



UNIVERSITAT OBERTA DE CATALUNYA (UOC)
MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS (*Data Science*)

TRABAJO FINAL DE MÁSTER

ÁREA: 3

Diagnóstico e identificación de la enfermedad de neumotórax

Autor: Javier Martínez Delgado

Tutor: Jordi de la Torre Gallart

Profesor: Ferran Prados Carrasco

Barcelona, 26 de agosto de 2022

Créditos/Copyright

Una página con la especificación de créditos/copyright para el proyecto (ya sea aplicación por un lado y documentación por el otro, o unificadamente), así como la del uso de marcas, productos o servicios de terceros (incluidos códigos fuente). Si una persona diferente al autor colaboró en el proyecto, tiene que quedar explicitada su identidad y qué hizo.

A continuación se ejemplifica el caso más habitual, aunque se puede modificar por cualquier otra alternativa:



Esta obra está sujeta a una licencia de Reconocimiento - NoComercial - SinObraDerivada
3.0 España de Creative Commons.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Diagnóstico e identificación de la enfermedad de neumotórax
Nombre del autor:	Javier Martínez Delgado
Nombre del colaborador/a docente:	Jordi de la Torre Gallart
Nombre del PRA:	Ferran Prados Carrasco
Fecha de entrega (mm/aaaa):	06/2022
Titulación o programa:	Máster en Ciencia de Datos
Área del Trabajo Final:	TFM-Área 3
Idioma del trabajo:	Español
Palabras clave	Clasificación de imágenes, segmentación de imágenes , deep learning

Resumen

Pneumothorax is a common problem produced by the entrance of air in the pleural space, making the lung unable to expand. For diagnosis, X-rays are often used, so this project will develop an image classification model to determine whether or not a patient has a pneumothorax and, if so, a segmentation model to indicate where it is located on the X-ray.

El neumotórax es un problema común producido por la entrada de aire al espacio pleural, haciendo que el pulmón no se pueda expandir. Para su diagnóstico se suele utilizar radiografías por lo que en este proyecto se realizará un modelo de clasificación de imágenes para determinar si un paciente presenta o no neumotórax y en caso afirmativo, un modelo de segmentación para indicar en qué parte de la radiografía se encuentra.

Palabras clave: Clasificación de imágenes, Segmentación de imágenes, Deep Learning, Neumotórax, Trabajo Final de Máster

Índice general

Resumen	v
Índice	vii
Llistado de Figuras	ix
Listado de Tablas	1
1. Introducción	3
1.1. Descripción de la propuesta y justificación del interés y la relevancia de la propuesta	3
1.2. Explicación de la motivación personal	3
1.3. Objetivos	3
1.4. Metodología empleada en el desarrollo del proyecto	4
1.5. Planificación del proyecto	4
1.6. Introducción al neumotórax	5
1.6.1. Qué es el neumotórax	5
1.6.2. Causas del neumotórax	5
1.6.3. Síntomas y diagnóstico	6
1.6.4. Tratamiento	6
2. Estado del arte	9
2.1. Clasificación de imágenes	9
2.2. Segmentación de imágenes	9
2.3. Función de pérdidas	10
2.4. Conclusión	10
3. Redes Neuronales Artificiales	11
3.1. La neurona	11
3.2. Aprendizaje de una red neuronal artificial	13
3.2.1. Gradiente descendente	13

3.2.2. Propagación hacia atrás o backpropagation	15
4. Redes Neuronales Convolucionales	17
4.1. Operación de convolución	17
4.2. Capa rectificadora lineal unitaria o ReLU	18
4.3. Capa de Max Pooling o agrupamiento máximo	19
4.4. Capa de aplanamiento o flattening	20
4.5. Capa completamente conectada	20
4.6. Modelos utilizados en el proyecto	21
4.7. Modelo de clasificación	21
4.7.1. Redes neuronales residuales o ResNet	22
4.7.2. Aprendizaje por transferencia	22
4.7.3. Indicadores para evaluar la clasificación	24
4.8. Modelo de segmentación	25
5. Implementación	27
5.1. Importar las librerías y el dataset	27
5.2. Análisis de datos exploratorio o EDA	28
5.2.1. Data augmentation	31
5.3. Modelo de clasificación	31
5.4. Modelo de segmentación	32
6. Resultados y vistas de futuro	33
6.1. Resultados	33
6.2. Vistas a futuro	35
7. Conclusión	37
Bibliografía	37

Índice de figuras

1.1. Diagrama de Gantt de la planificación del proyecto	4
3.1. Estructura de una neurona artificial [11]	13
3.2. Gradiente descendente [13]	14
3.3. Gradiente descendente estocástico [16]	15
4.1. Operación de convolución [19]	18
4.2. Capa rectificadora lineal unitaria [22]	18
4.3. Capa de agrupamiento máximo o Max Pooling [24]	19
4.4. Capa de aplanamiento o flattening [25]	20
4.5. Capa completamente conectada [26]	20
4.6. Red neuronal convolucional completa [27]	21
4.7. Funcionamiento de las redes neuronales residuales o ResNet [30]	22
4.8. Funcionamiento del aprendizaje por transferencia [31]	23
4.9. Arquitectura de la Res-UNet [32]	26
5.1. Número de imágenes con y sin máscara	29
5.2. Visualización de imágenes con y sin máscara	31
6.1. Resultados del modelo de segmentación	35

Índice de cuadros

6.1. Resultados de los diferentes hiperparámetros para el modelo de clasificación en el conjunto de validación.	33
---	----

Capítulo 1

Introducción

1.1. Descripción de la propuesta y justificación del interés y la relevancia de la propuesta

En este proyecto se van a realizar dos redes neuronales, la primera será una red neuronal convolucional para detectar si un paciente tiene o no neumotórax a través de imágenes de rayos x del tórax y la segunda red neuronal será una UNet para realizar la segmentación en aquellas imágenes que hayan dado positivo en la anterior, de manera que muestre claramente dónde está la evidencia de neumotórax. Esto es de gran importancia en la diagnosis médica ya que una red neuronal puede ayudar al médico a detectar un neumotórax que sin la red neuronal podría no haber visto, realizando una diagnosis temprana del paciente, evitando un posible falso negativo en la diagnosis y llegando incluso a poder salvar la vida de algún paciente.

1.2. Explicación de la motivación personal

Personalmente me interesa realizar este proyecto ya que vengo de la carrera de biología y me parece que es una herramienta muy útil como ayuda a los médicos para una correcta y temprana diagnosis.

1.3. Objetivos

El objetivo principal de este proyecto es desarrollar modelos de aprendizaje automático con el fin de ayudar en la toma de decisiones sobre el diagnóstico de pacientes. Se crearán modelos para las dos tareas principales, uno para la clasificación de las imágenes en la que se detectará si un paciente presenta neumotórax y otro para la segunda tarea en la que se segmentará

la imagen para determinar en que lugar se encuentra el neumotórax y ayudar al médico a realizar su diagnóstico. Para esto, contaremos con cuatro subobjetivos: estudiar el estado del arte para clasificación y segmentación de imágenes, estudiar los datos y llevar a cabo un análisis exploratorio, crear el modelo de clasificación y comprobar sus resultados y por último crear el modelo de segmentación y comprobar sus resultados

1.4. Metodología empleada en el desarrollo del proyecto

El proyecto partirá de un conjunto de imágenes con y sin neumotórax que se pasarán a la primera red neuronal que será una ResNet50. Si la salida de esta red es que la imagen presenta neumotórax, la imagen se pasará a la segunda red neuronal, una ResUNet, que se encargará de predecir la máscara y de mostrar en qué parte de la imagen se presenta el neumotórax. Se ha elegido estas dos redes neuronales ya que son el Estado del arte. También se utilizará aprendizaje por transferencia desde la ResNet50 ya que tiene una arquitectura ya creada y se ha entrenado con muchísimas imágenes por lo que se obtendrá un mejor rendimiento del modelo.

1.5. Planificación del proyecto

El proyecto se divide en cuatro fases: La primera fase se puede realizar al mismo tiempo que la segunda, se trata de leer sobre el estado del arte para decidir qué modelos son los más punteros para lo que se quiere hacer, mientras que la segunda fase es preparar los datos y hacer un análisis exploratorio. La segunda fase es entrenar el modelo de clasificación y evaluar su resultado y la cuarta fase es entrenar el modelo de segmentación y evaluar los resultados.

	Feb	Mar	Abr	May	Jun
Definición del trabajo					
Estado del arte					
Implementación					
Redacción de la memoria					
Defensa					

Figura 1.1: Diagrama de Gantt de la planificación del proyecto

1.6. Introducción al neumotórax

1.6.1. Qué es el neumotórax

El neumotórax es un problema común producido por la entrada de aire al espacio pleural, es decir, el espacio existente entre el pulmón y la pared torácica lo que hace que el pulmón no se pueda expandir ya que durante la respiración, las presiones en el espacio pleural son negativas para permitir la expansión del pulmón durante la inspiración, por lo que al entrar aire en este espacio, ejercerá una presión que no permitirá al pulmón expandirse dificultando la respiración [1]. Dependiendo de la cantidad de aire que entre al espacio pleural el neumotórax podrá ser pequeño y no necesitar tratamiento o bien ser un peligro inmediato para la salud de la persona [1].

Existe un tipo de neumotórax muy peligroso denominado neumotórax a tensión que se da cuando el aire presente en el espacio pleural hace colapsar los vasos sanguíneos que devuelven la sangre al corazón [1].

En cuanto a los tipos de neumotórax, se suelen clasificar en dos tipos: neumotórax espontáneo primario (PSP) si se da en pulmones que aparentemente están bien y neumotórax espontáneo secundario (SSP) si se da en pulmones afectados [2][3].

1.6.2. Causas del neumotórax

Para que el aire llegue al espacio pleural deberá haber sucedido alguna de las siguientes tres cosas [2]:

- Que se haya establecido una conexión entre los alvéolos y el espacio pleural.
- Comunicación entre la atmósfera y el espacio pleural ya sea de forma directa o indirecta. Por ejemplo: traumatismo por costilla rota, apuñalamiento, herida de bala o una complicación de una cirugía de biopsia de pulmón [1].
- Que el gas presente en el espacio pleural proceda de microorganismos.

Existen unos factores que aumentan la probabilidad de la persona a padecer un neumotórax como ser hombre, tener una estatura alta, tener un índice de masa corporal bajo, ser fumador,

estar sometidos a cambios de presión atmosférica o estar expuesto a música alta [2][3][4]. También se ha descubierto que hay un 10 % de factor familiar para los neumotórax espontáneos primarios [3].

También puede ser favorecido por otras enfermedades como la enfermedad pulmonar obstructiva crónica (EPOC), fibrosis quística, tuberculosis, cáncer pulmonar y neumonía bacteriana [2].

1.6.3. Síntomas y diagnóstico

En cuanto a los síntomas, suelen presentar dificultad respiratoria o disnea, dolor torácico, tos, taquicardia, cianosis e hipoaxemia [1][2].

Si el neumotórax es lo suficientemente grande, se podrá diagnosticar mediante radiografías de tórax mientras que si son pequeños deberá diagnosticarse mediante tomografía computarizada o escáner ultrasonido.[1]

1.6.4. Tratamiento

El tratamiento se divide en dos fases. En la primera fase se retirará el aire del espacio pleural usando una aguja y en la segunda parte se evitará que vuelva a suceder de forma recurrente [1][4]. En caso de ser muy pequeño el neumotórax, se vigilará para que no aumente su tamaño pero no necesitará intervención quirúrgica [1].

Con el fin de que no haya recurrencia, se intentará tapar cualquier agujero que pueda haber en los pulmones o la pleura. Para ello, se llevará a cabo abrasión, pleurectomía, pleurodesis y cobertura de las grapas con un parche [4]. Los métodos más utilizados son:

- **Parche de sangre propia:** No produce inflamación ni dolores fuertes. Podría producir fiebre durante unas horas. En pocas ocasiones se ha detectado una obstrucción por coágulo de sangre [4].
- **Picibanil:** Es una mezcla liofilizada de *Streptococcus pyogenes*. Podría producir fiebre y leves dolores torácicos [4].

- **Lodopovidone:** Se trata de un agente antiséptico. Es seguro y efectivo. Puede producir fiebre y leves dolores torácicos [4].
- **Materiales biodegradables:** Se cubre las grapas de la cirugía con un material biodegradable que va desapareciendo con el tiempo. Aumenta el coste de la cirugía debido a los materiales utilizados [4].
- **Polvos de talco:** Es un irritante químico que puede producir granulomas, disminución de la función pulmonar, inflamación y adhesión de las paredes pulmonares produciendo un cambio de la función pulmonar [4].
- **Tetraciclina y minociclina:** Son irritantes. La tetraciclina está prohibida en algunos países mientras que la minociclina es un derivado y es casi igual de efectiva sin tener tantos efectos negativos aunque al ser un irritante producirá dolor de pecho fuerte durante unas horas de forma inmediata [4].

Capítulo 2

Estado del arte

2.1. Clasificación de imágenes

Uno de los problemas que siempre ha surgido a la hora de entrenar redes neuronales profundas es su alto coste computacional pero tienen un mejor rendimiento que redes neuronales menos profundas. Al entrenar las redes neuronales profundas, se ha encontrado que la exactitud se mantiene constante un tiempo y comienza a descender rápidamente [5]. Al principio se pensaba que era debido al sobreentrenamiento pero más tarde se descubrió que al añadir más capas ocultas, el error de entrenamiento era mayor [5]. Esto se conoce como desaparición de gradiente, pudiendo llegar en algunos casos a explosión del gradiente y se trata de que el algoritmo se vuelve inestable en las primeras capas dificultando el aprendizaje de las neuronas.

Para solucionar este problema, se ha creado la red neuronal residual o ResNet. Se trata de la misma estructura pero saltándose una o más capas lo que la hace más fácil de entrenar ya que se puede seguir utilizando retropropagación mediante gradiente descendente estocástico y sin añadir parámetros adicionales, por lo que es más fácil de optimizar, puede tener muchas capas sin perder exactitud y tienen un gran poder de generalización en datos nuevos [5].

En este proyecto, utilizaremos una red neuronal residual de 152 capas que se ha entrenado con el dataset de ImageNet y ha ganado varias competiciones.

2.2. Segmentación de imágenes

Al hablar de clasificación de imágenes, se suele hablar de redes neuronales convolucionales, RNC o CNN para determinar si una imagen pertenece a una clase o a otra [6]. Pero en problemas relacionados con la imagen médica y diagnósticos necesitamos conocer el punto exacto en

el que se encuentra el problema para ayudar a los profesionales de la salud en un temprano y correcto diagnóstico ya que se ha demostrado que las redes neuronales pueden detectar cosas que el ojo humano no puede.

Con este fin, se ha creado una red neuronal con estructura casi simétrica en forma de 'u', lo que le da el nombre de U-Net y se trata de repetidas convoluciones de 3x3 seguidas de la función de activación ReLU y de la función de max-pooling de 2x2 creando un encoder y un decoder [6].

En los últimos años se ha creado una U-Net residual o ResUNet añadiéndole a la arquitectura que hemos hablado antes las conexiones residuales por lo que al igual que pasaba en el caso anterior, se reduce la pérdida y explosión de gradiente [7].

2.3. Función de pérdidas

La ResUnet que hemos comentado, utiliza una función de pérdidas que recibe el nombre de Dice ya que puede ser utilizada en variables continuas, aumenta la velocidad del entrenamiento de la red neuronal y tiene un buen rendimiento incluso si las clases están desbalanceadas [7].

Con el fin de mejorar el rendimiento de la red con clases desbalanceadas, se ha creado una función de pérdida denominada Tversky que, además, consigue un mejor balance entre precisión y exhaustividad o recall [8].

2.4. Conclusión

Por lo que hemos visto hasta el momento, las soluciones más relevantes que se han descubierto para problemas similares (de detección de problemas médicos en imágenes) están basadas en un clasificador que suele ofrecer unos mejores resultados al utilizar un modelo resnet ya que se evita el desvanecimiento por gradiente. En cuanto al modelo de segmentación se suele utilizar un modelo res-Unet con función de pérdida tversky.

Capítulo 3

Redes Neuronales Artificiales

3.1. La neurona

La neurona es la unidad básica de las redes neuronales artificiales y recibe este nombre ya que se presenta una estructura similar a las neuronas reales con el fin de imitar el proceso de aprendizaje del ser humano. En la neurona real podemos diferenciar unas dendritas mediante las que la neurona recibirá información desde otras neuronas o desde neuronas especializadas denominadas receptores, un cuerpo o soma donde se procesa la información y un axón que tendrá una longitud mayor que las dendritas y servirá para enviar la información a otra neurona mediante el proceso de sinapsis que podrá ser tanto química, siendo la más común, como eléctrica.

En las neuronas artificiales, la información que llega a una determinada neurona procederá de la capa de entrada o de otras neuronas en el caso de que nuestra neurona se encuentre dentro de las denominadas capas ocultas de la red neuronal artificial. Por lo tanto, la primera capa de la red neuronal artificial, denominada capa de entrada, el número de neuronas será el número de observaciones del conjunto de datos ya que cada neurona procesará una única observación, esto implicará que las entradas de una neurona serán las variables independientes de dicha observación del conjunto de datos. Con el fin de que no haya una entrada, o variable independiente, predominante que domine sobre el resto de entradas o variables se suelen normalizar o estandarizar antes de introducirlos en la red neuronal artificial, lo que hace que el algoritmo no tienda a tener overfitting en algunos valores y por lo tanto sea más estable. Es preferible realizar la estandarización, es decir, media cero y desviación típica igual a uno [9].

El valor de salida de una neurona será el valor de salida de la misma observación cuyas variables independientes se han introducido en la entrada de la neurona y podrá ser de distintos

tipos como un valor continuo, un valor binario o incluso categórico.

En cuanto a lo que sería la sinapsis, en las redes neuronales artificiales se ha establecido una serie de pesos, un peso por entrada. Se trata de coeficientes cuyo valor óptimo deberá de encontrarlo la red neuronal en su entrenamiento mediante el proceso de gradiente descendente y servirá como ponderación de los valores de entrada de la neurona.

En el soma de la neurona, donde se procesa la información, lo primero que se hace es realizar la suma ponderada de las entradas por sus pesos para obtener un único valor al que se le aplicará una función de activación para determinar si la neurona se activa o no. En caso de que la neurona sea activada, la información se transmitirá hacia la siguiente neurona. Si la red neuronal artificial sólo cuenta con una neurona se denomina perceptrón.

Ya que hemos hablado de la función de activación de una neurona, podemos destacar cuatro tipos predominantes de función de activación [10]:

- **Función umbral o escalón:** Si el resultado de la suma ponderada es cero o positivo, la neurona se activará.
- **Función sigmoidea:** Se trata de una función umbral suavizada. Nos ofrece una probabilidad de activación de la neurona.
- **Función rectificadora o ReLu:** Muy popular en redes neuronales artificiales ya que cualquier entrada negativa la transforma en cero y cualquier entrada positiva se conserva igual.
- **Función tangente hiperbólica o tanh:** Muy similar a la función sigmoidea pero en lugar de encontrarse entre cero y uno se encuentra entre menos uno y uno. Útil si se trabaja con valores negativos

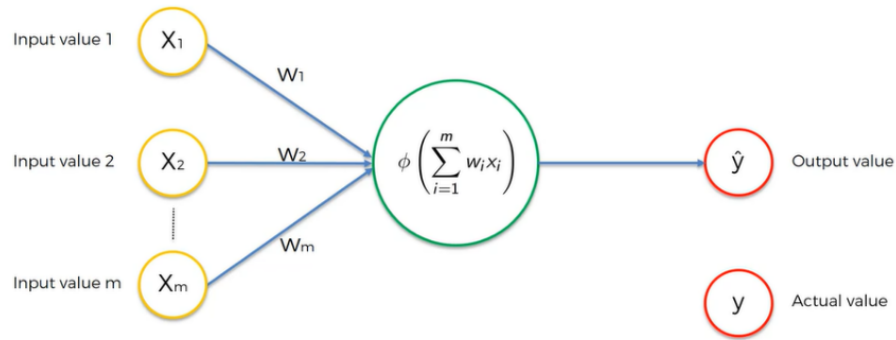


Figura 3.1: Estructura de una neurona artificial [11]

3.2. Aprendizaje de una red neuronal artificial

Para que una red neuronal artificial aprenda primero deberemos pasarle los datos de entrada a la red que, como hemos indicado, realizará la suma ponderada de las entradas y aplicará una función de activación al resultado generando una salida del perceptrón. Esta salida de la red neuronal será comparada con el valor real y se calcula la diferencia entre los dos valores, el predicho por la red y el real, mediante la función de coste. Por tanto, el objetivo principal del entrenamiento de la red neuronal artificial será minimizar esta función de coste y por tanto, hacer que la salida de la red neuronal sea lo más cercana posible al valor real [12].

El valor de la función de coste vuelve a entrar en la red neuronal y se va propagando hacia atrás con el fin de ir cambiando el valor de los pesos de la red neuronal artificial. Este proceso es conocido como propagación hacia atrás o backpropagation.

3.2.1. Gradiente descendente

El método de gradiente descendente se utiliza para buscar el mínimo global de la función de coste. Para ello, se calculará el ángulo de inclinación de la función de coste en un punto calculando la tangente mediante derivadas determinando cuál es la dirección de máxima bajada y cuáles son los pesos que hay que cambiar de una fase a otra. Para llevar a cabo el proceso de gradiente descendente de manera óptima se requiere que la función de coste sea una función convexa.

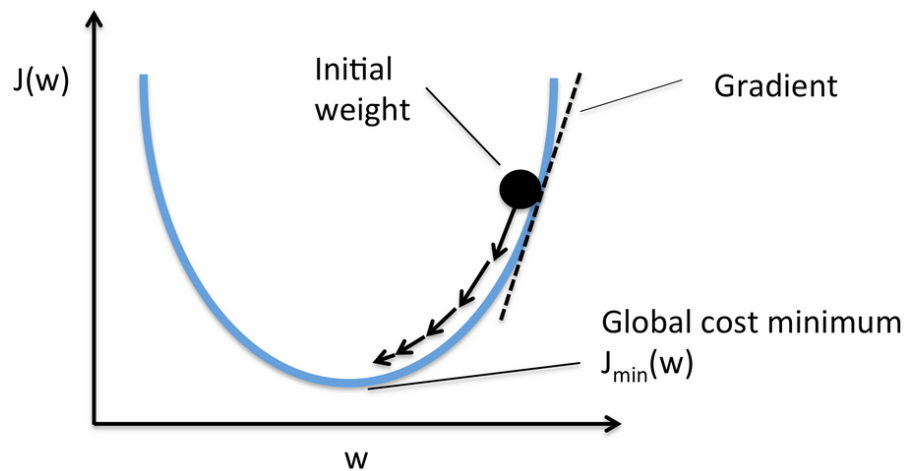


Figura 3.2: Gradiente descendente [13]

Gradiente descendente estocástico

En caso de que la función de coste no sea convexa, podría obtenerse mediante gradiente descendente un mínimo local pero no llegar a un mínimo global por lo que la red neuronal artificial no estaría optimizada. En el gradiente descendente calcula la función de coste global y actualiza los pesos por bloques tras haberle pasado todos los datos mientras que en el gradiente descendente estocástico calcula la función de coste y actualiza los pesos tras cada observación [14]. Esto ayudará a obtener un mínimo global pero tendrá más coste computacional. Por esto, se suele utilizar el gradiente descendente en lotes pequeños o mini-batches que creará lotes de observaciones y actualizará los pesos después de cada lote [15].

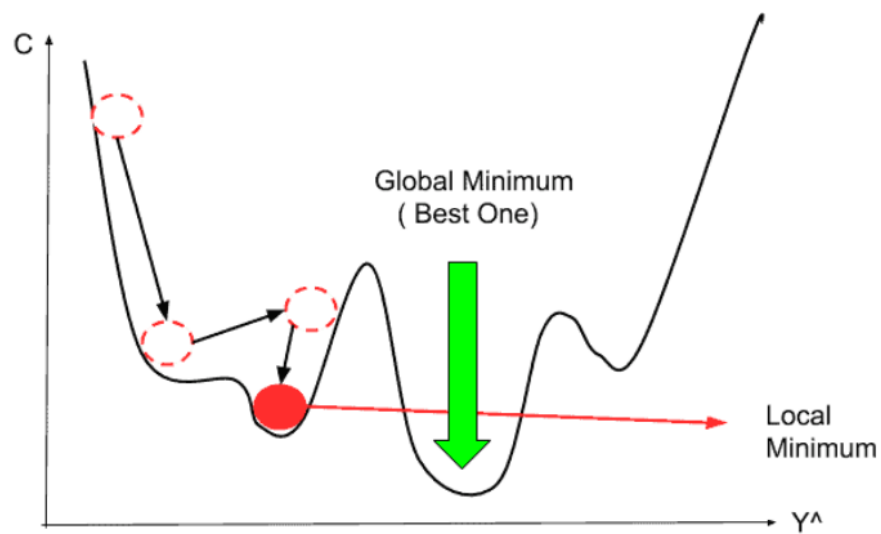


Figura 3.3: Gradiente descendente estocástico [16]

3.2.2. Propagación hacia atrás o backpropagation

Como hemos dicho antes, la propagación hacia atrás nos permite, a partir de la función de coste, ir hacia atrás en la red neuronal artificial y actualizar los pesos corrigiendo los errores causados por los pesos.

Capítulo 4

Redes Neuronales Convolucionales

Una red neuronal convolucional es una red neuronal a la que se le pasa una imagen de entrada y su salida es una etiqueta de categoría. Para esto, se convertirá la imagen en un array bidimensional, en el caso de que la imagen de entrada sea en blanco y negro, o en un array tridimensional en el caso de que la imagen de entrada sea en color. Si tenemos una imagen de 32×32 píxeles tendremos 1024 píxeles en una foto en blanco y negro mientras que si la foto es en color tendríamos $32 \times 32 \times 3 = 3072$ píxeles ya que tenemos un canal para el rojo, otro para el verde y otro para el azul, conocido como RGB. Dentro de cada array los valores estarán comprendidos entre 0 y 255 [17]. Las redes neuronales convolucionales están compuestas por varias fases distintas.

4.1. Operación de convolución

A la imagen de entrada se le aplica un kernel, también llamado detector de rasgo, filtros o núcleos de convolución, para detectar patrones concretos en la imagen. Esta detección se hace mediante una suma ponderada de la imagen original y el kernel obteniendo así una matriz o mapa de características que será una versión simplificada de la imagen original tras haber aplicado el kernel [18].

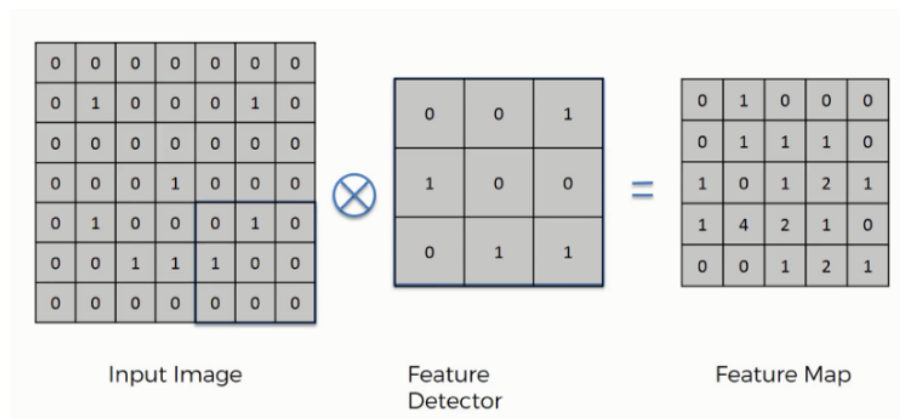


Figura 4.1: Operación de convolución [19]

Como es normal, no solo se utiliza un kernel si no que se utilizan varios kernels o filtros para buscar diferentes patrones en la imagen y así poder juntar todos los mapas de características que generan en la siguiente capa por lo que obtendremos un resumen de las características presentes en la imagen de entrada.

4.2. Capa rectificadora lineal unitaria o ReLU

Aunque se suele tratar por separado la capa rectificadora lineal unitaria y la capa de convolución, se podría tratar de forma conjunta ya que suele aplicarse una seguida de la otra. Al haberle pasado los kernels a la imagen de entrada y crear los mapas de características y siendo los kernels operaciones lineales mientras que el mundo real hay cosas que no son lineales, lo que hará la capa rectificadora será aumentar los detalles que no sean lineales [20][21].

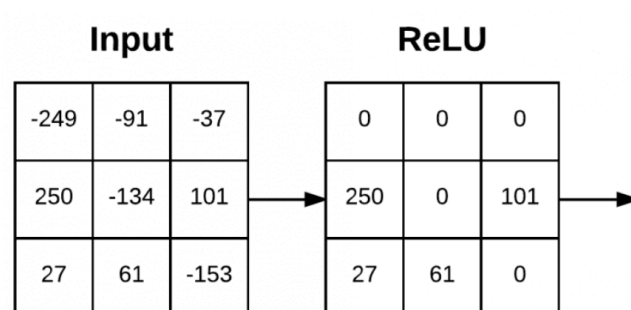


Figura 4.2: Capa rectificadora lineal unitaria [22]

4.3. Capa de Max Pooling o agrupamiento máximo

En esta capa se agrupará la información de píxeles cercanos devolviendo un mapa de características de menos dimensión mediante el uso de un kernel que esta vez, devolverá el valor máximo presente en el kernel. Esto hace que obtengamos las características más grandes detectadas por la capa de convolución pero reduciendo la dimensionalidad de la matriz reduciendo así el número de parámetros de la red neuronal reduciendo el cálculo computacional necesario. Además, este proceso ayudará a evitar el overfitting por lo que se podrá extraer patrones de forma más general [23].

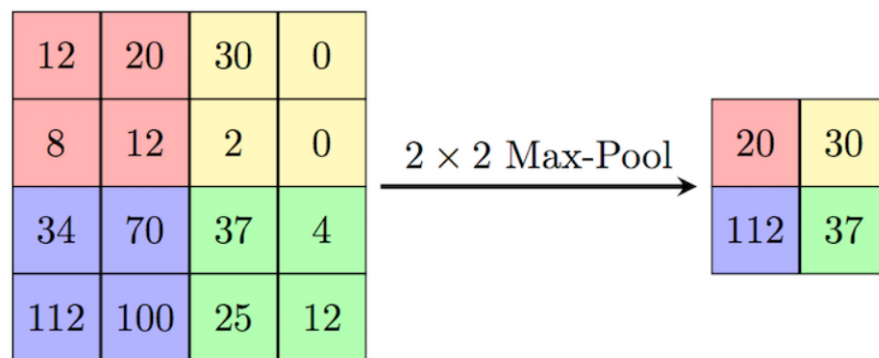


Figura 4.3: Capa de agrupamiento máximo o Max Pooling [24]

4.4. Capa de aplanamiento o flattening

En esta capa lo que se hace es simplemente aplanar la imagen creando un vector unidimensional que será la entrada de una futura red neuronal artificial.

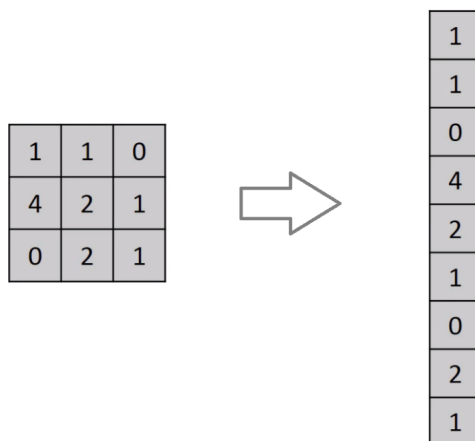


Figura 4.4: Capa de aplanamiento o flattening [25]

4.5. Capa completamente conectada

Se trata de una red neuronal artificial con capas totalmente conectadas que tiene como entrada el vector unidimensional y como salida asignará una categoría a la observación de entrada.



Figura 4.5: Capa completamente conectada [26]

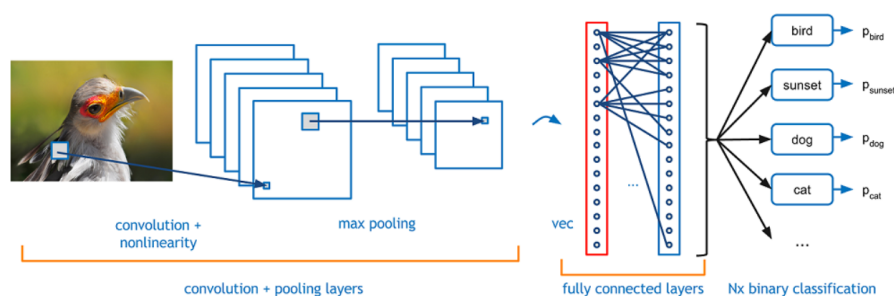


Figura 4.6: Red neuronal convolucional completa [27]

4.6. Modelos utilizados en el proyecto

Como se dijo en el apartado de objetivos, el objetivo de este proyecto es desarrollar una inteligencia artificial para reducir la carga de trabajo de la diagnosis de enfermedades a partir de imágenes al mismo tiempo que se intenta obtener mejores resultados para esa diagnosis. Para esto, deberemos implementar dos modelos: uno de clasificación que dada una imagen nos indique si el paciente presenta o no presenta un neumotórax y, en caso de que la respuesta del primer modelo sea que sí presenta un neumotórax, pasarle dicha imagen a un segundo modelo que nos indique en qué parte de la imagen se encuentra.

Para este proyecto se ha utilizado el dataset SIIM-ACR Pneumothorax Segmentation de Society for Imaging Informatics in Medicine (SIIM) que realizaron una competición en la página de Kaggle hace tres años [28]. Aunque a día de hoy ya a terminado la competición, se puede seguir accediendo a los datos en la plataforma para crear modelos. El dataset consiste en una carpeta de entrenamiento y otra de test que contendrán a su vez más carpetas con imágenes en formato DICOM (Digital Imaging and Communication in Medicine) que es un estándar en imágenes médicas. También proporciona un csv o fichero de valores separados por comas que contendrá dos variables, una para identificar la imagen y otra que, en caso de presentar neumotórax, tendrá la máscara de la imagen

4.7. Modelo de clasificación

Como vimos en la sección de las redes neuronales convolucionales, la función de coste se calcula según el error que cometemos al final por lo que al realizar la propagación hacia atrás o backpropagation, el gradiente tenderá a ir disminuyendo. Al haber tantas capas en una red convolucional profunda con el fin de obtener mejores resultados, el gradiente se hará tan pe-

queño que será incapaz de actualizar los pesos del modelo. Este problema es conocido como desvanecimiento del gradiente [29].

4.7.1. Redes neuronales residuales o ResNet

Para solucionar el problema del desvanecimiento de gradiente se crearon las redes neuronales residuales. Este tipo de red añade unos saltos, mapeos de identidad o capas de omisión de conexión combinando las dos rutas antes de la capa rectificadora.

En el momento de la propagación hacia atrás, el gradiente podrá utilizar estas capas de omisión de conexión de manera que se reduciría de forma mucho más lento que si no existiesen estas capas de omisión de conexión. Gracias a esto, se pudo entrenar una red neuronal de 152 capas evitando el problema de desvanecimiento de gradiente. En cuanto a los hiperparámetros se utilizará el optimizador Adam ya que es el que ofrece mejores resultados a la hora de identificar patrones en imágenes y se intentará buscar el mejor learning rate y decay entre un rango determinado mediante grid search manual debido a los tiempos de entrenamiento. En cuanto a la función de pérdida se utilizará categorical crossentropy ya que es la función de pérdida más utilizada en proyectos de clasificación multiclase.

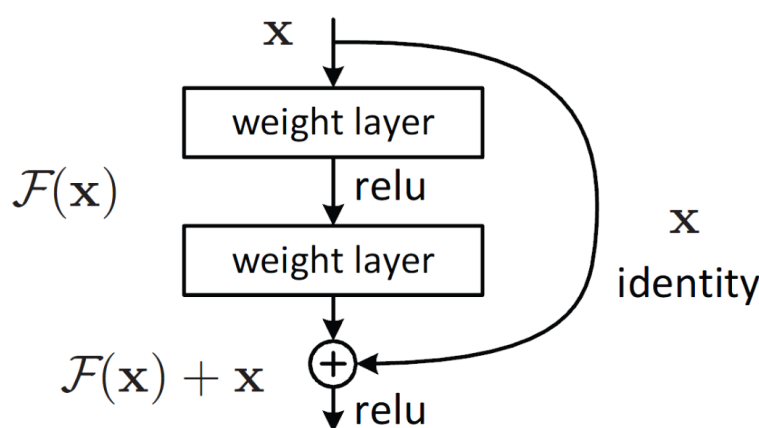


Figura 4.7: Funcionamiento de las redes neuronales residuales o ResNet [30]

4.7.2. Aprendizaje por transferencia

Se trata de una técnica de machine learning en la que una red neuronal que ya ha sido entrenada para realizar una tarea se reutiliza como punto de partida de otra red neuronal similar,

reduciendo así en gran medida el tiempo de entrenamiento. Por esto, haremos aprendizaje por transferencia de una red ResNet entrenada con el dataset de ImageNet con once millones de imágenes y capaz de predecir once mil categorías.

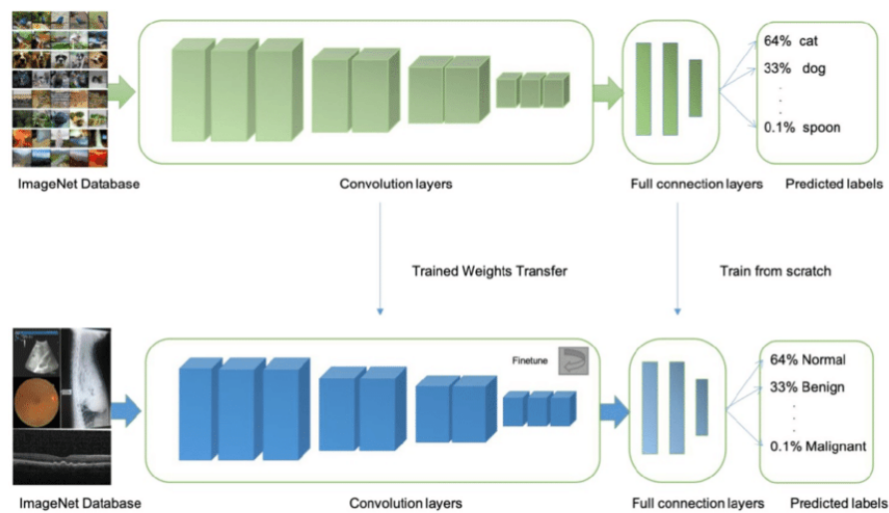


Figura 4.8: Funcionamiento del aprendizaje por transferencia [31]

4.7.3. Indicadores para evaluar la clasificación

- Exactitud = Accuracy: Conjunto de registros correctamente clasificados

$$\frac{TP + TN}{Total}$$

- Error: Conjunto de registros clasificados de forma incorrecta

$$ERR = 1 - accuracy$$

$$\frac{FP + FN}{Total}$$

- Sensibilidad = Recall = TPR: Proporción de positivos correctamente clasificados

$$\frac{TP}{P}$$

- Especificidad = TNR: Proporción de negativos correctamente clasificados

$$\frac{TN}{N}$$

- Precisión: Proporción de clasificados como positivos, que realmente lo son

$$\frac{TP}{TP + FP}$$

- F1: utiliza la sensibilidad y la precisión

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

- Curva ROC: Muestra sensibilidad y especificidad

Para este proyecto nos interesará que si diagnosticamos a un paciente con un neumotórax, realmente tenga un neumotórax y que no sea un falso positivo. Por tanto, nos interesará utilizar como métrica principal el F1-score para determinar si se está clasificando de manera correcta el que un determinado paciente presente neumotórax.

4.8. Modelo de segmentación

El modelo de segmentación que utilizamos es el Res-UNet que combina una UNet, que es una red neuronal en forma de 'U' con las capas de omisión de conexión que hemos visto en las redes neuronales residuales con el fin de solucionar el problema de desvanecimiento de gradiente.

Se trata de una red neuronal en la que la salida tiene el mismo tamaño que la entrada por lo que, al igual que los autoencoders, estará dividido en tres partes: un codificador, un cuello de botella y un decodificador.

- **Codificador:** El codificador o proceso de contracción está compuesto por una serie de bloques e irán codificando la imagen mediante operaciones de convolución y agrupamiento máximo por lo que las dimensiones de la imagen irán disminuyendo mientras que los mapas de características se irán duplicando después de cada bloque.
- **Cuello de botella:** Conecta el codificador con el decodificador permitiendo que se vuelva a recrear la imagen.
- **Decodificador:** Está compuesto por una serie de bloques al igual que el codificador pero en lugar de disminuir el tamaño de la imagen, lo aumentará. Para ello, tomará los datos de la capa anterior combinados con los datos del bloque respectivo del codificador lo que garantizará que las características aprendidas en el codificador sean utilizadas a la hora de reconstruir la imagen.

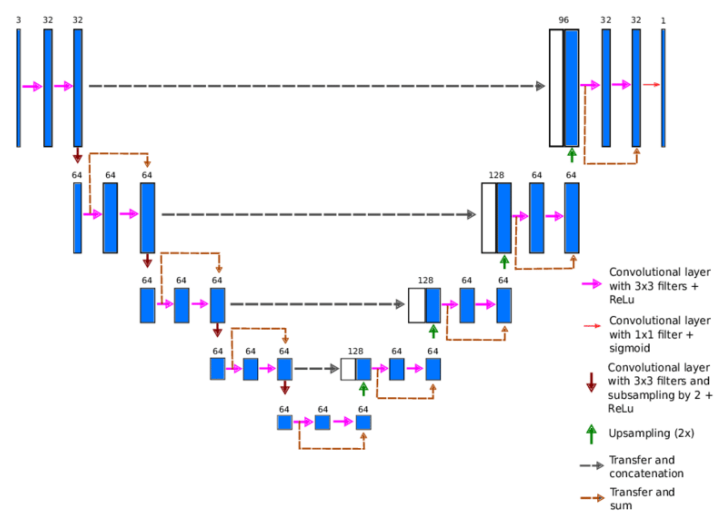


Figura 4.9: Arquitectura de la Res-UNet [32]

Capítulo 5

Implementación

5.1. Importar las librerías y el dataset

En cuanto a las librerías que debemos cargar para el proyecto, podemos encontrar:

- **Pandas:** Es una librería muy común que ofrece la manipulación de datos en forma de DataFrames.
- **Numpy:** Se suele importar junto con pandas y ofrece funciones para cálculo numérico.
- **os:** Ofrece la posibilidad de crear diferentes directorios y trabajar con ellos.
- **Matplotlib y Seaborn:** Matplotlib nos permitirá visualizar las imágenes mientras que utilizaremos Seaborn para las gráficas.
- **Scikit-learn:** Utilizaremos sklearn para normalizar los datos y para dividir el dataset en un conjunto de testing y otro de test.
- **Pydicom:** Esta librería nos permitirá trabajar con el formato DICOM de manera mucho más sencilla.

- **Tensorflow y Keras:** Tensorflow es un framework de alto nivel para machine learning y Keras es un framework de alto nivel que se ejecuta por encima de tensorflow.

A la hora de importar el dataset, podemos observar que la variable que ImageId, que identifica la imagen de la observación, no es una ruta al directorio donde se encuentra la imagen si no que se trata de un identificador numérico sin ninguna indicación de la carpeta en la que se encuentra y sin indicación del formato. Por esto, lo que tenemos que hacer es crear un bucle que itere sobre la propia carpeta de imágenes y encuentre el identificador, guardando la ruta completa en una nueva variable.

En cuanto a la otra variable, EncodedPixels, tendrá las máscaras, en caso de presentar neumotórax. Una máscara asociará a cada pixel de la imagen un valor determinado mostrando así en qué lugar se encuentra el neumotórax de forma precisa con el fin de poder entrenar nuestro modelo de segmentación de imágenes. Estas máscaras estarán en formato RLE o Run Length Encoding ya que al tener el mismo tamaño que la imagen original se ha de reducir su tamaño con el fin de disminuir el espacio que requiere. Esta técnica de compresión almacena secuencias que contienen muchos datos consecutivos como un valor único con su recuento.

5.2. Análisis de datos exploratorio o EDA

En esta sección intentaremos visualizar los datos que tenemos por lo que lo primero que haremos será ver cuántas imágenes presentan máscara y cuántas imágenes son de pacientes sanos.

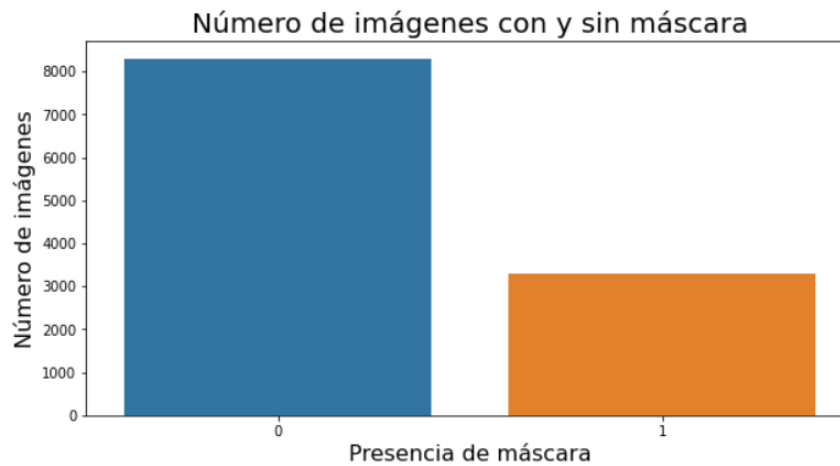
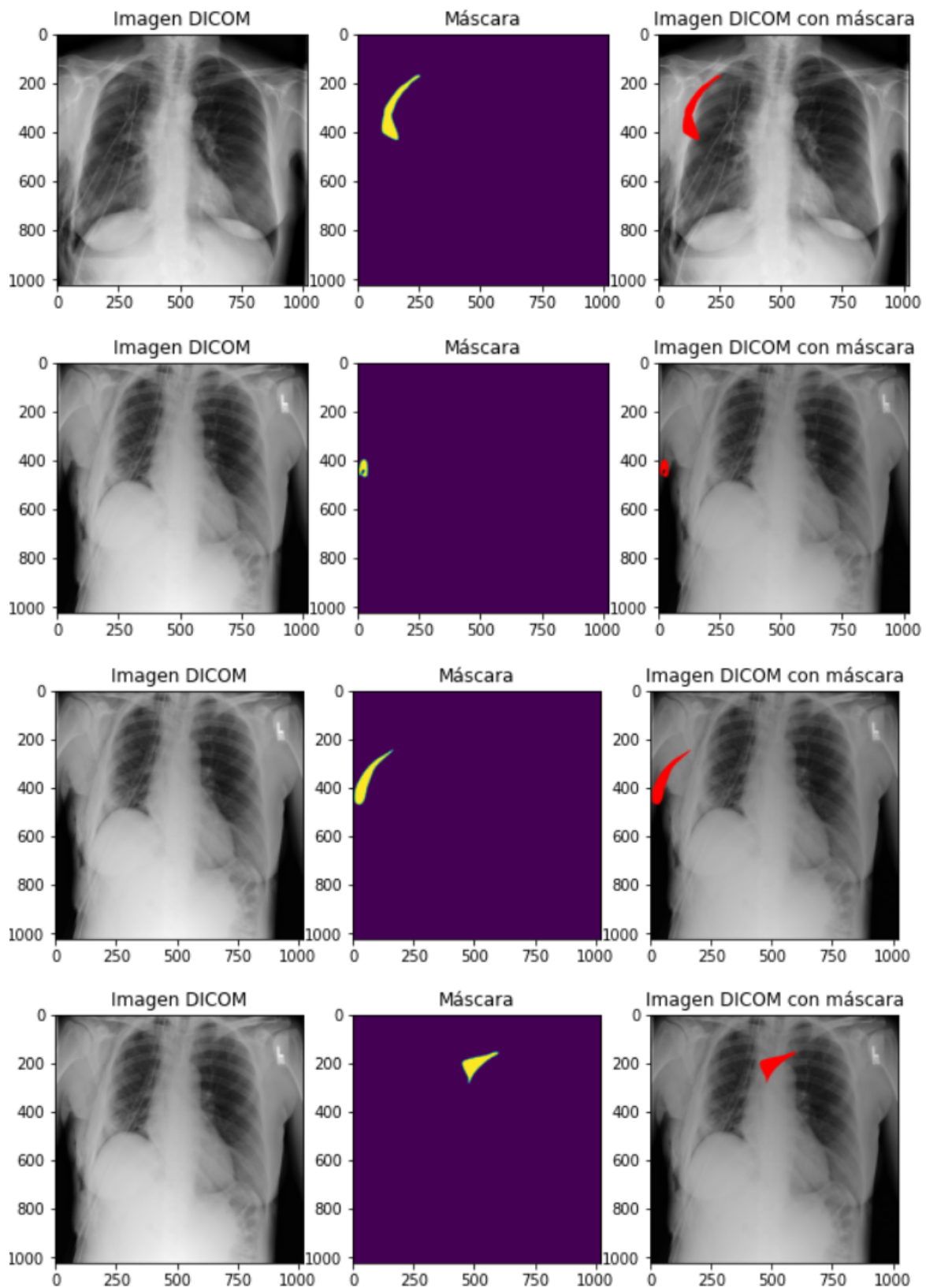


Figura 5.1: Número de imágenes con y sin máscara

Una vez visto que hay 11582 imágenes de las cuales sólo 3286 presentan máscara, procedemos a visualizar las imágenes con y sin su máscara. Para ello, no hará falta pasarlas a formato PNG o JPG ya que hemos importado la librería `pydicom` que nos permitirá trabajar de manera más sencilla con este formato. Aun así, transformaremos las imágenes a formato JPG y las guardaremos en una carpeta llamada `train-images` ya que el modelo de clasificación no permite el formato DICOM.



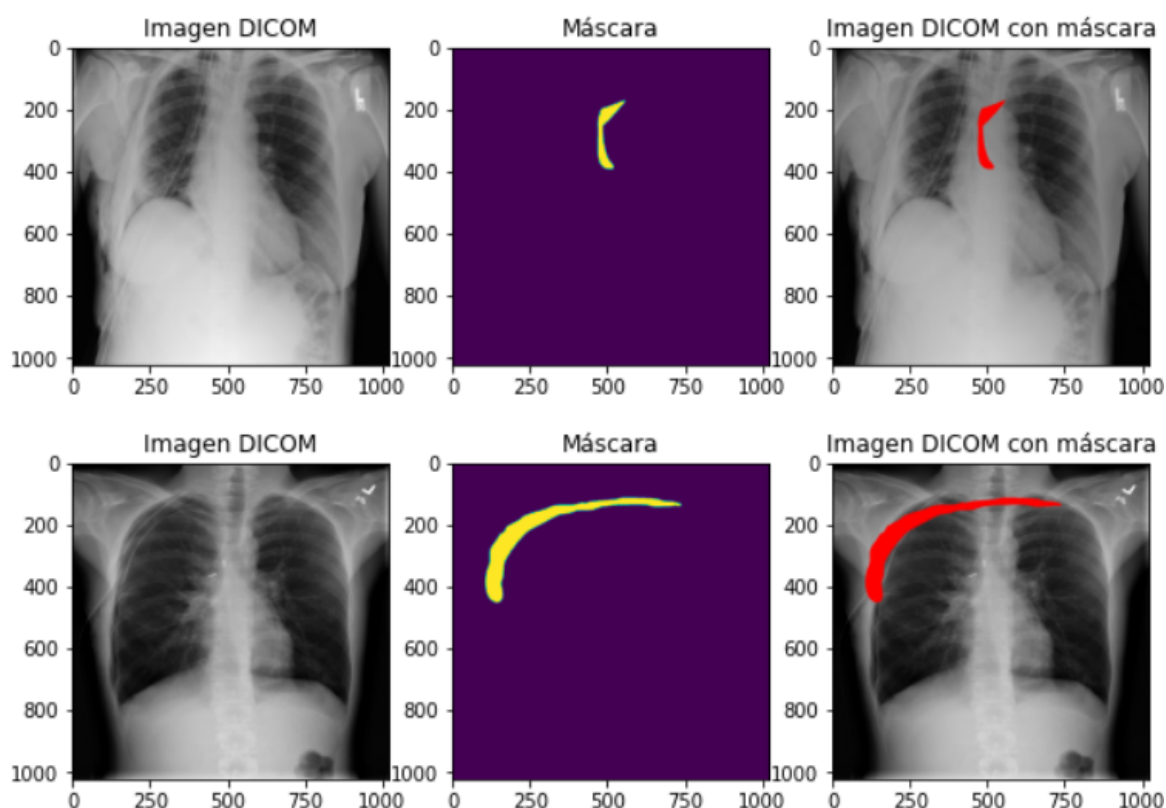


Figura 5.2: Visualización de imágenes con y sin máscara

5.2.1. Data augmentation

Mediante esta técnica podremos aumentar los datos de entrada y modificar las imágenes haciendo zoom, flip horizontal, escalado y rotación haciendo que el modelo reconozca los patrones de forma óptima y evitando el overfitting. En concreto, se ha utilizado el flip horizontal, una rotación del 5 % y un rango de zoom de 0.2 %

5.3. Modelo de clasificación

Como hemos dicho antes, para el modelo de clasificación utilizamos un modelo resnet con los pesos de imagenet y sin las últimas capas de clasificación y lo utilizamos como modelo base. Este modelo lo congelamos exceptuando las últimas diez capas y le agregamos una capa de pooling, una capa de aplanamiento, dos capas densas con función de activación relu seguidas por capas de dropout y una capa totalmente conectada con dos neuronas ya que utilizamos la función de activación softmax.

Para compilar el modelo, se utiliza la función de pérdida de categorical crossentropy ya

que se ha utilizado la función de activación softmax en la última capa del modelo y para las métricas buscaremos la accuracy. A la hora de encontrar los hiperparámetros óptimos se buscará el mejor learning rate y decay para el optimizador Adam ya que a la hora de encontrar patrones en modelos de clasificación de imágenes suele ser el más estable y por tanto el más utilizado. Una vez se tienen los mejores hiperparámetros para el optimizador Adam, se prueban otros optimizadores.

5.4. Modelo de segmentación

Para el modelo de segmentación también se utilizará el optimizador Adam y, como hemos indicado anteriormente, se utilizará la función de pérdidas tversky que también se utilizará como métrica a tener en cuenta.

Capítulo 6

Resultados y vistas de futuro

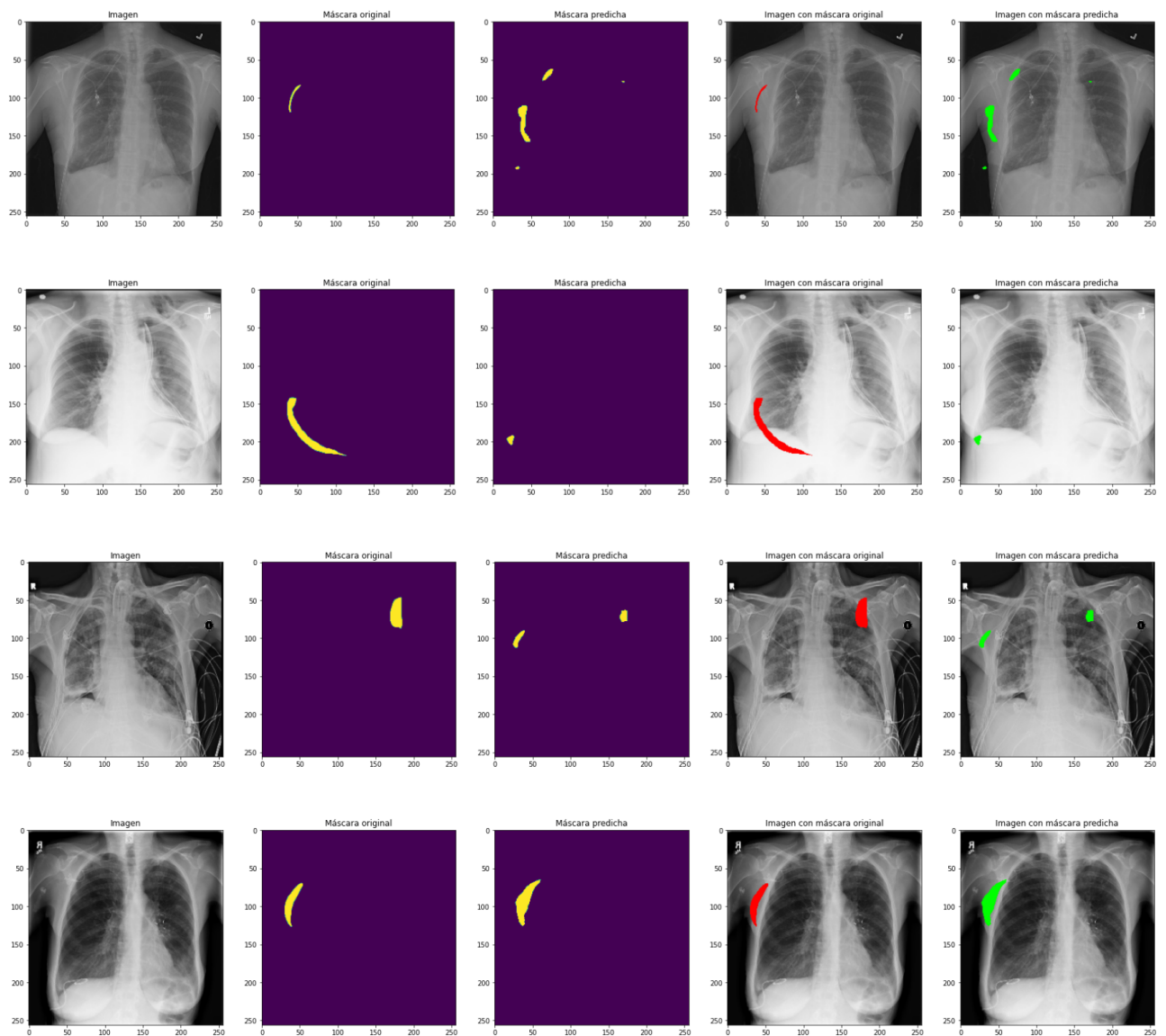
6.1. Resultados

Tras haber creado el modelo de clasificación, se ha entrenado el modelo con diferentes hiperparámetros de tasa de aprendizaje y decay para determinar qué hiperparámetros se ajustan mejor para nuestro objetivo que, como hemos dicho, será el F1-score para los pacientes con neumotórax ya que si se diagnostica un paciente con neumotórax se espera que no sea un falso positivo. Tras haber probado distintos hiperparámetros, los mejores resultados se han obtenido con una tasa de aprendizaje de 0.0001 y un decay de 0.01 como se observa en la tabla. También se ha probado el optimizador SGD pero se obtenían resultados de f1 total entorno a 0.7

Tasa de aprendizaje	Decay	F1 Sanos	F1 neumotórax	F1 Total	Tiempo
0.0001	0.01	0.90	0.78	0.87	52 m
0.001	0.0001	0.90	0.76	0.86	84 m
0.001	0.01	0.90	0.76	0.86	57 m
0.001	0.1	0.89	0.76	0.86	50 m
0.0001	0.00001	0.88	0.74	0.85	45 m
0.001	0.001	0.90	0.72	0.85	99 m
0.0001	0.001	0.90	0.72	0.84	47 m
0.0001	0.1	0.88	0.72	0.84	170 m
0.0001	0.0001	0.89	0.63	0.83	53 m
0.001	0.00001	0.88	0.54	0.81	68 m

Cuadro 6.1: Resultados de los diferentes hiperparámetros para el modelo de clasificación en el conjunto de validación.

En cuanto al modelo de segmentación, se han obtenido resultados bastante satisfactorios al comparar la máscara predicha por el modelo con la máscara original como se puede apreciar en las imágenes. En la imagen, la primera columna muestra la imagen original y a continuación se muestra la máscara original y la predicha y la imagen con la máscara original y con la máscara predicha.



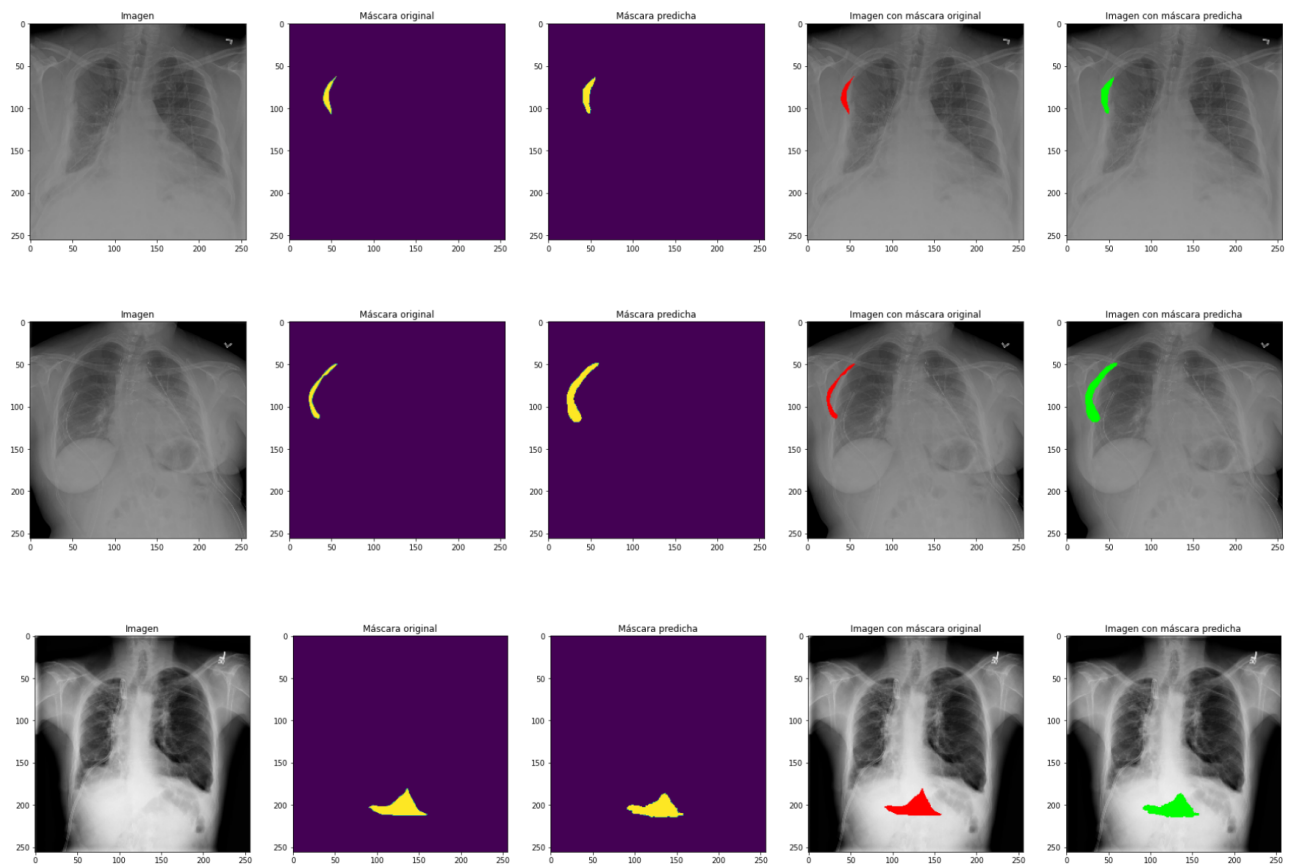


Figura 6.1: Resultados del modelo de segmentación

6.2. Vistas a futuro

El uso de inteligencia artificial en el campo de la medicina es una implementación relativamente nueva y que día a día va aumentando hasta conseguir grandes avances. En cuanto al proyecto, considero que el modelo de segmentación es aceptable para indicar al personal sanitario en qué lugar se encuentra el neumotórax aunque el modelo de clasificación sería mejorable si se tratase de aumentar el F1-score de los pacientes diagnosticados con neumotórax para evitar posibles falsos positivos.

Capítulo 7

Conclusión

El objetivo de este proyecto era identificar si un paciente presenta o no neumotórax y, en caso afirmativo, poder mostrar al personal sanitario en qué lugar se encuentra ayudando al correcto y temprano diagnóstico del paciente. Después de haber creado y entrenado los modelos considero que sus resultados son más que satisfactorios para el objetivo propuesto

Bibliografía

- [1] Jonathan B. Imran and Alexander L. Eastman. *Pneumothorax* , 2017
- [2] M. Noppen. *Spontaneous pneumothorax: epidemiology, pathophysiology and cause* , 2010
- [3] Rob Hallifax and Julius P. Janssen. *Pneumothorax Time for New Guidelines?* , 2019
- [4] Ke-Cheng Chen, Pei-Hsing Chen and Jin-Shing Chen. *New options for pneumothorax management* , 2020
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Deep Residual Learning for Image Recognition* , 2015
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation* , 2015
- [7] Foivos I.Diakogiannis, François Waldner, Peter Caccetta and Chen Wu. *ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data* , 2020
- [8] Nabila Abraham and Naimul Mefraz Khan. *A novel focal tversky loss function with improved attention u-net for lesion segmentation* , 2018
- [9] Yann LeCun, Leon Bottou, Genevieve B.Orr and Klaus-Robert Müller. *Efficient BackProp* , 1998
- [10] Xavier Glorot, Antoine Bordes and Yoshua Bengio. *Deep Sparse Rectifier Neural Networks* , 2011
- [11] Manik Soni. *Fundamentals of neural network in machine learning* , 2020
- [12] Phylliida. *A list of cost functions used in neural networks, alongside applications* , 2015
- [13] Aristides Zenonos. *Gradient Descent vs. Normal Equation for Linear Regression problems* , 2019

-
- [14] Andrew Trask. *A Neural Network in 13 lines of Python (Part 2 - Gradient Descent)* , 2015
 - [15] Michael Nielsen. *Neural Networks and Deep Learning* , 2015
 - [16] MLTut. *Stochastic Gradient Descent- A Super Easy Complete Guide* , 2020
 - [17] Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner. *GradientBased Learning Applied to Document Recognition* , 1998
 - [18] Jianxin Wu. *Introduction to Convolutional Neural Networks* , 2017
 - [19] SuperDataScience. *Convolutional Neural Networks (CNN): Step 1- Convolution Operation* , 2018
 - [20] Jay Kuo. *Understanding Convolutional Neural Networks with A Mathematical Model* , 2016
 - [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* , 2015
 - [22] Adrian Rosebrock. *Convolutional Neural Networks (CNNs) and Layer Types* , 2021
 - [23] Dominik Scherer, Andreas Müller and Sven Behnke. *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition* , 2010
 - [24] FirelordPhoenix. *Pictorial example of max-pooling* , 2018
 - [25] Caner Dabakoglu. *What is Convolutional Neural Network (CNN) ? — with Keras* , 2018
 - [26] Nushaine Ferdinand. *A Simple Guide to Convolutional Neural Networks* , 2020
 - [27] Adit Deshpande. *A Beginner's Guide To Understanding Convolutional Neural Networks* , 2019
 - [28] Society for Imaging Informatics in Medicine (SIIM). *SIIM-ACR Pneumothorax Segmentation* , 2019
 - [29] Chi-Feng Wang. *The Vanishing Gradient Problem* , 2019
 - [30] Muneeb ul Hassan. *ResNet: Residual CNNs for Image Classification Tasks* , 2019
 - [31] Amananandrai. *Pretrained Models for Transfer Learning in Keras for Computer Vision* , 2020
 - [32] Artem Sevastopolsky, Stepan Drapak, Konstantin Kiselev, Blake M.Snyder and Anastasia Georgievskaya. *Stack-U-Net: Refinement Network for Image Segmentation on the Example of Optic Disc and Cup* , 2018