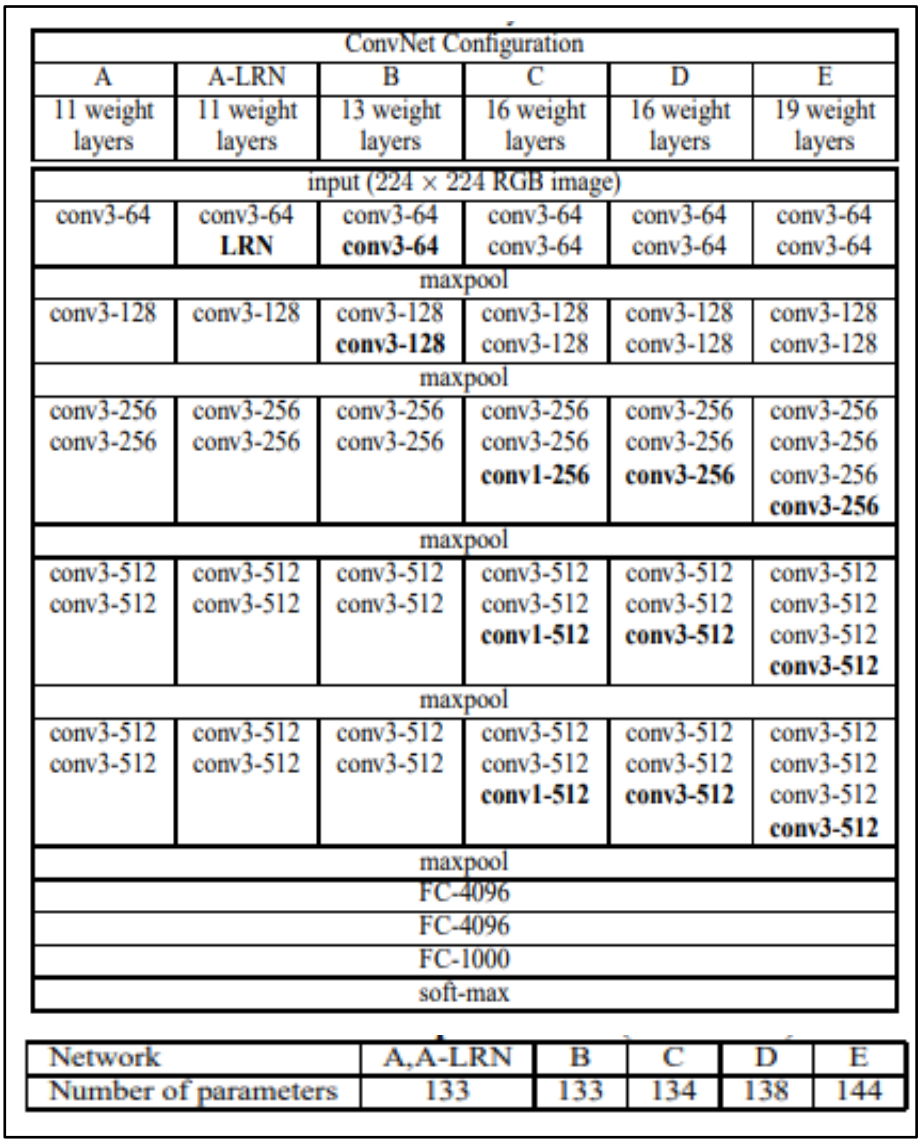
2.3 Arquitecturas revisadas y entrenamiento

En este proyecto se han utilizado dos arquitecturas: VGGNet y ResNet. Se ha apostado por estas arquitecturas fundamentalmente por dos criterios: el primero corresponde a los buenos resultados dados en el trabajo de Wang *et al.* (2017); el segundo corresponde, nuevamente, a consideraciones computacionales.

2.3.1 VGGNet 16 y 19

Aunque esta arquitectura tiene distintas configuraciones, tal como se muestra en el artículo de Simonyan and Zisserman (2014), nosotros hemos optado por la de 16 y 19 capas. Ambas se diferencian en añadir o no una capa convolucional en los bloques 3, 4 y 5.

Como se muestra en la figura siguiente, la VGGNet presenta distintas configuraciones en función de la cantidad de capas convolucionales que presenta cada bloque. En este proyecto se han utilizado las configuraciones D y E.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Figura 13: Esquemas de VGGNet-16 (Simonyan y Zisserman, 2014)

En la siguiente imagen se muestra el esquema utilizado en el proyecto para el caso de la VGGNet-16.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 256, 256, 3)	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0

Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0

Figura 14: Esquema VGGNet-16 utilizado

2.3.2 ResNet-50

En el caso de Resnet se ha optado por la arquitectura de 50 capas. En este caso no reproducimos la imagen debido al tamaño de la arquitectura y nos remitimos a la figura 7 y a los notebooks anexos en el soporte electrónico.

2.3.3 Capas Densas (*Fully Connected Network*)

Después de extraer las características con la arquitectura pre-entrenada, entrenamos unas capas densas completamente conectadas. Las capas convolucionales se unirán a las densas por medio de una capa de *Global Average Pooling*. A continuación, vemos la siguiente imagen con su esquema y comentaremos detalladamente esta parte de la arquitectura.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dropout_17 (Dropout)         (None, 512)               0
_________________________________________________________________
batch_normalization_25 (Batc (None, 512)               2048
_________________________________________________________________
dense_25 (Dense)             (None, 30)                15390
_________________________________________________________________
dropout_18 (Dropout)         (None, 30)                0
_________________________________________________________________
batch_normalization_26 (Batc (None, 30)                120
_________________________________________________________________
dense_26 (Dense)             (None, 20)                620
_________________________________________________________________
batch_normalization_27 (Batc (None, 20)                80
_________________________________________________________________
dense_27 (Dense)             (None, 1)                 21
=================================================================
Total params: 18,279
Trainable params: 17,155
Non-trainable params: 1,124
_________________________________________________________________
None
```

Figura 15: Esquema MLP utilizado

La arquitectura está formada por dos capas ocultas y una capa de salida. La primera capa densa está formada por 30 unidades, la segunda capa densa por 20. Al ser un problema de clasificación binario la capa de salida está formada por una unidad con función de activación sigmoidal. La función de activación en las capas ocultas es Relu. Aunque se han probado otras funciones de activación, esta es la que ha dado mejores resultados. Además, para evitar *covariate shift* y *overftting* se han utilizado capas de *batch normalization y drop out*.

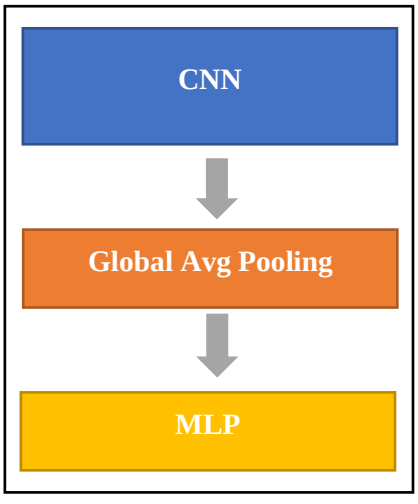


Figura 16: Esquema del modelo

2.3.4 Entrenamiento

Para entrenar las últimas capas de la red se ha utilizado el método de optimización de Adam (Kingma y Ba, 2014). Hay que destacar que en lugar de usar una sola llamada al método de *Keras "fit"* se ha utilizado una variación del método que permite en cada paso guardar los pesos del modelo siempre y cuando haya una mejora en el mínimo de la función de coste sobre los datos de validación:

```
%%time
LOAD_MODEL = False
monitor = 'loss'
mode = 'min'
if not LOAD_MODEL:
    epochs = 2000
    batch_size = 128
        import pickle
    filepath="best_model.h5"
    filepath_2="last_model.h5"
        checkpoint = ModelCheckpoint(filepath, monitor='val_'+monitor,
                                     verbose=1, save_best_only=True,
                                     mode=mode) # graba sólo los que mejoran en validación
    checkpoint_2 = ModelCheckpoint(filepath_2, monitor='val_'+monitor, verbose=1,
                                     save_best_only=False,
                                     mode=mode) # graba el último
        callbacks_list = [checkpoint, checkpoint_2]
    for e in range(epochs):
        print("Epoca: ", e+1)
        inds_tr=np.random.permutation(len(X_tr_sc))
        inds_tr_va=np.random.permutation(len(X_va_sc))
                    history = modelo_desacoplado.fit(X_tr_sc[inds_tr], y_tr[inds_tr],
                                        batch_size=batch_size, epochs=1, callbacks=callbacks_list,
                                        verbose=1,
                                        validation_data=(X_va_sc[inds_tr_va], y_va[inds_tr_va]),
                                        class_weight=class_weights )
```

Figura 17: Entrenamiento de las capas densas

2.4 Modelos de *benchmarking*

Para poder comparar nuestro modelo final se construyen dos modelos más sencillos:

- Regresión Logística (Cox, 1958)
- *Random Forest* (Ho, 1995)

En estos dos modelos se usan las configuraciones que vienen por defecto en la librería de Python *Scikit-Learn* (más adelante veremos una mejora de estos modelos usando *Grid Search*). Estos modelos se conectan a la salida de la red convolucional y sustituyen por tanto a las capas densas.

2.5 Modelo final y Resultados

A la vista de los resultados siguientes creemos que el mejor modelo es la VGGNet-16 con un módulo final de capa densa.

Para evaluar utilizaremos las siguientes medias de *performance*:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \qquad Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN} \qquad F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

Y el área bajo la curva, AUC, de la curva ROC.

		VGGNet-16		VGGNet-19		ResNet-50	
		0	**1**	**0**	**1**	**0**	**1**
Regresión Logística	Score Avg.	0.74		0.74		0.7	
	Precision	0.88	0.1	0	0.1	0.9	0
	Recall	0.02	0.97	0	1	1	0
	F1-Score	0.04	0.18	0	0.18	0.95	0
	AUC	0.59		0.55		0.51	
Random Forest	Score Avg.	0.9		0.9		0.9	
	Precision	0.9	0	0.9	0	0.9	0
	Recall	1	0	1	0	1	0
	F1-Score	0.95	0	0.95	0	0.95	0
	AUC	0.54		0.51		0.52	
Red	Accuracy Avg.	0.64		0.49		0.44	
	Precision	0.93	0.16	0.92	0.12	0.91	0.11
	Recall	0.65	0.59	0.48	0.66	0.43	0.61
	F1-Score	0.76	0.25	0.63	0.21	0.58	0.18
	AUC	0.62		0.63		0.56	

Tabla 4: Resultado de los modelos en la muestra de test

Aunque aparentemente, si miráramos solamente el *accuracy*, podríamos decir que los modelos Random Forest y Regresión Logística son mejores que la red, hay que tener en cuenta que este valor es medio. Al estar los datos desbalanceados esto indica que el modelo prácticamente predice que no hay nódulo o masa, es decir, no discrimina prácticamente nada. Por ello, las siguientes medidas *precision*, *recall* y *F1*, se calculan independientemente para ambos valores del *target*.

La red neuronal, a pesar de tener un *accuracy* peor que el resto de los modelos, tiene mejores resultados en las otras medidas. Si miramos la AUC podemos observar que los modelos de red discriminan mejor (la AUC es superior). La tabla anterior también nos muestra que las arquitecturas VGGNet funcionan mejor que la ResNet, habiendo una ligera mejora en la arquitectura de 16 capas frente a la de 19, lo que hace que el modelo final seleccionado sea la VGGNet-16.

A continuación, mostraremos las curvas ROC para cada arquitectura de CNN junto con las capas densas y su gráfico de distribución:

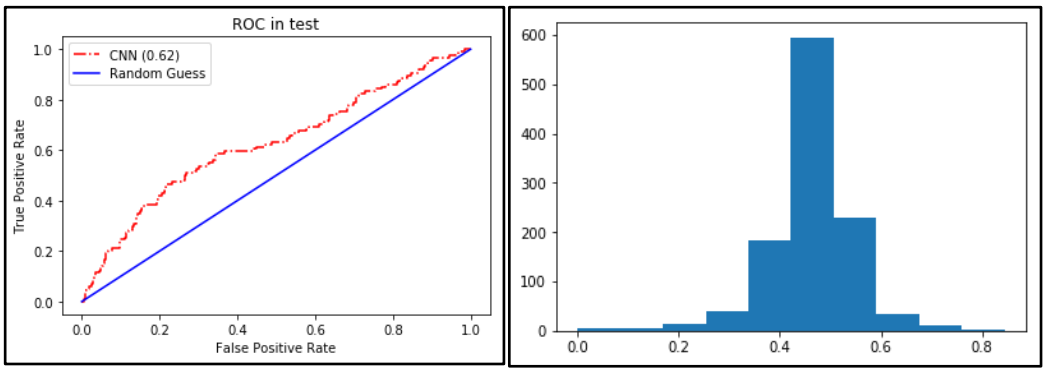


Figura 18: ROC y distribución de probabilidad de alarma en VGGNet-16

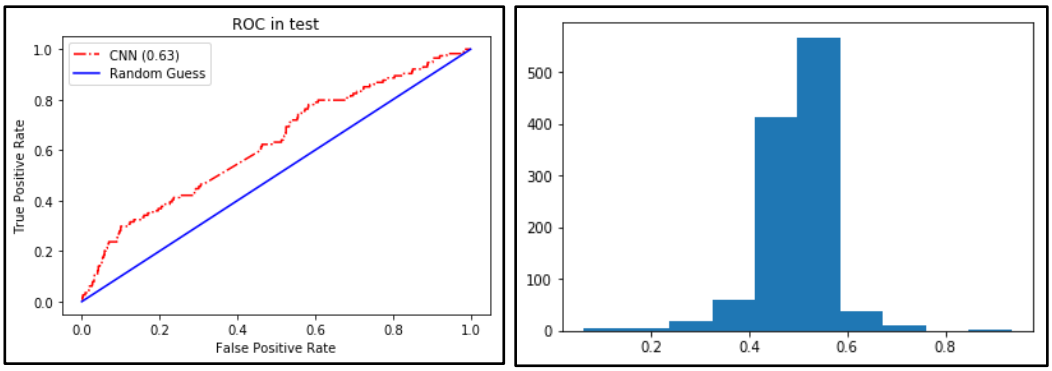


Figura 19: ROC y distribución de probabilidad de alarma en VGGNet-19

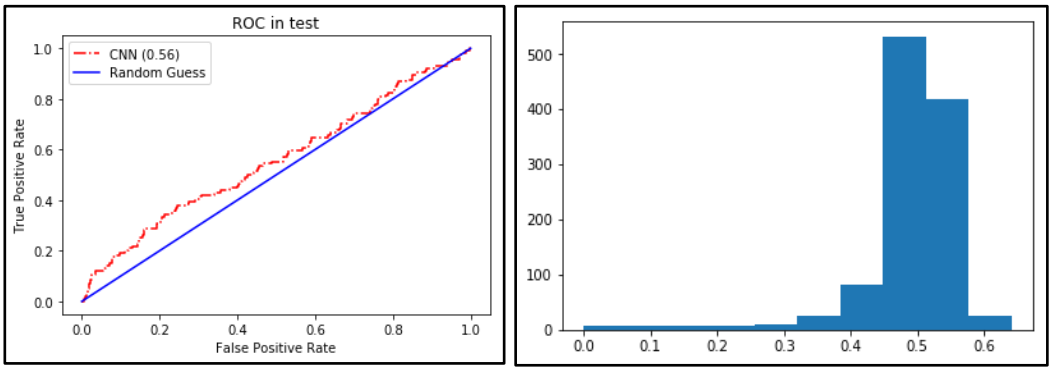


Figura 20: ROC y distribución de probabilidad de alarma en ResNet-50

Como se muestra en los gráficos previos, la arquitectura ResNet prácticamente no discrimina.

Analizando la curva ROC de la VGGNet-16 podemos decir que el punto umbral, que discrimina entre hallazgo encontrado y no encontrado, es 0.503, es decir, aquellos valores que sean superior a ese número tendrán mayor probabilidad de presentar un nódulo y/o una masa (ver tabla 5).

Max discriminación			
min distancia	thresholds	FPRate	TPRate
0.549	0.503	0.318	0.553

Tabla 5: Óptimo de la curva ROC de VGGNet-16

2.6 Sustitución de la Capa Densa por otros modelos

En muchas ocasiones se puede cambiar la capa densa por otros modelos de *machine learning*. A continuación, evaluaremos el uso de tres de los modelos más utilizados en problemas de clasificación: regresión logística, *random forest* y *gradient boosting*.

A diferencia de los modelos usados para *benchmarking*, en este caso se ha tratado de obtener los hiper-parámetros que mejor resultados daban usando *Grid Search*.

El esquema de los modelos es el siguiente

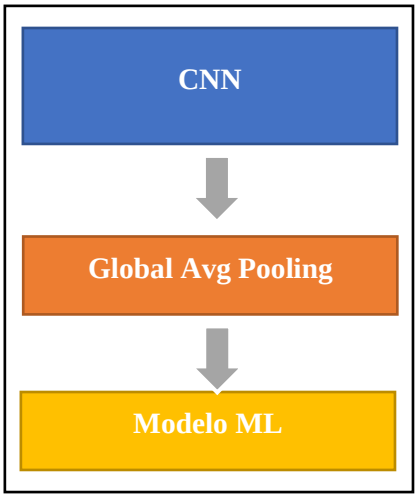


Figura 21: Esquema del modelo

2.6.1 Regresión Logística

La regresión logística (Cox, 1958), a pesar de tener un AUC inferior al modelo MLP, tiene una mejor distribución de sus predicciones, ya que, tiene una menor concentración alrededor

de 0.50, lo que hace que pueda diferenciar mejor la probabilidad de obtener un positivo (es decir, un hallazgo de masa o nódulo).

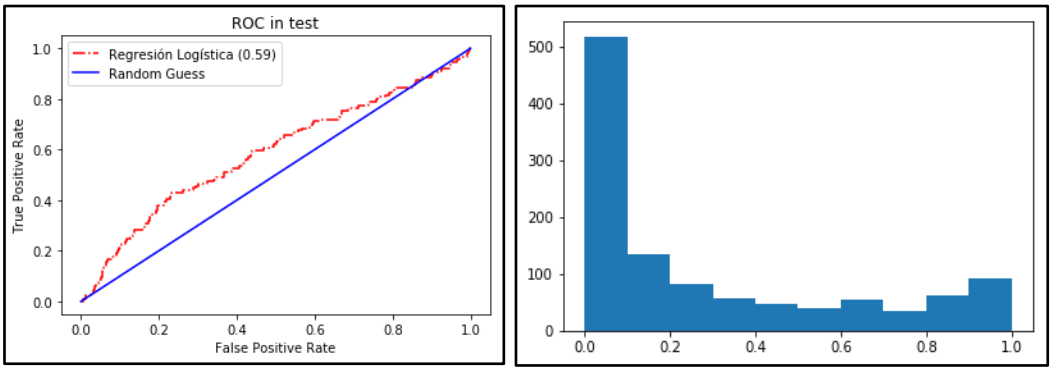


Figura 22: VGGNet-16 + Regresión Logística

2.6.2 Random Forest

En el caso de la utilización de un *random forest* (Ho, 1995), aunque el AUC es similar al caso anterior, la concentración en valores cercanos al 0,47, y que todos los valores sean menores a 0,50, hace que el modelo no sea una buena opción.

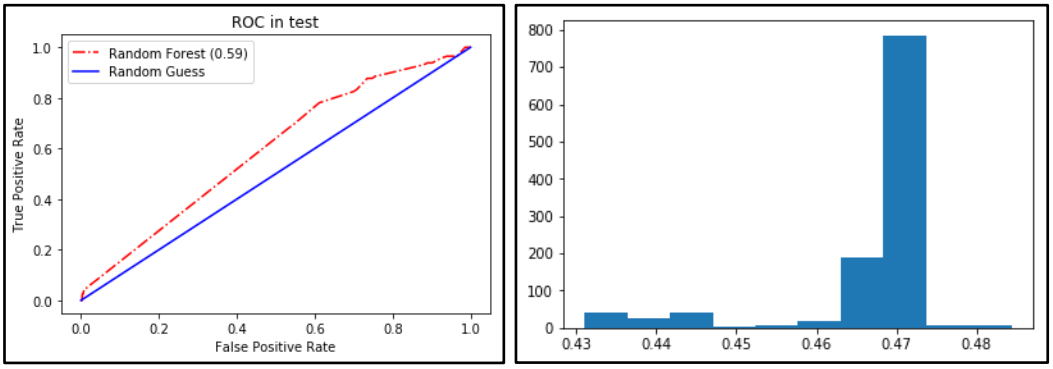


Figura 23: VGGNet-16 + Random Forest

2.6.3 Gradient Boosting Classifier

Como se observa en el siguiente gráfico, el modelo *gradient boosting* (Breiman, 1997) no consigue buenos resultados y su discriminación es nula (como se aprecia en la curva ROC).

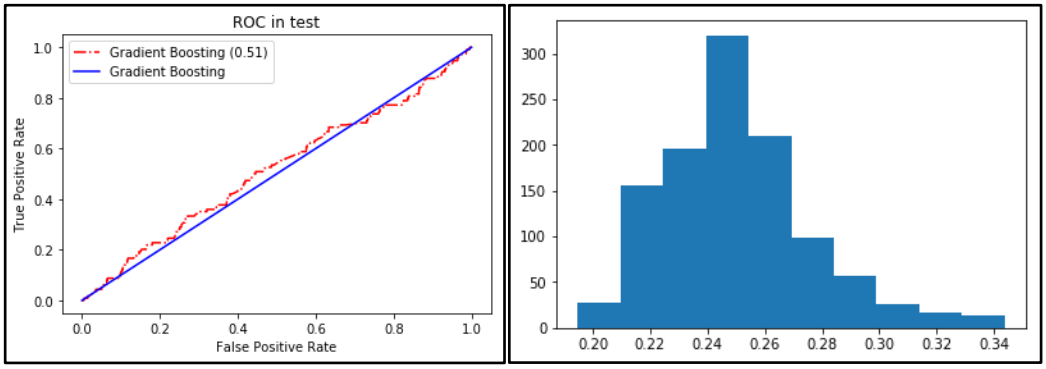


Figura 24: VGGNet-16 + Gradient Boosting Classifier

Para terminar, presentamos un cuadro comparativo usando tanto los valores por defecto de los hiper-parámetros como los valores óptimos encontrados por *grid search*.

		PARÁMETROS POR DEFECTO		GRID SEARCH	
		0	**1**	**0**	**1**
Regresión Logística	Score Avg.	0.74		0.73	
	Precission	0.88	0.1	0.91	0.11
	Recall	0.02	0.97	0.41	0.65
	F1-Score	0.04	0.18	0.57	0.19
	AUC	0.59		0.59	
Random Forest	Score Avg.	0.9		0.85	
	Precission	0.9	0	0.9	0
	Recall	1	0	1	0
	F1-Score	0.95	0	0.95	0
	AUC	0.54		0.59	
GBC	Score Avg.			0.89	
	Precission			0.9	0
	Recall			1	0
	F1-Score			0.95	0
	AUC			0.51	
MLP	Accuracy Avg.	0.64			
	Precission	0.93	0.16		
	Recall	0.65	0.59		
	F1-Score	0.76	0.25		
	AUC	0.62			

Tabla 6: Cuadro comparativo modelos

Como hemos observado los modelos basados en árboles no funcionan bien, lo que nos hace descartar este tipo de modelos como capa final.

El modelo de regresión logística funciona mejor. Lo podríamos asemejar, en el caso de una red neuronal, al caso de utilizar una única capa (la de salida) con una unidad sigmoide.

2.7 Comentarios finales a los resultados

Somos conscientes de que los resultados aún se encuentran muy alejados de lo deseado, sin embargo, son cercanos a los obtenidos por Wang et al. (2017).

Los motivos de estos resultados se deben a:

- Muestra demasiado pequeña para poder entrenar con cierta fiabilidad
- Datos no balanceados
- Probables problemas de etiquetado de las radiografías
- Equipo poco potente, ausencia de GPUs, no pudiendo realizar un entrenamiento completo

En la siguiente sección plantearemos las conclusiones del estudio y daremos algunas directrices de cómo se podría mejorar el modelo y hacia dónde creemos que se está dirigiendo la comunidad científica.

3 Conclusiones y Extensiones

3.1 Conclusiones

En el presente trabajo hemos tratado de hacer un modelo de inteligencia artificial para poder predecir hallazgos patológicos, masa y nódulo, que son síntomas de enfermedades como el cáncer. Es cierto que los resultados finales están aún alejados de unos resultados deseados, ya que, el modelo no discrimina lo suficiente. Como hemos mencionado a lo largo de todo el trabajo, unos mejores medios informáticos y una muestra más amplia podrían ayudar a mejorar estos resultados. No obstante, los resultados se acercan a otros conseguidos en trabajos de investigación con más medios como por ejemplo los trabajos de Wang *et al.* (2017, 2018).

Sin embargo, esta falta de medios ha propiciado que utilicemos una metodología que de alguna forma nos ayude a obtener los mejores resultados posibles. Por ello, este trabajo se debería ver como una metodología de trabajo y análisis, es decir, una prueba de concepto de cómo se pueden predecir hallazgos patológicos a través del análisis de imágenes radiológicas.

Podemos brevemente resumir el trabajo en 4 pasos:

- Conversión de las radiografías en tensores para su tratamiento a través de la red.

- Utilización de las capas de una red convolucional pre-entrenada (*transfer learning*) para la extracción de las características principales de las imágenes. Como hemos visto la red que mejor ha funcionado es la VGGNet-16.
- Entrenamiento de las últimas capas del modelo con una red neuronal artificial profunda (*Multilayer Perceptron*). En este apartado se ha tratado encontrar la arquitectura adecuada más sencilla y los hiper-parámetros que mejor funcionan.
- De manera complementaría se ha sustituido la red neuronal artificial profunda por tres modelos: regresión logística, *random forest* y *gradient boosting classifier*. Estos tres modelos se han entrenado usando *Grid Search* para encontrar los hiper-parámetros óptimos. Hemos de destacar el resultado de la regresión logística.

3.2 Futuras mejoras al modelo y próximos pasos

***Fine tuning* de todas las capas o de algunas de la VGGNet-16**

Utilizando un mayor poder de computación se podría entrenar la red convolucional. Se partiría de los valores iniciales de los pesos proporcionados por el modelo pre-entrenado, pero éstos se reajustarían entrenando con nuestros datos.

Entrenamiento completo de la VGGNet-16

En este caso no se acudirá a la técnica del *transfer learning* y se entrenará la red por completo. Para esto se necesitarán, aparte de una gran potencia de computación, una mayor cantidad de datos y tiempo de entrenamiento.

Trabajar con formatos de mayor calidad en la imagen: *dicom* o *nii.gz*

Para este trabajo se ha utilizado el formato *png* (que es el proporcionado por el NIH), pero este formato no es el nativo de radiografías y podrían no contener toda la información. A futuro se deberían utilizar formatos como *dicom* o *nii.gz* que son los habituales en este tipo de imágenes.

Utilización de otras bases de datos de radiografías

En enero de 2019 se ha publicado un nuevo artículo *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison* (Irvin *et al.*, 2019) en el cual se usa una nueva base de datos de aproximadamente 224,000 radiografías y 65,000 pacientes.

3.3 Posibles líneas de investigación futuras

Clasificación multi-etiqueta de los 14 hallazgos del *NIH*

En este caso se planteará el detectar los 14 hallazgos y, por tanto, en vez de tener un problema binario tendremos un problema multi-etiqueta, ya que, en una misma imagen puede haber varios hallazgos diferentes.

Construcción de mapas de calor

No sólo es importante detectar si una imagen contiene o no un hallazgo determinado, sino en qué parte de la imagen es más probable encontrarlo (Rajpurkar *et al.*, 2018). Esto ayudará a los facultativos a encontrarlos rápidamente.

Prueba de otras arquitecturas más complejas

En este proyecto se han probado VGGNet-16, VGGNet-19 y ResNet-50. Pero existen muchas más arquitecturas que se podrían evaluar como: AlexNet (Krizhevsky, Sutskever y Hinton, 2012), GoogLeNet (Szegedy *et al.*, 2018) e InceptionV3 (Szegedy *et al.*, 2015).

Creación de una arquitectura de red convolucional ad-hoc

Diseñar una arquitectura de CNN específica. En este caso no se podría usar *transfer learning*, a no ser que se usaran módulos de algunas redes pre-entrenadas.

Combinación de CNN con RNN, teniendo en cuenta el histórico de cada paciente

En el estudio de las patologías es importante e interesante el estudio de la evolución de las imágenes de cada paciente. En este caso habría que recurrir a las redes neuronales recurrentes (LSTM, GRU, …) combinadas con las CNN.

Uso de algoritmos de detección de objetos en tiempo real

Uso de algoritmos como R-CNN (Girshick *et al.*, 2013), Fast R-CNN (Girshick, 2015), Mask R-CNN (He *et al.*, 2017), y YOLO (Redmon y Angelova, 2014; Redmon *et al.*, 2015). Estos algoritmos se podrían adaptar a la detección de hallazgos en ecografías.

3.4 Algunos comentarios sobre los *notebooks* adjuntos

Importación y conversión de las imágenes.

En este *notebook* las imágenes se importan y se convierten a tensores. Además se añade el etiquetado y se crea la variable *target* que será nuestro objetivo a predecir. Los objetos generados se guardarán como un archivo binario usando la librería *Pickle*. En este código se aprovechará para escalar los valores.

Feature Extraction

En este *notebook* se pasarán los datos por la arquitectura pre-entrenada (*transfer learning*) obteniendo los datos finales para el entrenamiento de los modelos presentados. En el código se adaptan los datos para que cumplan los requisitos de dimensionalidad de los modelos pre-entrenados de la librería *Keras*.

Red Neuronal Artificial Profunda

Este es el código central del trabajo, en él se presenta el tratamiento de los datos no balanceados, el diseño de la red profunda, su entrenamiento, validación y la comparación con los modelos de *benchmarking* (recordemos que los hiper-parámetros de estos modelos son los valores por defecto).

Regresión Logística, Random Forest y *Gradient Boosting Classifier*

Para finalizar y como extensión a los resultados presentamos un código con los tres modelos alternativos. El código contiene tanto el entrenamiento como los tests. Para entrenar los modelos se ha usado la librería *Hypopt* que nos permite hacer *Grid Search* con una muestra dada de validación.

3.5 Contenido digital adjunto

En el *pendrive* se adjunta lo siguiente:

- Proyecto en formato PDF
- Presentación en formato PDF
- Notebooks del Modelo final
- Notebooks de los otros modelos probados
- Datos
- Bibliografía principal

4 Bibliografía

Aronson, A. R. y Lang, F.-M. (2010) «An overview of MetaMap: historical perspective and recent advances», *Journal of the American Medical Informatics Association*, 17(3), pp. 229-236. doi: 10.1136/jamia.2009.002733.

Berzal, F. (2018) *Redes Neuronales & Deep Learning*. Disponible en: https://www.amazon.es/Redes-Neuronales-Deep-Learning-Edición/dp/1731314337 (Accedido: 3 de enero de 2019).

Breiman, L. (1997) «Arcing the Edge».

ChestXray-NIHCC (2018). Disponible en: https://nihcc.app.box.com/v/ChestXray-NIHCC (Accedido: 10 de enero de 2019).

Chollet, F. (2017) *Deep learning with Python*. Disponible en: https://www.oreilly.com/library/view/deep-learning-with/9781617294433/ (Accedido: 2 de enero de 2019).

CNN Architectures — LeNet, AlexNet, VGG, GoogLeNet and ResNet – mc.ai (2018). Disponible en: https://mc.ai/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet/ (Accedido: 17 de enero de 2019).

Common architectures in convolutional neural networks. (2018). Disponible en: https://www.jeremyjordan.me/convnet-architectures/ (Accedido: 18 de enero de 2019).

Cox, D. R. (1958) «The Regression Analysis of Binary Sequences», *Journal of the Royal Statistical Society. Series B (Methodological)*. WileyRoyal Statistical Society, pp. 215-242. doi: 10.2307/2983890.

Deng, J. *et al.* (2009) «ImageNet: A large-scale hierarchical image database», en *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 248-255. doi: 10.1109/CVPR.2009.5206848.

Diccionario de la lengua española - Edición del Tricentenario (2019). Disponible en: http://dle.rae.es/ (Accedido: 9 de enero de 2019).

Diccionario médico. Clínica Universidad de Navarra (2019). Disponible en: https://www.cun.es/diccionario-medico (Accedido: 9 de enero de 2019).

Girshick, R. *et al.* (2013) «Rich feature hierarchies for accurate object detection and semantic segmentation». Disponible en: http://arxiv.org/abs/1311.2524 (Accedido: 11 de marzo de 2019).

Girshick, R. (2015) «Fast R-CNN». Disponible en: http://arxiv.org/abs/1504.08083 (Accedido: 11 de marzo de 2019).

Global Cancer Observatory (2018). Disponible en: http://gco.iarc.fr/ (Accedido: 7 de enero de 2019).

He, K. *et al.* (2015) «Deep Residual Learning for Image Recognition», en *IEEE Conference on Computer Vision and Pattern Recognition*. Disponible en: http://image-net.org/challenges/LSVRC/2015/ (Accedido: 14 de enero de 2019).

He, K. *et al.* (2017) «Mask R-CNN». Disponible en: http://arxiv.org/abs/1703.06870 (Accedido: 11 de marzo de 2019).

Ho, T. K. (1995) «Random Decision Forests», *Proceedings of the 3rd International Conference on Document Analysis and Recognition*.

Ioffe, S. y Szegedy, C. (2015) «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift». Disponible en: http://arxiv.org/abs/1502.03167 (Accedido: 11 de marzo de 2019).

Irvin, J. *et al.* (2019) «CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison». Disponible en: http://arxiv.org/abs/1901.07031 (Accedido: 30 de marzo de 2019).

Karim, M. R., Sewak, M. y Pujari, P. (2018) *Practical Convolutional Neural Networks : Implement advanced deep learning models using Python.* Packt Publishing.

Kingma, D. P. y Ba, J. (2014) «Adam: A Method for Stochastic Optimization». Disponible en: http://arxiv.org/abs/1412.6980 (Accedido: 11 de febrero de 2019).

Krizhevsky, A., Sutskever, I. y Hinton, G. E. (2012) «ImageNet Classification with Deep Convolutional Neural Networks», *Advance in neural information processing systems*. Disponible en: http://code.google.com/p/cuda-convnet/ (Accedido: 14 de enero de 2019).

Leaman, R., Khare, R. y Lu, Z. (2015) «Challenges in clinical natural language processing for automated disorder normalization», *Journal of Biomedical Informatics*, 57, pp. 28-37. doi: 10.1016/j.jbi.2015.07.010.

Lin, M., Chen, Q. y Yan, S. (2014) «Network In Network», *Arxiv (Preprint)*. Disponible en: https://arxiv.org/pdf/1312.4400.pdf (Accedido: 14 de enero de 2019).

Litjens, G. *et al.* (2017) «A survey on deep learning in medical image analysis», *Medical Image Analysis*. Elsevier, 42, pp. 60-88. doi: 10.1016/J.MEDIA.2017.07.005.

Martín-Sánchez, J. C. *et al.* (2018) «Projections in Breast and Lung Cancer Mortality among Women: A Bayesian Analysis of 52 Countries Worldwide», *Cancer Research*, 78(15), pp. 4436-4442. doi: 10.1158/0008-5472.CAN-18-0187.

New Global Cancer Data: GLOBOCAN 2018 | UICC (2018). Disponible en: https://www.uicc.org/new-global-cancer-data-globocan-2018 (Accedido: 7 de enero de 2019).

NIH Chest X-ray Dataset Sample (2017). Disponible en: https://www.kaggle.com/nih-chest-xrays/sample/home (Accedido: 21 de diciembre de 2018).

NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community | National Institutes of Health (NIH) (sin fecha). Disponible en: https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community (Accedido: 2 de enero de 2019).

Oakden-Rayner, L. (2017) *Exploring the ChestXray14 dataset: problems – Luke Oakden-Rayner*. Disponible en: https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/ (Accedido: 13 de enero de 2019).

Rajpurkar, P. *et al.* (2018) «Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists», *PLOS Medicine*. Editado por A. Sheikh, 15(11), p. e1002686. doi: 10.1371/journal.pmed.1002686.

Redmon, J. *et al.* (2015) «You Only Look Once: Unified, Real-Time Object Detection». Disponible en: http://arxiv.org/abs/1506.02640 (Accedido: 9 de marzo de 2019).

Redmon, J. y Angelova, A. (2014) «Real-Time Grasp Detection Using Convolutional Neural Networks». Disponible en: http://arxiv.org/abs/1412.3128 (Accedido: 9 de marzo de 2019).

ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks – CV-Tricks.com (2018). Disponible en: https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/ (Accedido: 17 de enero de 2019).

Review: GoogLeNet (Inception v1)— Winner of ILSVRC 2014 (Image Classification) (2018). Disponible en: https://medium.com/coinmonks/paper-review-of-googlenet-inception-

v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7 (Accedido: 17 de enero de 2019).

Rogers, T. K. (2010) «Primary care radiography in the early diagnosis of lung cancer.», *Cancer imaging : the official publication of the International Cancer Imaging Society*. BioMed Central, 10(1), pp. 73-6. doi: 10.1102/1470-7330.2010.0007.

Shen, D., Wu, G. y Suk, H.-I. (2017) «Deep Learning in Medical Image Analysis.», *Annual review of biomedical engineering*. NIH Public Access, 19, pp. 221-248. doi: 10.1146/annurev-bioeng-071516-044442.

Simonyan, K. y Zisserman, A. (2014) «Very Deep Convolutional Networks for Large-Scale Image Recognition», *arXiv preprint*. Disponible en: http://arxiv.org/abs/1409.1556 (Accedido: 14 de enero de 2019).

Srivastava, N. *et al.* (2014) «Dropout: A Simple Way to Prevent Neural Networks from Overfitting», *Journal of Machine Learning Research*, 15, pp. 1929-1958. Disponible en: http://jmlr.org/papers/v15/srivastava14a.html (Accedido: 11 de marzo de 2019).

Szegedy, C. *et al.* (2015) «Rethinking the Inception Architecture for Computer Vision». Disponible en: http://arxiv.org/abs/1512.00567 (Accedido: 13 de marzo de 2019).

Szegedy, C. *et al.* (2018) «Going Deeper with Convolutions», en *IEEE Conference on Computer Vision and Pattern Recognition*. Disponible en: https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf (Accedido: 14 de enero de 2019).

Topol, E. J. (2019) «High-performance medicine: the convergence of human and artificial intelligence.», *Nature medicine*, 25(1), pp. 44-56. doi: 10.1038/s41591-018-0300-7.

Wang, L. (2017) «Early Diagnosis of Breast Cancer.», *Sensors (Basel, Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI), 17(7). doi: 10.3390/s17071572.

Wang, X. *et al.* (2017) «ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases». Disponible en: http://arxiv.org/abs/1705.02315 (Accedido: 2 de enero de 2019).

Wang, X. *et al.* (2018) «TieNet: Text-Image Embedding Network for Common Thorax Disease Classification and Reporting in Chest X-rays». Disponible en: http://arxiv.org/abs/1801.04334 (Accedido: 2 de enero de 2019).