

---

# DETECCIÓN DE NEUMONÍA MEDIANTE DEEP LEARNING.

---

**Ivan Fayos Tormo**

Universidad de Valencia  
Máster en Ciencia de Datos  
ifator@alumni.uv.es

**Alberto Solano Cañadas**

Universidad de Valencia  
Máster en Ciencia de Datos  
alsoca2@alumni.uv.es

21 de mayo de 2020

## 1. Introducción

La neumonía es una enfermedad del sistema respiratorio que consiste en la inflamación de los espacios alveolares de los pulmones, produciendo tos con flema o pus, fiebre, escalofríos y dificultad para respirar. Se trata de una enfermedad que varía en gravedad, siendo potencialmente mortal en las personas más vulnerables como ancianos o niños. Para poner números a esta grave enfermedad, la OMS estimó en 2015 que el número de muertos entre niños menores de 5 años debido a esta enfermedad ascendió a 922.000, el 15 % de todas las muertes en esta franja de edad.

Los países en vías de desarrollo sufren por partida doble el impacto de esta enfermedad. Por un lado, es común la malnutrición o directamente la desnutrición en algunas personas, lo que debilita su sistema inmunitario y los hace propensos a caer enfermos. Además, estas personas conviven día a día con un ambiente altamente contaminado. Las partículas en suspensión producto de la polución no solo son factores que causan la neumonía, sino que además agravan la condición de los que ya han enfermado. Por otro lado, la falta de recursos médicos y personal sanitario complica la situación de los pacientes, sobretodo aquellos que no tienen recursos como para acceder a una sanidad privada como ocurre por ejemplo en India, donde la diferencia entre los recursos médicos que posee el sector privado y el público son considerables, estando este último completamente saturado por las numerosas personas que necesitan atención sanitaria.

En este contexto en el que los médicos no disponen de mucho tiempo, surge la necesidad de un diagnóstico rápido y preciso de esta enfermedad. La radiografía de tórax es el procedimiento estándar para detectar en los pulmones indicios de la presencia de características patológicas asociadas a la neumonía. Sin embargo, esto requiere del ojo de un experto para poder distinguir si el paciente tiene realmente neumonía, o por contra está sano. Nuestra propuesta es crear una herramienta de clasificación automática de radiografías que sea complementaria a la decisión del médico, es decir, que le ayude a diagnosticar.

La idea por tanto es tratar de aprovechar imágenes de radiografías que previamente un experto haya etiquetado (paciente sano o paciente enfermo) para entrenar un sistema capaz de diagnosticar esta enfermedad, lo cual permitiría aliviar la carga de trabajo de los médicos. Además, hemos tratado de dar un paso más allá para no distinguir únicamente entre pacientes sanos y con neumonía, sino que a su vez hemos querido diferenciar entre aquellos que presentan una neumonía de tipo vírico y aquellos que padecen una neumonía de tipo bacteriano. Al parecer, la neumonía viral se manifiesta con un patrón “intersticial” más difuso en los pulmones, mientras que la neumonía bacteriana exhibe una “consolidación lobular focal”, es decir, está más concentrada, luego al ser esta diferencia apreciable a nivel de radiografía, es razonable pensar que podemos elaborar modelo que pueda llegar a aprender los patrones necesarios para realizar esta distinción. Esto sería también de gran utilidad ya que el tratamiento es completamente diferente en cada caso.

## 2. Análisis exploratorio de datos

Los datos que hemos utilizado consisten en un total de 5863 imágenes etiquetadas de radiografías de tórax (1583 pacientes sin neumonía y 4280 con neumonía) extraídas de la plataforma Kaggle. (<https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia> [1]). Estas imágenes se corresponden con pacientes pediátricos de 1 a 5 años, es decir niños, por lo que el modelo que elaboraremos con estos datos estará siempre orientado al diagnóstico de pacientes en esta franja de edad. Es importante recalcar esto, ya que las diferencias anatómicas entre un adulto y un niño son considerables, por lo que no podemos pretender que un modelo elaborado con estas imágenes sea capaz de extrapolarse para realizar predicciones en radiografías de adultos. Pensamos que este hecho, lejos de ser un

problema, otorga más especificidad al modelo en el diagnóstico de esta enfermedad precisamente en el grupo de mayor riesgo, los niños.

Inspeccionando las imágenes del dataset escogido vimos que, por lo general, la calidad de las mismas era bastante buena, en el sentido de que el trabajo de preprocessado de los datos no iba a ser una tarea tan pesada en comparación con otros conjuntos de datos. Por ejemplo, encontramos que todas las radiografías están tomadas frontalmente y no se mezclan radiografías tomadas desde distintas perspectivas (como por ejemplo perspectivas laterales). Además, la mayoría de ellas capturan una zona similar (tórax completo) del paciente, al margen de pequeñas diferencias tal y como se puede observar en la Figura 1.

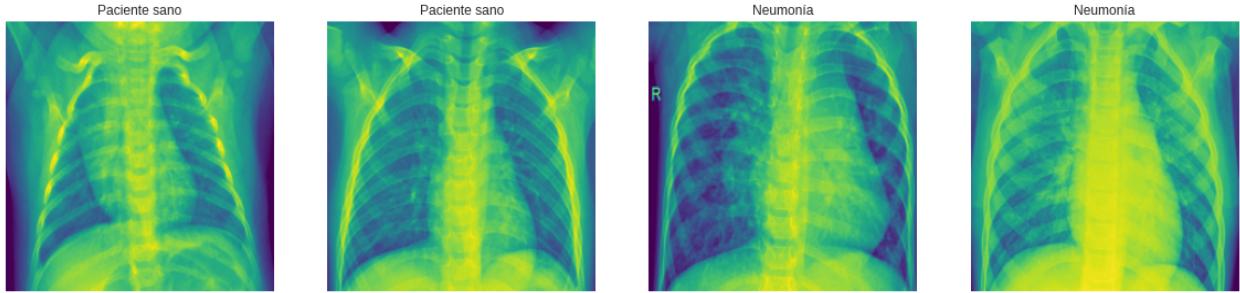


Figura 1: Imágenes extraídas del conjunto de datos utilizado correspondientes a pacientes tanto sanos como con neumonía.

El primer problema al que nos enfrentamos fue el hecho de que las imágenes no tenían todas el mismo tamaño, de hecho, el rango en el que se distribuyen las dimensiones de las imágenes es bastante amplio (Figura 2).

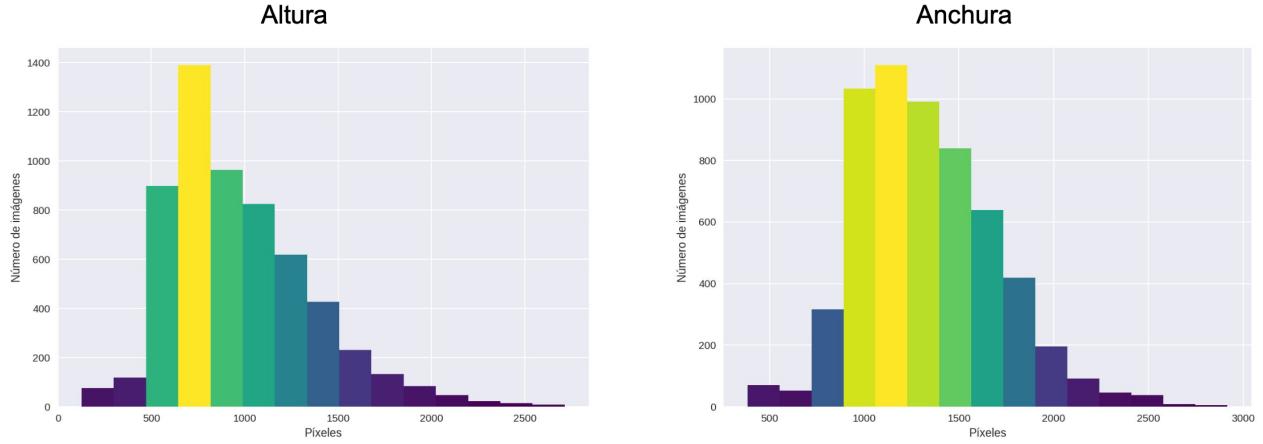
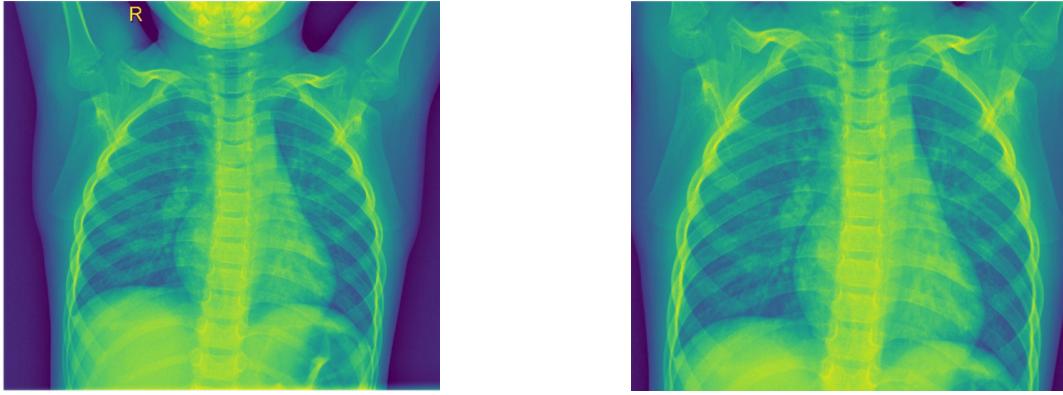


Figura 2: Histogramas con las dimensiones de las imágenes del conjunto de datos utilizado, a la izquierda se muestra la distribución en altura y a la derecha en anchura.

Decimos que esto es un problema ya que, de cara a la elaboración del modelo, en general cuando se utilizan redes convolucionales, se suele tratar de emplear imágenes del mismo tamaño. Utilizar como entrada de la red imágenes de distinto tamaño puede ser problemático a la hora de conectar las capas convolucionales con las “fully connected”. Además, una de las técnicas más habituales y potentes se denomina (Transfer Learning) y consiste en la utilización de redes ya entrenadas previamente, tal y como hemos realizado nosotros (Ver Sección 3), por lo que si queremos emplear estas arquitecturas junto a los pesos asociados al entrenamiento de las mismas, deberíamos emplear imágenes del mismo tamaño que las que se emplearon para entrenar dichas redes, ya que el potencial de extracción de características de la red se maximiza para imágenes de estas dimensiones.

Por otro lado, realizando una inspección manual de las imágenes nos dimos cuenta de que, mientras que las imágenes más pequeñas en su mayoría están bastante bien delimitadas, las de mayores dimensiones solían cubrir una región más amplia que el tórax, es decir, en los laterales presentaban una franjas negras considerables (varios centímetros desde el último punto con tejido) y también en el eje vertical abarcaban una región más extensa que los propios pulmones

(incluyendo por ejemplo parte de la cadera o el cráneo). Por este motivo, ideamos una función que recortara en cierta medida todas las imágenes de nuestro dataset en proporción a las dimensiones de las mismas, de esta manera estaríamos obligando desde un primer momento a la red a fijarse únicamente en la zona de los pulmones, facilitándole a la red la tarea de aprendizaje.



(a) Radiografía antes de ser recortada con nuestra función.

(b) Misma imagen después del recorte.

Figura 3: Ejemplo del resultado conseguido mediante el recorte personalizado, con nuestra función, de una de las imágenes del dataset.

En la Figura 3 se puede ver el resultado al hacer un preprocesado previo, recortando las imágenes de acuerdo a nuestra función. El recorte se basa en el cálculo de un factor de reescalado, que principalmente tiene que ver con el cociente entre las dimensiones de la imagen y la imagen de mayor tamaño del total de las imágenes.

$$\alpha_{width} = 1 - \frac{3.5}{10} * \frac{width}{max\_width} \quad (1)$$

$$\alpha_{height} = 1 - \frac{2}{10} * \frac{height}{max\_height} \quad (2)$$

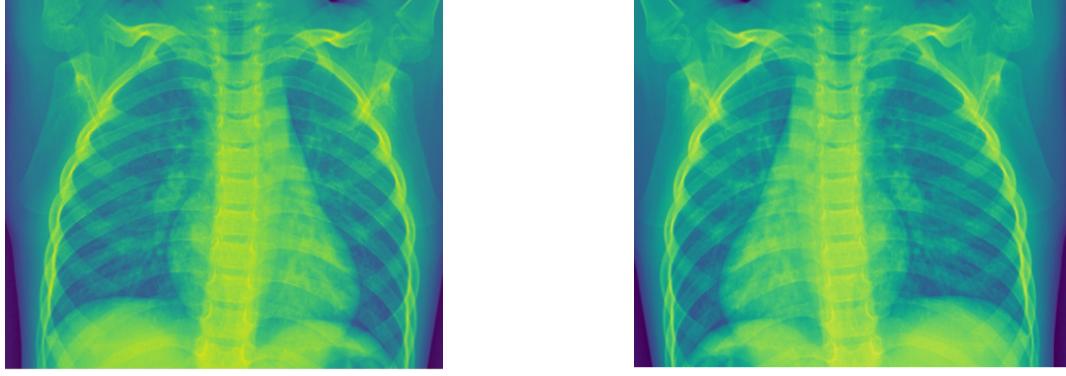
El nuevo tamaño que tendrá la imagen una vez recortada será el producto de los coeficientes de las ecuaciones 1 y 2 con las dimensiones actuales de la imagen (anchura y altura respectivamente). De esta manera, estaríamos recortando un máximo de un 35 % en anchura y un 20 % en altura para la imagen de mayor tamaño. Estos números obedecen a un criterio basado en la inspección de las imágenes, ya que por lo general se necesitaba un recorte mayor en la anchura que en la altura de la imagen, aunque se podría probar a cambiar estos valores o la forma de recortar las imágenes para tratar de conseguir un resultado mejor.

## 2.1. Data Augmentation

Es habitual en problemas relacionados con clasificación de imágenes aplicar ciertas transformaciones a las imágenes del conjunto de entrenamiento, de esta manera se consigue ampliar el número de imágenes disponibles para entrenar el modelo. A partir de una única imagen se pueden crear tantas nuevas imágenes como transformaciones se le apliquen a esta, y para el modelo serán imágenes diferentes que no guardan relación con la original.

Siguiendo esta misma filosofía hemos aplicado algunas transformaciones a nuestras imágenes de forma que, a la hora de realizar el bucle de entrenamiento, cuando las imágenes “entren” al modelo estarán sujetas a unas ciertas transformaciones aleatorias que consisten en:

- **Reflexión horizontal aleatoria:** La imagen tiene una probabilidad del 50 % de aparecer reflejada con respecto a la columna vertebral.
- **Desplazamiento aleatorio de los valores del brillo:** La imagen puede tomar valores en el rango [-10 %, +10 %] del brillo de la imagen original de manera aleatoria.
- **Desplazamiento aleatorio de los valores del contraste:** La imagen puede tomar valores en el rango [0 %, +10 %] del contraste de la imagen original de manera aleatoria.



(a) Imagen original.

(b) Imagen reflejada.

Figura 4: Ejemplo de la reflexión que proponemos. Esta transformación la realizamos porque la presencia de neumonía puede darse indistintamente tanto en el pulmón derecho como en el izquierdo o puede que en ambos, con lo que pretendemos enfatizar en la red que las etiquetas son invariantes ante esta rotación.

De esta manera, cuando se complete una “epoch” dentro del bucle de entrenamiento, es decir, un periodo completo por el que todas las imágenes pasan por la red, en la siguiente epoch las imágenes, al volver a pasar de nuevo por la red, volverán a estar sujetas a la aleatoriedad de las transformaciones, por lo que el modelo, probablemente, nunca percibirá dos imágenes idénticas, lo cual contribuye a evitar el sobreajuste.

### 3. Elaboración del modelo de clasificación.

El modelo que hemos utilizado esta basado en redes convolucionales, más concretamente, hemos aprovechado la arquitectura y los pesos de una red preentrenada como hemos adelantado en secciones previas (*Transfer Learning*).

El uso de este tipo de redes viene motivado por el amplio consenso que existe acerca del uso de redes convolucionales en los problemas que tienen que ver con el uso de imágenes. Esta práctica tiene sus orígenes en el año 2012, cuando por primera vez se utilizó una red convolucional llamada AlexNet [2] para participar en un concurso anual llamado “ImageNet Challenge”, el cual consistía en conseguir el menor error de clasificación sobre una base de datos (ImageNet [3]) que contenía mas de 1 millón de imágenes pertenecientes a un total de 1000 categorías. Los resultados que se obtuvieron con esta red sentaron un precedente histórico (reduciendo más de un 10 % el error cometido por el mejor modelo que se tenía hasta la fecha) no solo en esta competición, sino en toda la comunidad. Desde ese momento, la popularidad junto al potencial que exhiben las redes convolucionales no ha parado de crecer, estableciendo un estándar en los problemas relacionados con imágenes.

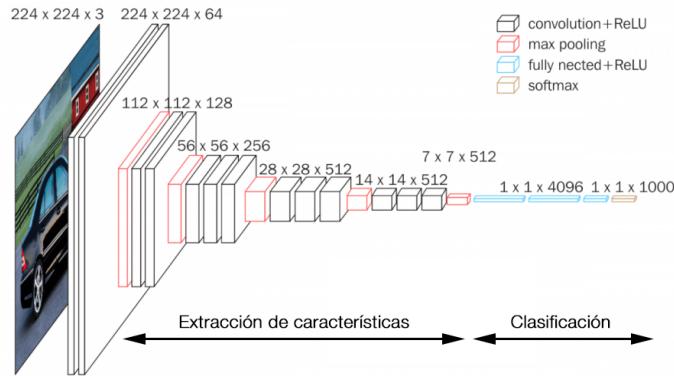


Figura 5: Arquitectura de la red VGG16.

En particular en este trabajo hemos utilizado la arquitectura VGG16 [4] (Ver Figura 5), ya preentrenada con el conjunto de datos de la base de datos de ImageNet. Entrenaremos dos redes distintas pero con la misma arquitectura, una se especializara en distinguir entre pacientes sanos y pacientes con neumonía, y la otra distinguirá entre neumonía

bacteriana y neumonía vírica, de forma que el clasificador final será en dos etapas. Las imágenes en primer lugar pasarán por el primer clasificador, y las que se clasifiquen en pacientes con neumonía se pasarán al segundo clasificador para diferenciar entre vírica o bacteriana.

Para poder utilizar esta arquitectura en nuestro problema y utilizar la técnica de Transfer Learning, hemos tenido que adaptarla.

En primer lugar, como es común al utilizar esta técnica, hay que reemplazar la última capa densa. Esta red está entrenada para distinguir entre 1000 clases, por lo que la última capa tiene 1000 neuronas de salida. En nuestro caso como tan solo queremos distinguir entre 2 clases (sano y neumonía en un caso, y bacteriana y vírica en el otro caso) tenemos que reemplazar esta capa densa por otra capa densa con tan solo 2 neuronas de salida.

Por otro lado, las imágenes utilizadas para entrenar la VGG16 eran imágenes con dimensiones  $224 \times 224$  con 3 canales de color (RGB) por lo que la red solo admite imágenes con 3 canales. En nuestro caso, las radiografías son en blanco y negro, es decir, tan solo tienen un canal. Para lidiar con este problema simplemente hemos añadido una capa convolucional a la entrada de la red, la cual recibe como entrada imágenes con dimensiones  $224 \times 224$  y un solo canal y tiene como salida 3 filtros. De esta forma, la salida de esta capa convolucional ya puede ser tomada como entrada por la red VGG16 original.

Finalmente, para aprovechar el poder de extracción de características de la red VGG16 preentrenada con el dataset de ImageNet, “congelamos” los pesos de las capas convolucionales de la red (parte especializada en la extracción de características) excepto la primera capa convolucional añadida. Por “congelar” se entiende que los pesos de estas capas no se van a modificar durante el entrenamiento de la red. Tan solo se entrenarán las capas densas del final (parte de la red especializada en la clasificación) y la primera capa convolucional añadida.

Se puede observar un esquema de como quedaría la arquitectura de la red tras todas estas modificaciones en la Figura 6.

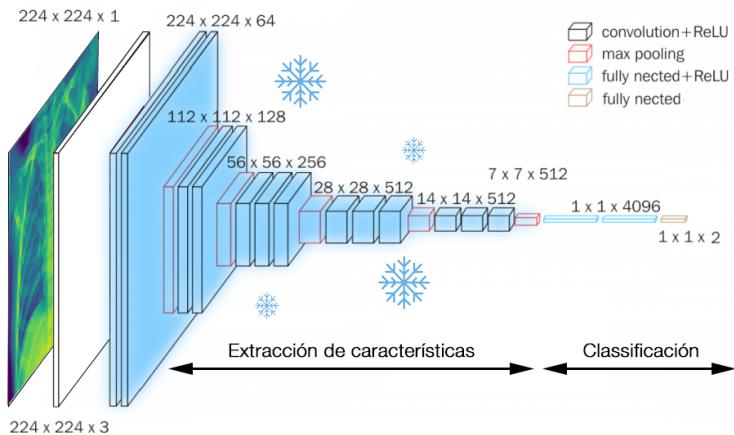


Figura 6: Arquitectura utilizada en este trabajo tras aplicar todas las transformaciones necesarias a la arquitectura de la red VGG16.

Al explorar los datos, nos hemos dado cuenta de que existía un desbalanceo entre las clases normal y neumonía. Tenemos 1583 muestras normales (27 %) y 4280 pacientes con neumonía (73 %).

Para lidiar con este desbalanceo hemos empleado una técnica bastante conocida en el ámbito del aprendizaje máquina que consiste en modificar la función de coste del modelo para que penalice de forma distinta en función de la clase a la que pertenezca la muestra. La idea principal es introducir unos pesos que hagan que el coste de clasificar erróneamente la clase minoritaria sea mucho mayor al coste de clasificar erróneamente la clase mayoritaria. En particular, en nuestro caso hemos utilizado unos pesos que añaden una penalización a los errores cometidos en la clase minoritaria proporcional al

desbalanceo en las muestras. Es decir, equivocarse en la clase minoritaria supone un coste de  $0.73/0.27 \approx 2.7$  veces el coste de equivocarse en la clase mayoritaria.

#### 4. Explicabilidad del modelo

A lo largo de los últimos años hemos visto como ha surgido una nueva rama dentro del machine learning llamada *Explainable Artificial Intelligence (XAI)*, que trata de entender y explicar las razones por las cuales un modelo ha predicho un determinado resultado. En los problemas de clasificación, como es este caso, se buscaría entender el motivo por el cual el modelo estaría prediciendo una de las etiquetas posibles y no otra.

Un caso habitual de clasificación consiste en predecir una variable objetivo en base a los valores de distintas variables predictores y numerosos registros. Podríamos decir que un modelo sería explicable si supiéramos los intervalos que internamente el modelo elabora para cada una de las variables para catalogar la observación en una de las etiquetas de la variable objetivo. En este sentido, modelos como los árboles de decisión cumplen esta definición de modelos explicables. Por otro lado, existen modelos parcialmente explicables como podría ser el algoritmo Random Forest aplicado al mismo ejemplo anterior. Este algoritmo internamente es bastante más complejo porque es un “ensamble” de Árboles de Decisión. Cada árbol que lo compone posee sus propios criterios para las predicciones, por lo que no es viable tratar de mirar con lupa todos los árboles para entender el conjunto de todos ellos, sin embargo, se puede obtener la importancia que el modelo otorga a cada una de las variables para realizar las predicciones, por lo que tendríamos cierta información sobre lo que el modelo se basa para predecir. Por último, para este mismo problema de clasificación si empleamos por ejemplo una red neuronal, al margen de los resultados que obtuviéramos, a priori no entenderíamos a nivel interno los mecanismos por los cuales la clasificación estaría o no funcionando. Por este motivo, se suelen considerar las redes neuronales como “cajas negras” y esto se extiende a otros problemas como puede ser la predicción de series temporales o la clasificación en imágenes como es nuestro caso.

Como cada vez se genera más interés en eliminar esta opacidad de los modelos, han ido surgiendo en los últimos años herramientas dedicadas a estos propósitos. La necesidad es clara, en sistemas en los que se pretende que haya una cooperación algoritmo-humano, es fundamental que el humano sea capaz de entender y confiar en las predicciones que la máquina le está proporcionando. En el caso clínico (nuestro caso) las herramientas de machine learning aplicadas a diagnóstico siempre se usan de manera complementaria al criterio del experto, y en ello radica su potencial, por ello en este campo es aún más necesario y se enfatiza cada vez más en la explicabilidad de los modelos.

En particular, en nuestro trabajo utilizaremos una técnica basada en la oclusión que nos proporciona una librería especializada en la IA explicable llamada *Captum* [5]. La idea principal es ocultar una porción de la imagen y observar como la pérdida de información producida por esta oclusión modifica la predicción del modelo. La estrategia que empleamos consiste en definir una ventana de  $15 \times 15$  píxeles que vamos desplazando mediante pasos de  $10 \times 10$  píxeles por toda la imagen. Para cada oclusión realizada se guarda cuánto han cambiado las probabilidades de pertenencia a cada clase que predice la red y en función de estas se asigna una importancia. Con esto podemos realizar una mapa de calor donde las zonas más intensas representan zonas de la imagen más decisivas para la clasificación de nuestro modelo.

#### 5. Resultados

Con un test de 624 muestras (63 % de neumonía y 37 % de sanos) hemos conseguido un *accuracy* de:

$$acc = 0.92$$

clasificando entre pacientes con neumonía y pacientes sanos. Podemos ver además dónde estamos cometiendo los errores, si diagnosticando a pacientes sanos con neumonía, o al revés, diagnosticando como sanos a pacientes con neumonía. Mostramos la matriz de confusión del modelo en la Figura 7a.

Se puede observar como se cometen más errores de tipo 1 que de tipo 2, es decir, se diagnostica más veces con neumonía a pacientes sanos que como sanos a pacientes con neumonía. En este caso es preferible que un paciente sano sea diagnosticado con neumonía y no al revés, ya que no detectar a un paciente con neumonía puede suponer futuras complicaciones incluso muertes (preferimos un buen *recall*).

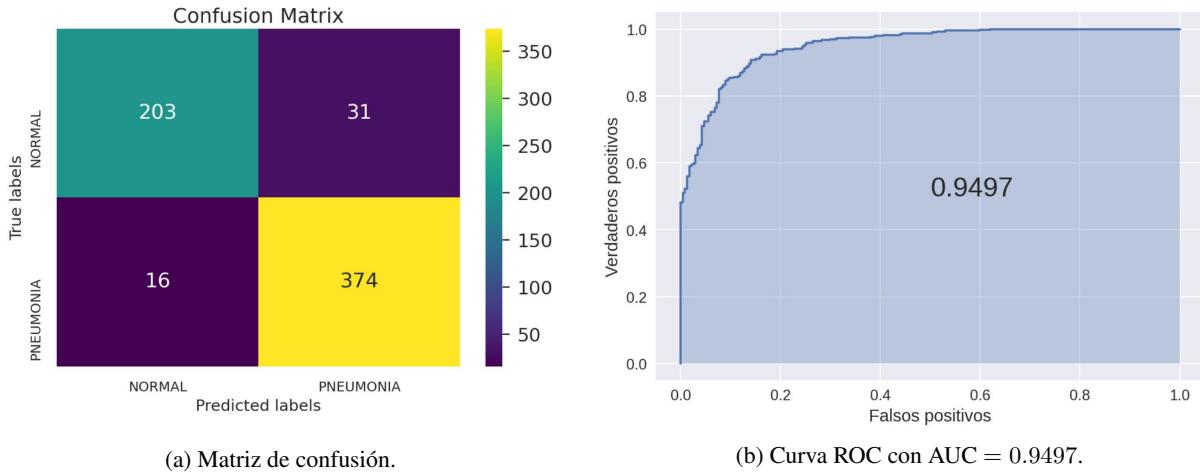


Figura 7: Métricas del clasificador de radiografías en pacientes sanos y pacientes con neumonía.

Hemos evaluado también el modelo con métricas que no son sensibles al desbalanceo de clases como es el AUC (área bajo la curva ROC). Podemos ver la curva ROC descrita por el modelo en la Figura 7b. Esta curva nos permite además estudiar el intercambio de verdaderos positivos por falsos positivos. Es decir, imaginemos que queremos aumentar el acierto en la detección de pacientes con neumonía. A partir del modelo que hemos entrenado, este aumento en la detección de pacientes de neumonía, ¿cuántos falsos positivos (pacientes sanos que diagnosticamos erróneamente como enfermos de neumonía) supondría? Pues bien, según la curva ROC, si por ejemplo queremos aumentar la detección de casos de neumonía y detectar el 99 % de las radiografías con neumonía, a cambio, sacrificaríamos el número de radiografías de pacientes sanos correctamente clasificadas y aumentaríamos el número de pacientes diagnosticados con neumonía erróneamente al 40 % aproximadamente (del total de muestras clasificadas como neumonía el 40 % en realidad serían radiografías de pacientes sanos). Esto nos permite fijar cuantos diagnósticos de neumonía a pacientes sanos estamos dispuestos a pagar a cambio de asegurarnos una mayor detección de casos de neumonía.

Una vez hemos entrenado este primer modelo, hemos añadido un segundo clasificador para distinguir entre neumonía bacteriana y vírica. Hemos diseñado una “pipeline” con ambos modelos de forma que introducimos la radiografía en el primer modelo, este la clasifica en normal y neumonía, y las clasificadas como neumonía se envían al segundo modelo que realiza una segunda clasificación en bacteriana y vírica. Con esto conseguimos no sacrificar el *accuracy* del primer clasificador por haber incluido una segunda distinción de las neumonías bacteriana y vírica.

Con toda esta “pipeline”, sobre los mismo datos test, distinguiendo entre las clases normal, bacteriana y vírica conseguimos un *accuracy* de:

$$acc = 0.86$$

(el descenso en el *accuracy* respecto al modelo anterior es debido a errores entre la clasificación vírica y bacteriana ya que las clasificaciones entre normal y neumonía son las mismas). Además mostramos la matriz de confusión entre las tres clases en la Figura 8.

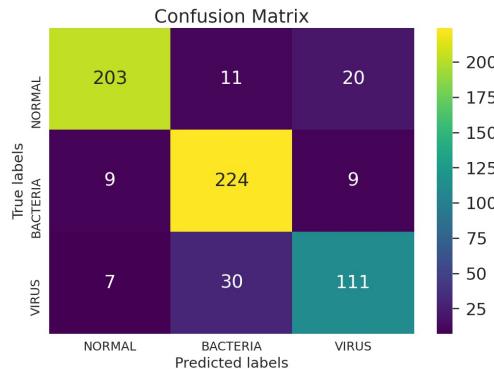


Figura 8: Matriz de confusión de la “pipeline” con los dos modelos.

Se observa que los errores más comunes cometidos por el modelo es confundir la neumonía bacteriana con la vírica (30 casos) y la neumonía vírica con pacientes sanos (20 casos).

Hemos implementado un script que toma todas las radiografías de una carpeta y genera para cada una de ellas una imagen con la radiografía original, las probabilidades de pertenencia a cada clase según el modelo y el mapa de calor de las zonas de la radiografía que el modelo considera importantes para realizar la clasificación entre normal y neumonía.

Se puede encontrar este script, con los pesos de los modelos entrenados de este trabajo en el siguiente enlace de GitHub: [https://github.com/ifayost/cdb\\_pneumonia](https://github.com/ifayost/cdb_pneumonia)

Mostramos 3 ejemplos (uno para cada clase) de estos resultados en la Figura 9.

A la vista de los mapas de calor (Figura 9), podemos destacar varios fenómenos de interés. En primer lugar, tenemos que decir que los mapas de calor hacen referencia al primer clasificador, es decir, están indicando las zonas de importancia para distinguir entre sano y con neumonía. Por lo general, las zonas a las que asigna mayor importancia parecen seguir un determinado patrón el cual no parece ser producto de la aleatoriedad. De hecho, raramente podemos encontrar imágenes en las que se le otorgue importancia a zonas, por ejemplo que se corresponden con áreas exteriores al pulmón. Otro punto a favor de esta herramienta es que por lo general hemos detectado que genera un mapa con las importancias con una extensión mayor por los pulmones cuando se trata de neumonía vírica que cuando se trata de neumonía bacteriana. Esto coincide con la descripción de la neumonías víricas y bacterianas que habíamos anticipado cuando describíamos el problema de clasificación al comienzo del informe.

Con todo, pese a que hemos estado justificando la necesidad de un sistema capaz de explicar las decisiones que toma en este caso nuestro clasificador, creemos que la explicabilidad en nuestro modelo puede mejorarse y no es del todo completa, sobretodo en algunas imágenes en las que los mapas de calor parecen menos aclarativos (zonas asignadas como importantes de dudosa interpretación). Además, en aquellas imágenes en las que los mapas parecen ser más informativos (Figuras 9b y 9c) sería conveniente contrastar esta información con la opinión de un neumólogo a la cual no hemos podido tener acceso, o con una especie de “ground truth” que señalara las zonas más relevantes.

## 6. Conclusiones

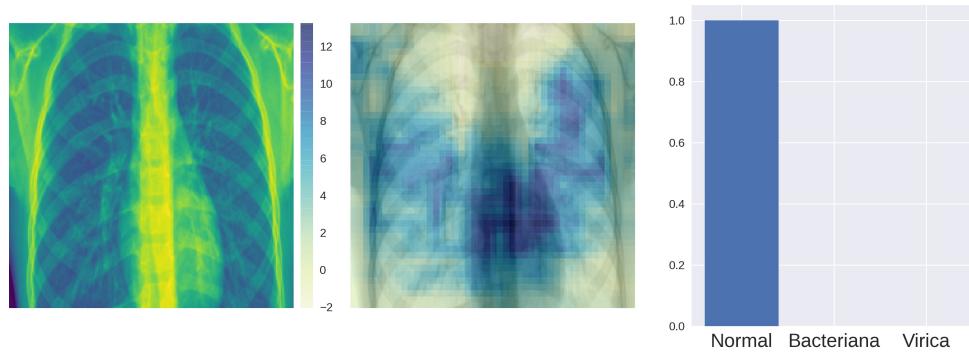
A lo largo de este trabajo nos hemos propuesto distintas metas que hemos ido superando progresivamente. En primer lugar, hemos conseguido ajustar un modelo capaz de distinguir entre radiografías de pacientes con neumonía y pacientes sanos con un 92 % de acierto. Este era el objetivo más importante ya que sería la principal fuente de ayuda para un médico a la hora de diagnosticar a pacientes con neumonía y la herramienta que más agilizaría el proceso de diagnóstico.

Por otro lado, nos gustaría tener un sistema que mostrara a los doctores que partes de la radiografía son las que el modelo considera mas importantes para realizar la clasificación. Con esto conseguiríamos ayudar a los doctores indicándoles cuales son las partes de la radiografía más interesantes para realizar el diagnóstico. Hemos conseguido alcanzar este objetivo utilizando técnicas de ocultación sobre las radiografías mediante la librería *Captum*. Este método nos proporciona un mapa de calor que nos indica que zonas de la radiografía considera el modelo más importantes para realizar la clasificación.

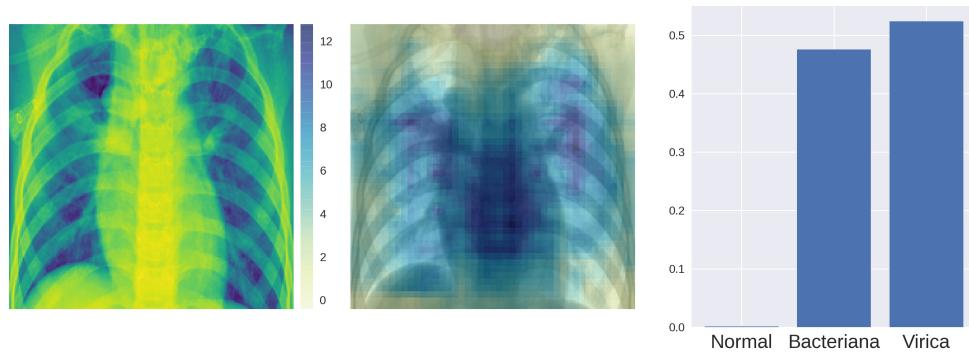
Finalmente, ya que los tratamientos que se ofrecen a un enfermo de neumonía de tipo bacteriana y a un enfermo de neumonía de tipo vírica son muy distintos, hemos considerado oportuno realizar un segundo modelo encargado de clasificar radiografías de neumonía en tipo bacteriana y tipo vírica. Hemos conseguido superar este objetivo y hemos construido una “pipeline” con ambos modelos encadenados que recibe radiografías y las clasifica en normal (paciente sano), neumonía bacteriana y neumonía vírica. Esta “pipeline” consigue un 86 % de acierto.

Con todo esto consideramos que hemos construido una herramienta que puede ser de gran utilidad para un médico a la hora de diagnosticar a pacientes con neumonía a partir de radiografías. Esta herramienta puede tener especial interés en países en vías de desarrollo donde el ratio de habitantes por cada médico especializado es abrumador y la cantidad de trabajo de diagnóstico que recibe un médico es desbordante. Una herramienta como la propuesta en este trabajo ayudaría a agilizar en gran medida el trabajo del médico.

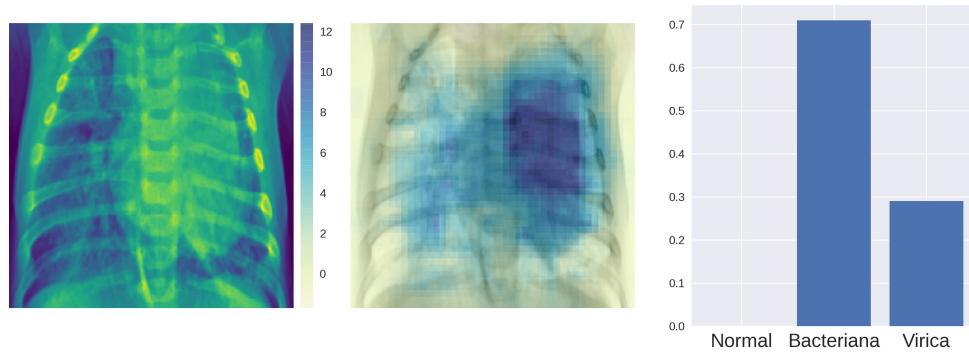
Proporcionamos todo el código utilizado en este trabajo, así como un script para realizar diagnósticos automáticos como las de la Figura 9 de todas las radiografías que se incluyan en una determinada carpeta en el siguiente repositorio de GitHub: [https://github.com/ifayost/cdb\\_pneumonia](https://github.com/ifayost/cdb_pneumonia).



(a) Radiografía de paciente sano.



(b) Radiografía de paciente con neumonía vírica.



(c) Radiografía de paciente con neumonía bacteriana.

Figura 9: Predicciones y mapas de calor obtenidos por modelo.

## Referencias

- [1] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122 – 1131.e9, 2018.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey 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.
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.
- [5] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Jonathan Reynolds, Alexander Melnikov, Natalia Lunova, and Orion Reblitz-Richardson. Pytorch captum. <https://github.com/pytorch/captum>, 2019.