Trabajo Fin de Máster

Tratamiento y análisis estadístico de radiografías de tórax de pacientes diagnosticados de SARS-CoV-2

Iria Roca Otero

Máster en Técnicas Estadísticas

Curso 2020-2021

Propuesta de Trabajo Fin de Máster

Título en galego: Tratamento e análise estatística de radiografías de tórax de pacientes diagnosticados de SARS-SoV-2
Título en español: Tratamiento y análisis estadístico de radiografías de tórax de pacientes diagnosticados de SARS-CoV-2
English title: Statistical treatment and analysis of chest radiographs from SARS-CoV-2 diagnosed patients
Modalidad: Modalidad A
Autor/a: Iria Roca Otero, Universidad de Santiago de Compostela
Director/a: Francisco Gude Sampedro, Hospital Clínico Universitario de Santiago de Compostela (CHUS); Carmen María Cadarso Suárez, Universidad de Santiago de Compostela
Breve resumen del trabajo: El objetivo principal de este trabajo es el desarrollo de modelos estadísticos que permitan predecir el riesgo de muerte de pacientes diagnosticados de SARS-CoV-2 entre Marzo y Junio de 2020, a través del estudio de las radiografías de tórax, en combinación con otros factores clínicos y analíticos.
Otras observaciones:

Don Francisco Gude Sampedro, Jefe de la Unidad de Epidemiología Clínica del Hospital Clínico Universitario de Santiago de Compostela (CHUS), doña Carmen María Cadarso Suárez, Catedrática de la Universidad de Santiago de Compostela, informan que el Trabajo Fin de Máster titulado

Tratamiento y análisis estadístico de radiografías de tórax de pacientes diagnosticados de SARS-CoV-2

fue realizado bajo su dirección por doña Iria Roca Otero para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago de Compostela, a 21 de Junio de 2021.

El director:

GUDE SAMPEDRO FRANCISCO - 33239241D

Firmado digitalmente por GUDE SAMPEDRO FRANCISCO - 33239241D
Fecha: 2021.06.22 14:08:40 +02'00'

Don Francisco Gude Sampedro

La directora:

CADARSO SUAREZ CARMEN MARIA - 33253802B

Firmado digitalmente por CADARSO SUAREZ CARMEN MARIA - 33253802B
Fecha: 2021.06.22 13:48:06 +02'00'

Doña Carmen María Cadarso Suárez

La autora:

Firmado por ROCA OTERO IRIA - 33539998H el día 22/06/2021 con un certificado emitido por AC FNMT Usuarios

Doña Iria Roca Otero

Agradecimientos

En primer lugar quiero agradecer a mi compañera Jenifer por toda su colaboración y ayuda para la realización de este trabajo, desde la extracción de los datos, el procesamiento de las imágenes, la revisión del presente texto... Sin duda no lo habría conseguido sin su ayuda, y por ello le estoy increíblemente agradecida.

Quiero agradecer también al profesor Pablo García Tahoces por orientarme en los pasos a seguir en el análisis de las radiografías, un campo completamente nuevo para mí, y en el que sin su guía habría sido muy complicado avanzar.

Y, por supuesto, agradecer a mis directores, los profesores Francisco Gude y Carmen Cardarso, por su apoyo, por sus consejos, y por darme la oportunidad de realizar este trabajo, que supuso para mí un reto muy enriquecedor, y que sin ellos no habría sido posible.

¡Gracias a todos!

Índice general

Resumen

Resumen en español

La COVID-19, la enfermedad causada por el virus SARS-CoV-2, se ha extendido por todo el mundo en un periodo de tiempo relativamente corto. Sus manifestaciones clínicas varían desde una infección asintomática hasta la neumonía, que puede derivar en un Síndrome de Dificultad Respiratoria Aguda, insuficiencia multiorgánica y, en última instancia, la muerte.

El estudio de las radiografías de tórax juega un papel muy importante a la hora de valorar la gravedad de la patología. El objetivo principal de este trabajo es el desarrollo de modelos estadísticos que permitan predecir el riesgo de muerte a través del estudio de la opacidad de las radiografías de estos pacientes, en combinación con otros factores clínicos y analíticos.

Para ello, se dispone de las radiografías digitales, así como de diversas variables clínicas, de una muestra de pacientes diagnosticados de SARS-CoV-2 en el Área Sanitaria de Santiago de Compostela entre Marzo y Junio de 2020. En este estudio se presenta una forma de extraer información relevante de estas radiografías, a través de la aplicación de metodologías propias del procesamiento y análisis de imágenes digitales, y se utilizan las características extraidas para la construcción de un modelo que nos permita predecir la probabilidad que tiene un paciente de fallecer.

English abstract

COVID-19, the disease caused by the SARS-CoV-2 virus, has spread around the world in a relatively short period of time. Its clinical manifestations range from an asymptomatic infection to pneumonia, which can lead to Acute Respiratory Distress Syndrome, multiple organ failure and, ultimately, death.

The study of chest x-rays plays a very important role in assessing the severity of the pathology. The main objective of this work is the development of statistical models that allow us to predict the risk of death through the study of the opacity of the radiographs of these patients, in combination with other clinical and analytical factors. The data available to achieve this goal are the digital radiographs and other clinical variables from a sample of patients diagnosed with SARS-CoV-2 in the Health Area of Santiago de Compostela between March and June 2020. This study presents a way to extract relevant information from these radiographs, through the application of known digital image processing and analysis methodologies, and the extracted characteristics are used in the construction of a model that allows us to predict the probability of death for any patient.

Capítulo 1

Introducción

1.1. SARS-CoV-2

A finales de 2019 se identificaron varios casos de una neumonía con origen desconocido en la región de Wuhan, en China. Posteriormente, el patógeno que la causaba fue identificado: se trata de un virus de la familia de los coronavirus, bautizado como SARS-CoV-2, transmisible entre individuos a través de aerosoles y pequeñas gotas de saliva y otros fluídos (Lu et al. 2020). La enfermedad asociada a este patógeno, denominada COVID-19, se ha extendido por todo el planeta en un periodo relativamente corto de tiempo, siendo reconocida como pandemia por la Organización Mundial de la Salud en marzo de 2020. En Europa la expansión comenzó por Italia, continuando con un rápido incremento en el número de casos en España y posteriormente en el resto de países europeos; esta rápida expansión supuso un colapso de los sistemas sanitarios de todo el mundo, y los distintos hospitales y laboratorios se han volcado en el estudio de la enfermedad para intentar predecir la mortalidad y la gravedad de la sintomatología (Remuzzi y Remuzzi 2020). Sus manifestaciones clínicas varían desde una infección asintomática o con sintomatología leve (como fiebre, tos, fatiga...), hasta los casos más graves en los que se produce neumonía, que puede progresar a Síndrome de Dificultad Respiratoria Aguda, insuficiencia multiorgánica y, en última instancia, la muerte (Huang et al. 2020).

En este contexto, resulta de vital importancia identificar las causas asociadas a un peor pronóstico. Gracias a los múltiples estudios que se han ido publicando en el último año, se han identificado algunas de las características clínicas que pueden llegar a producir episodios agudos de la enfermedad e incluso la muerte (Gude-Sampedro et al. 2020, Elezkurtaj et al. 2021, Bhaskaran et al. 2021), entre las que destacan la edad, el sexo, el tabaquismo o la diabetes, entre otras. Sin embargo, todavía queda mucho por investigar; un posible campo de estudio es el análisis conjunto de las variables clínicas con datos extraídos de las pruebas más comúnmente realizadas a los enfermos, como puede ser el caso de las radiografías de tórax.

1.2. Radiografías

Aunque en los últimos años se ha avanzado mucho en el campo de las imágenes médicas, la radiografía de tórax continúa siendo una de las técnicas más utilizadas para el diagnóstico de muchas enfermedades pulmonares, en especial en pacientes con sospecha de neumonía. Esto es así porque es una técnica de bajo precio, sencilla de realizar, y que supone aplicar una menor cantidad de radiación al paciente que con el uso de otras técnicas más precisas, como puede ser la tomografía computerizada (TC, antiguamente conocida como Tomografía Axial Computarizada o TAC) (Coche 2011). Las radiografías, sin embargo, son de difícil interpretación, ya que son una proyección bidimensional de

un volumen tridimensional, por lo que los diversos tejidos se superponen entre sí, lo que complica la identificación de regiones dañadas. Existe además el denominado "ruido anatómico", que no permite detectar anomalías que se encuentran en el tejido pulmonar situado por debajo de otros tejidos, como por ejemplo las costillas o el corazón (Lee et al. 2017).

El principio básico de una radiografía es que los valores de gris que toma se corresponden con las variaciones en el número de rayos-X que han pasado a través del cuerpo del paciente; los tejidos blandos (órganos, grasa, etc.) permiten el paso de una mayor cantidad de rayos-X, mientras que los tejidos duros como los huesos bloquean su paso (Delrue et al. 2011). Por lo tanto, el análisis de las radiografías se centra en investigar los cambios en estos tonos de gris.

Hace años, las radiografías se realizaban mediante pantalla de película tradicional, pero estas han sido reemplazadas por las radiografías digitales, que son mucho más fáciles de realizar y de manejar. La calidad de estas radiografías digitales, al igual que la de cualquier imagen digital, depende de su resolución espacial (es decir, la cantidad de unidades mínimas de color o píxeles en la que se divide la imagen), y el contraste (es decir, el grado de diferencia existente entre los distintos colores que conforman la imagen) (Shephard 2003).

En el caso de la enfermedad por COVID-19, a pesar de que en algunos hospitales se optó por realizar a los pacientes TC para estudiar el daño pulmonar, en la mayoría se utilizaron radiografías de tórax, dado que son más fáciles y rápidas de llevar a cabo, aportan una menor carga radiológica al paciente, y muchos de los aparatos con los que se realizan se pueden mover de una sala a otra, lo que en un contexto de una enfermedad con tan alta capacidad de contagio es mucho más seguro (Wong et al. 2020). Dado el alto número de pacientes que cursan esta enfermedad, y el gran desconocimiento que todavía tenemos sobre los factores que afectan a su pronóstico, es imprescindible desarrollar técnicas que permitan el análisis de estas radiografías. Existen múltiples estudios centrados en extraer características de las radiografías de tórax para el análisis de la evolución de la COVID-19, aunque mayoritariamente aplicando técnicas de *deep learning* como redes neuronales (ver, por ejemplo: Tabik et al. 2020 o Wang et al. 2020), que requieren generalmente de una gran cantidad de datos y resultan computacionalmente muy costosas.

1.3. Textura

Dado que las radiografías de tórax no son más que imágenes digitales en blanco y negro, se pueden analizar aplicando técnicas específicas del análisis de imagen. Una imagen digital en blanco y negro se caracteriza fundamentalmente por las variaciones globales de los valores de gris (el "tono"), y por las variaciones entre valores de gris cercanos, es decir, la distribución espacial de los valores de gris, lo que comúnmente llamamos "textura" (Haralick et al. 1973). La textura es, por tanto, una característica fundamental: proporciona la apariencia de los objetos presentes en las imágenes, dado que nos permite identificar los cambios entre los distintos elementos. Según la percepción humana, la textura se puede definir en función de tres características fundamentales: la rugosidad, el contraste y la direccionalidad, siendo la más importante la rugosidad, que mide la granularidad de la imagen (Karmakar et al. 2017).

A pesar de que visualmente la textura es fácil de percibir, no existe una definición matemática estricta, y varios autores han propuesto distintas metodologías para su cálculo: por ejemplo, Haralick et al. (1973) y Levine (1985) proponen la extracción de funciones primitivas (momento angular, correlación, entropía, etc.) a partir de una matriz de diferencias de gris. Otros autores, como Weszka et al. (1976) proponen un enfoque estadístico, a partir del estudio de la relación entre pares de píxeles (conocido como "estadísticas de segundo orden"), siendo el principal método estadístico utilizado el que se basa en la definición de las distribuciones de probabilidad conjunta de pares de píxeles. Por último, existen autores, como Amadasun y King (1989), que proponen una metodología alternativa

para la obtención de la textura, basada en el cálculo de las diferencias en los tonos de gris de píxeles adyacentes, cuyos resultados se acercan más a la percepción humana.

La textura se ha utilizado ampliamente en el análisis de imágenes médicas, por ejemplo, en la clasificación de masas en mamografías (Khuzi et al. 2008), en el estudio de la tuberculosis (Hakim y Basari 2019), o para la detección de neumonía en radiografías de tórax (Deepa et al. 2018), y, por supuesto, múltiples autores han propuesto su estudio para el análisis de las radiografías de enfermos de COVID-19 (ver, por ejemplo: Hussain et al. 2020, Thepade et al. 2020, Cavallo et al. 2021). Sin embargo, la mayoría de ellos han utilizado propiedades básicas de la textura, y, hasta donde sabemos, ninguno ha aplicado una metodología que permita identificar la textura de una radiología de forma análoga a la percepción visual, como puede ser el método propuesto por Amadasun y King (1989). En este trabajo, por lo tanto, hemos decidido implementar el cálculo de la rugosidad según proponen estos autores, y analizar si esta variable puede resultar de interés para predecir la probabilidad que tienen los pacientes de COVID-19 de fallecer.

Capítulo 2

Extracción y preprocesado de datos

2.1. Obtención de los datos

Se dispone de datos demográficos y clínicos, además de las radiografías de tórax, de 228 individuos con diagnóstico de SARS-CoV-2 mediante RT-PCR (*Real-Time Polymerase Chain Reaction*) positiva. Se han incluido los casos diagnosticados entre el 01/03/2020 y el 25/06/2020 pertenecientes al Área Sanitaria de Santiago de Compostela. Las RT-PCR fueron realizadas bien en individuos con síntomas compatibles con la enfermedad de COVID-19, tales como fiebre, anosmia, dificultad para respirar, etc., o bien en aquellos que tuvieran contacto con algún caso positivo. El estudio se realizó de conformidad con las directrices de la Declaración de Helsinki y los principios de buena práctica clínica.

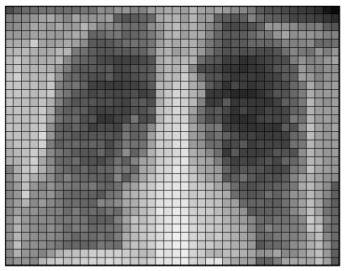
Los datos recogidos de la historia clínica de los pacientes fueron los siguientes: edad, sexo, si es fumador/a activo/a (entendiendo por fumadores activos aquellos que consumen un cigarrillo o más al día) o ex-fumador/a (aquellos que dejaron de fumar hace 1 año o más), si presenta otras comorbilidades como diabetes o enfermedad pulmonar obstructiva crónica (EPOC), si el paciente falleció o no (*exitus*), la fecha de la muerte en caso de que ocurra el evento, o, en su defecto, la fecha de alta, y la fecha de inicio de la enfermedad. Este tiempo de inicio es el mínimo entre la fecha de inicio de los síntomas y la fecha en la que se realizó la RT-PCR. Se recogieron además las radiografías frontales de tórax realizadas a los pacientes entre el inicio de la enfermedad y el alta o *exitus*, descartándose aquellas de mala calidad, resultando en un total de 607 radiografías.

Este trabajo forma parte del proyecto "Pronóstico de los pacientes infectados por el nuevo coronavirus SARS-CoV-2", con referencia: COV20_00404, financiando por el Instituto de Salud Carlos III de Madrid.

2.2. Imagen Digital: formato DICOM

Una imagen monocromática se puede definir como una función bidimensional que asigna a cada par de coordenadas espaciales un valor de intensidad o amplitud de la imagen en dicho punto; en las imágenes en blanco y negro (como es el caso de las radiografías), esta intensidad se denota habitualmente como valor de gris. Tanto las coordenadas x e y como la amplitud son variables continuas. Para convertir una imagen en imagen digital es necesario transformar dichas variables para que tomen valores discretos: el proceso de discretización de las coordenadas espaciales se denomina muestreo, y el de la amplificación se denomina cuantificación (González et al. 2003). Así, una imagen digital se puede representar como una función: $f : (x, y) \in \mathbb{N}^* \times \mathbb{N}^* \rightarrow f(x, y) \in \mathbb{N}$. [1]

[1] En esta notación estamos considerando que: $0 \in \mathbb{N}$, $0 \notin \mathbb{N}^*$



$$\longrightarrow \begin{pmatrix} f(1,1) & f(1,2) & \dots & f(1,n) \\ f(2,1) & f(2,2) & \dots & f(2,n) \\ \vdots & & \vdots & \\ f(m,1) & f(m,2) & \dots & f(m,n) \end{pmatrix}$$

Figura 2.1: Representación matricial de una imagen digital. Radiografía extraída de la base de datos, y modificada para reducir su número de píxeles.

Cada elemento de la matriz de la Figura 2.1 se denomina píxel. Un píxel es, por tanto, la unidad mínima de color homogéneo que conforma una imagen digital, y está determinado por su posición (x, y) y por su tono o valor de gris $f(x, y)$.

Además de la matriz de valores de gris, una imagen digital suele ir acompañada de una cabecera, que contiene información sobre los atributos propios la imagen (tamaño del píxel, dimensiones, etc.). En el caso de las radiografías (y, en general, de las imágenes médicas), el formato estándar es el formato DICOM (*Digital Imaging and Communication On Medicine*) (Mildenberger et al. 2002). Estos archivos contienen, por una parte, la matriz de valores de gris, con un tamaño aproximado de 2000×3000 píxeles (aunque las dimensiones exactas dependen del aparato con el que es realizada la radiografía, y de la calidad de la misma), y una cabecera en la que se encuentran, entre otros, los siguientes atributos[2]:

1. Fecha y hora en la que fue realizada la radiografía.
2. Tipo de radiografía (`RXType`): puede ser PA (posteroanterior, es decir, la fuente de rayos X se posiciona de forma que los rayos entran por la espalda y salen por la parte frontal), AP (anteroposterior: los rayos entran por el pecho y salen por la espalda), o L (lateral). En nuestro caso se han descartado las radiografías laterales, dado que por su propia naturaleza sólo contienen el perfil del tórax.
3. Modalidad de radiografía (`Modality`): el tipo de radiografía. En nuestro caso hay dos modalidades: CR (*Computed Radiography*) y DX (*Digital Radiography*). Las CR son una versión más antigua de radiografías digitales, en las que se utilizan láminas foto-estimulables para capturar la imagen, que posteriormente es procesada para transformarla en imagen digital. En las DX, los rayos-X se capturan directamente en unos circuitos especiales que generan la imagen digital en tiempo real. En este caso se han mantenido las radiografías obtenidas con ambas modalidades.
4. `PhotometricInterpretation`: codifica la interpretación de los valores del píxel. En las radiografías disponibles toma los siguientes valores: *Monochrome1* (si el valor mínimo de los píxeles de la imagen se corresponde con el blanco), y *Monochrome2* (si el valor mínimo de los píxeles de la imagen se corresponde con el negro). Ver 2.3.1 para más detalles.
5. Marca (`Manufacturer`) y modelo (`Model`) del aparato utilizado para realizar la radiografía. Esta información va a resultar clave, dado que al no existir un estándar para la codificación de la intensidad de los rayos-X, existen diferencias entre las distintas casas comerciales que pueden dar lugar a problemas en el análisis de la radiografía, como se verá en la siguiente sección.

Las imágenes en formato DICOM, al contener tanta información, no son fácilmente manejables: requieren de programas específicos para su visualización, y la extracción de la información que contienen tampoco es trivial. Aunque existen paquetes en `R` que permiten leer los archivos DICOM, como

[2]Se puede consultar la lista completa de atributos en: https://dicom.innolitics.com/ciods

por ejemplo el paquete `oro.dicom` (Whitcher et al. 2011), es en `Python` donde más librerías se han desarrollado para el análisis y procesado de este tipo de archivos. En este trabajo, por lo tanto, se ha optado por extraer las características propias de las radiografías (la matriz de grises y la información relevante de la cabecera) generando scripts en `Python`, que se pueden consultar en el Anexo A.

2.3. Dificultades

Debido principalmente a que las radiografías disponibles se han tomado con distintos aparatos, existen diferencias en las matrices de grises que hacen que no sean directamente comparables entre sí, y por lo tanto es necesario de un preprocesado de los datos para poder realizar los análisis.

2.3.1. Rango de grises

El rango de valores que puede tomar cada píxel de una imagen digital depende del número de *bits* utilizados para representarlo. Por ejemplo, un píxel de 8 *bits* ($= 1byte$) puede tomar $2^8 = 256$ valores diferentes de gris (recordemos que los *bits* tienen formato binario, por lo que los valores que pueden tomar son 0 o 1) (Sprawls 1993). Generalmente, las imágenes médicas contienen entre 12 y 16 *bits* por píxel, lo que se traduce en entre 4096 y 65536 posibles valores de gris (Kimpe y Tuytschaever 2007). Cuanto mayor es el número de *bits* por píxel, mayor es la resolución de la imagen, pero también se incrementa su tamaño. El tamaño habitual de una radiografía es de aproximadamente 2000×3000 píxeles, con lo que una imagen de 12 *bits* por píxel se traduce en un tamaño de 9MB, mientras que de 16 *bits* por píxel sería equivalente a 12MB. Dado que el tamaño de las imágenes afecta no sólo a la capacidad de almacenamiento si no también a su procesado, es necesario buscar un equilibrio óptimo entre resolución y tamaño; en la práctica, cada fabricante determina el número de *bits* por píxel a su elección, lo que provoca que el rango de valores de gris difiera mucho entre los distintos aparatos. Esta diferencia en el rango de valores que toman los píxeles puede afectar al análisis de las radiografías, dado que cuanto menor sea el rango más difícil resulta detectar diferencias entre los distintos elementos de la imagen, como se puede ver en la Figura 2.2.

(a) 1 *bit* por píxel **(b)** 2 *bits* por píxel **(c)** 3 *bits* por píxel

Figura 2.2: Distintas resoluciones de imagen según el número de *bits* por píxel para un individuo aleatorio de la base de datos.

Además de afectar a la resolución de las radiografías, estas diferencias en rango también producen que los valores de gris no sean directamente comparables entre radiografías tomadas con distintos aparatos. Esto se debe a que los valores extremos en las radiografías se corresponden con el blanco y el negro absolutos, y todos los valores intermedios serían tonalidades de gris más o menos oscuras según se encuentren más cerca de uno u otro extremo (Pawley 2006). Entonces, en dos radiografías con rangos muy diferentes, un píxel con el mismo valor de gris puede estar representando un color cercano

al negro en una y cercano al blanco en otra. En nuestro caso, disponemos de radiografías realizadas con aparatos de varias marcas diferentes; los rangos que toman cada uno de estos aparatos se pueden ver en la Figura 2.3.

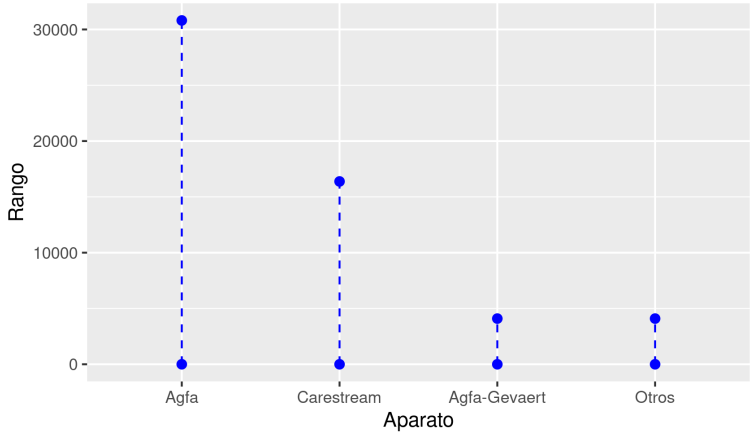


Figura 2.3: Diferentes rangos según el aparato con el que fue realizada la radiografía; distinguimos tres grandes grupos: "Agfa" ($n = 525$, el 86.49 %), "Carestream" ($n = 16$, el 2.64 %), y "Agfa-Gevaert" ($n = 61$, el 10.05 %) junto con "Otros" (5, el 0.82 %), que engloba diversas marcas.

Tomando el caso, por ejemplo, de una radiografía realizada con un aparato de marca "Agfa" y otra realizada con "Agfa-Gevaert", y suponiendo que el 0 se corresponde con el negro en ambos casos, un píxel con un valor de gris de 4000 se traduciría en un blanco en "Agfa-Gevaert" y en un gris cercano al negro en "Agfa".

Por lo tanto, para poder comparar radiografías realizadas con distintos aparatos es necesario aplicar previamente una normalización de los niveles de gris. Para ello utilizamos la normalización por escala de intensidad, o *intensity scaling* (Sun et al. 2015): consideremos una imagen digital con matriz de píxeles $\mathbf{M}$ de tamaño $m \times n$. Para cada píxel con coordenadas $(x, y) \in \{\mathbb{N}^* \times \mathbb{N}^* \ / \ x \leq m,\ y \leq n\}$, y valor de gris $f(x, y) \in \mathbb{N}$, se obtiene:

$$f'(x, y) = \frac{f(x, y) - LI}{HI - LI}$$

Donde LI denota la intensidad mínima y HI la intensidad máxima. Estas intensidades extremas pueden ser: el mínimo y máximo globales, los deciles 1 % y 99 %, $\mu \pm 3\delta$, etc. (Kociolek et al. 2018). En nuestro caso hemos optado por usar el mínimo y el máximo globales:

$$LI = \text{mín}(f(x, y),\ \forall (x, y) \in \{\mathbb{N}^* \times \mathbb{N}^* \ / \ x \leq m,\ y \leq n\})$$

$$HI = \text{máx}(f(x, y),\ \forall (x, y) \in \{\mathbb{N}^* \times \mathbb{N}^* \ / \ x \leq m,\ y \leq n\})$$

por sencillez y porque nos permite tener los valores acotados en el intervalo $[0, 1]$.

Esta normalización se ha implementado en el script `ROILung.py`, que se puede consultar en el Anexo A.

Esta transformación permite que los valores sean comparables entre sí, pero no resuelve el problema de que cada aparato puede tomar un número muy diferente de valores de grises. Por ejemplo, el cardinal del conjunto de valores de gris de las radiografías realizadas con aparatos "Agfa" se encuentra en torno a 30800, mientras que en el caso de la marca "Carestream" es inferior a 16400, y en el resto de

aparatos es menor de 4100. Esto va a afectar a lo semejantes que se perciban los distintos tonos de gris: visualmente no hay mucha diferencia para el ojo humano, pero matemáticamente sí se van a apreciar estos cambios.

2.3.2. Codificación de los valores extremos

La intensidad o valor de gris de cada píxel mide la cantidad de fotones de rayos-X que han sido capaces de atravesar los tejidos correspondientes a ese punto específico (Sprawls 1993). Los tejidos con menor densidad (como los órganos) absorben una menor cantidad de fotones, lo que se traduce en colores más oscuros que los tejidos más densos (como los huesos), que se corresponderían con tonos blancos. Dependiendo del aparato y del modelo utilizado para realizar la radiografía, los valores mínimos de intensidad pueden hacer referencia bien a tonos blancos o bien a tonos negros, lo que provoca que sea necesario realizar una transformación de las matrices de valores de gris para poder realizar la comparación entre radiografías tomadas con distintos aparatos. Esta información se encuentra recogida en la cabecera DICOM, en el parámetro `Photometric Interpretation` (PI):

- Si PI es *Monochrome1*, entonces el valor mínimo se corresponde con el blanco.
- Si PI es *Monochrome2*, entonces el mínimo está codificando el color negro.

Para solventar este problema, aplicamos una transformación lineal a las radiografías cuyo PI es *Monochrome1*, para que en todos los casos el negro se corresponda con el valor mínimo y el blanco con el valor máximo. Es decir: dada una imagen digital cuyo PI es *Monochrome1*, con matriz de píxeles $\mathbf{M}$ de tamaño $m \times n$. Para cada valor de gris $f(x, y) \in \mathbb{N}$, se define:

$$f'(x, y) = (HI + LI) - f(x, y)$$

siendo:

$$LI = \text{mín}(f(x, y),\ \forall (x, y) \in \{\mathbb{N}^* \times \mathbb{N}^*\ /\ x \leq m,\ y \leq n\})$$

$$HI = \text{máx}(f(x, y),\ \forall (x, y) \in \{\mathbb{N}^* \times \mathbb{N}^*\ /\ x \leq m,\ y \leq n\})$$

Esta transformación se encuentra implementada en el script `ModifyDICOM.py`, que se puede consultar en el Anexo A.

2.3.3. Valores atípicos

En algunas de las radiografías disponibles se han detectado valores atípicos en las matrices de grises: en algunos casos se corresponden con objetos como joyas, elementos metálicos de la ropa, etc.; en otros pueden estar señalando elementos ajenos a la radiografía (por ejemplo, notas introducidas por los técnicos), e incluso en algunos casos pueden estar identificando fallos en algún sensor del aparato.

Estos valores atípicos se suelen encontrar fuera de la región de los pulmones, por lo que generalmente no suponen un problema a la hora del estudio del daño pulmonar, pero sí afectan a la transformación lineal (ver sección 2.3.2), por lo que es necesario eliminarlos antes de realizar los análisis. Para ello, y con el objetivo de reducir al mínimo la pérdida de sensibilidad, identificamos los valores que quedan por debajo del cuantil 0.0001 % y por encima del 99.9999 %, y les asignamos el valor mínimo y máximo del total restante, respectivamente.

Este proceso se encuentra implementado en el script `ModifyDICOM.py`, que se puede consultar en el Anexo A.

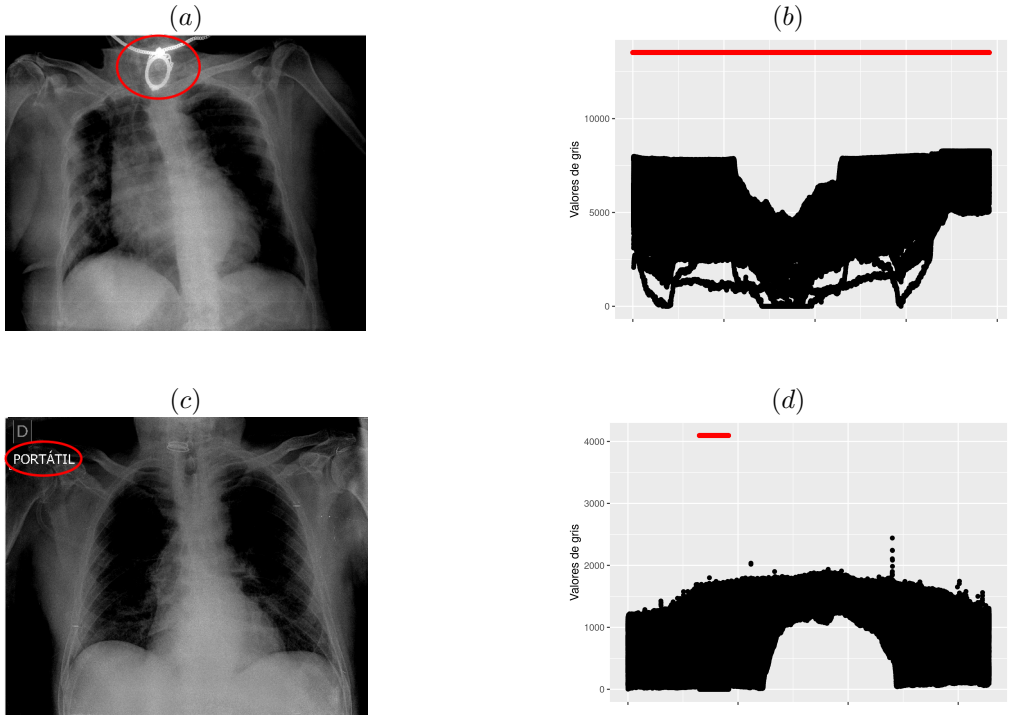


Figura 2.4: Ejemplos de radiografías con valores atípicos: en (a) se muestra una radiografía en la que se puede observar un objeto extraño (una joya), y en (c) una radiografía con elementos ajenos a la radiografía (letras escritas sobre la radiografía). En (b) y (d) se recogen los valores de gris de las radiografías (a) y (c), respectivamente.

2.4. Extracción de regiones de interés

En este estudio estamos interesados en analizar posibles daños en el pulmón de los pacientes. Sin embargo, las radiografías de tórax contienen una región mucho mayor de la que se quiere analizar; además, el tamaño de los pulmones y la posición de los pacientes difiere mucho entre radiografías. Es necesario, por tanto, determinar la región de interés (*Region Of Interest*, ROI) de la que queremos extraer los datos. En la literatura es común utilizar dos técnicas diferentes para extraer los ROI: la extracción de regiones rectangulares en las que se encuentren contenidos los pulmones (véase, por ejemplo, Eze et al. 2020), y, más habitualmente, el recorte de los pulmones mediante el uso de software específico (véase: Cavallo et al. 2021, o Zhang et al. 2020). Utilizamos por tanto estas dos técnicas para extraer los ROI:

1. Por un lado, siguiendo las indicaciones de profesionales expertos en el análisis de radiografías de tórax, se realizó de forma manual el recorte de regiones rectangulares conteniendo la zona pulmonar. Para ello se utilizó un software desarrollado por la USC en colaboración con AIMEN Centro Tecnolóxico (https://www.aimen.es/) para el manejo y visualización de archivos DICOM (Franco et al. 2013), que nos permite seleccionar manualmente el rectángulo donde se encuadran los pulmones, y nos devuelve como *output* el centro y las dimensiones de dicho rectángulo. Una vez tenemos la caja torácica recortada, se dividen los rectángulos en tres partes iguales, y se descarta la región central, quedándonos con dos matrices de valores de gris: la correspondiente al pulmón izquierdo y la correspondiente al pulmón derecho (ver Figura 2.5).

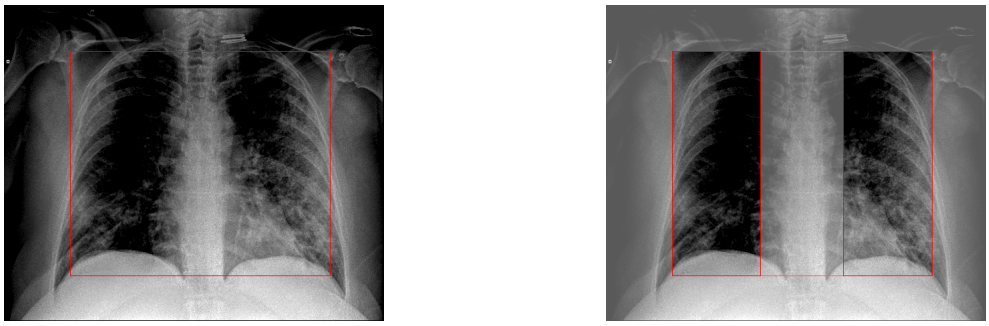


Figura 2.5: Ejemplo de recorte rectangular para una radiografía de la base de datos.

En ocasiones la posición del paciente hace que el recorte en forma rectangular no nos permita seleccionar la región pulmonar. Para solventar este problema, usamos otra funcionalidad del software que nos permite obtener las regiones determinadas por puntos seleccionados manualmente. El *output* en este caso son los vértices del polígono (A_0, A_1, A_2, A_3). Este polígono se debe primero ajustar a un rectángulo, y posteriormente girar para que la posición de los pulmones quede recta. Este proceso se ha implementado en `Python`: en primer lugar, se calculan los nuevos vértices $(\hat{A}_0, \hat{A}_1, \hat{A}_2, \hat{A}_3)$ eligiendo la distancia máxima entre los vértices originales, y de forma que los segmentos que los unen sean paralelos. Una vez se tiene el rectángulo, se calcula el sentido y ángulo de giro α según la pendiente de la recta que une $\hat{A}_0$ y $\hat{A}_1$; por último, utilizando este ángulo, se gira toda la imagen, y se obtienen el centro y las dimensiones del rectángulo final (ver Figura 2.6).

Es decir: sean $A_0 = (x_0, y_0)$, $A_1 = (x_1, y_1)$, $A_2 = (x_2, y_2)$, y $A_3 = (x_3, y_3)$ las coordenadas de los vértices del polígono en el sentido de las agujas del reloj, y sea: $d(A_2, A_3) > d(A_1, A_0)$, y $d(A_0, A_3) > d(A_1, A_2)$ (donde d representa la distancia euclídea). Sea m la pendiente de la recta que une A_0 y A_1:

$$m = \frac{y_1 - y_0}{x_1 - x_0}$$

Entonces $\hat{A}_0$ viene dado por la intersección entre la recta de pendiente m que pasa por A_0 y la recta de pendiente $-1/m$ que pasa por A_3:

$$\hat{x}_0 = \frac{y_0 - y_3 - (1/m) \cdot x_3 - m \cdot x_0}{(-1/m) - m};\ \hat{y}_0 = y_3 - (1/m) \cdot (\hat{x}_0 - x_3)$$

Análogamente, $\hat{A}_1$ se obtiene como la intersección entre la recta de pendiente m que pasa por A_1 y la recta de pendiente $-1/m$ que pasa por A_2, y $\hat{A}_2$ es el punto de intersección entre la recta de pendiente m que pasa por A_3 y la recta de pendiente $-1/m$ que pasa por A_2, mientras que A_3, por su parte, se mantiene fijo ($\hat{A}_3 = A_3$). Obtenemos el ángulo de giro, α, a partir de la pendiente de la recta que une A_0 y A_1: $\alpha = \arctan(m)$, y calculamos el centro y las dimensiones del rectángulo final: sea $(\hat{x}, \hat{y})$ el punto medio del rectángulo de vértices $(\hat{A}_0, \hat{A}_1, \hat{A}_2, \hat{A}_3)$. El centro del rectángulo tras el giro, (X, Y), viene dado por:

$$X = \hat{x} \cdot \cos(\alpha) + \hat{y} \cdot \sin(\alpha);\ Y = \hat{y} \cdot \cos(\alpha) - \hat{x} \cdot \sin(\alpha)$$

Finalmente, las dimensiones se obtienen como la distancia euclídea entre los nuevos vértices.

En el caso de que los nuevos vértices estimados se sitúen fuera de la imagen original, se guarda el valor del píxel como valor perdido (*nan*).

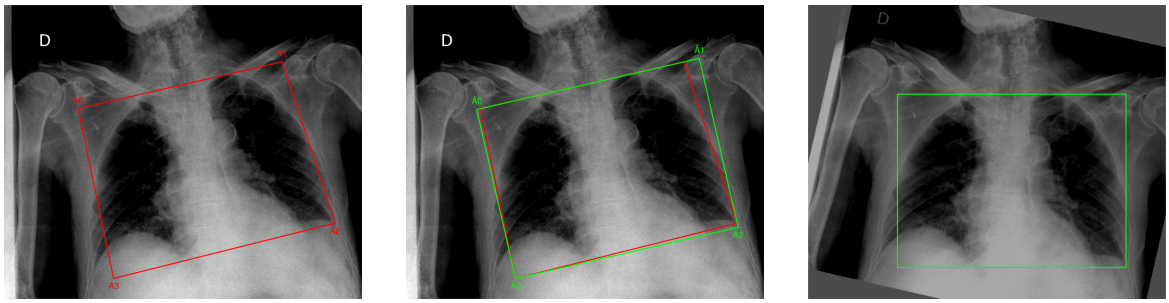


Figura 2.6: Ejemplo de giro para una radiografía de la base de datos.

Este proceso se realiza en la función `ROIRectangular.py` (ver Anexo A): se extraen las zonas de interés de la región marcada manualmente, aplicando el giro en las que sea necesario, y se escriben los niveles de gris de cada ROI en formato *txt*.

2. Por otro lado, se procede a recortar directamente los pulmones con el programa ITK-SNAP [3] (Yushkevich et al. 2006), el cual nos permite identificar manualmente las regiones de interés y guardar dichas zonas en un archivo MHA del mismo tamaño que la DICOM, con valores: 0 si no pertenece a la región de interés, 1 si pertenece al pulmón derecho, y 2 si pertenece al pulmón izquierdo (ver Figura 2.7). Posteriormente, estos archivos se superponen a la imagen original para extraer únicamente los valores de gris que corresponden a las regiones recortadas, y se escriben los valores para cada ROI en archivos *txt*. Este proceso se realiza en el script `ROILung.py` (ver Anexo A).

Figura 2.7: Representación gráfica del proceso de recorte con el programa ITK-SNAP.

El recorte con el programa ITK-SNAP parece la opción óptima, dado que es mucho más preciso y no introduce tanto ruido en los ROI. Sin embargo, requiere de mayor conocimiento anatómico (cómo distinguir el pulmón de otras regiones, cómo identificar la zona del corazón o del diafragma, ...), y es un proceso más lento y costoso, ya que no sólo el recorte en sí es más complicado, si no que además requiere de la supervisión por parte de un radiólogo o un neumólogo. El recorte rectangular, por su parte, es mucho más sencillo (no se necesitan grandes conocimientos anatómicos para realizarlo), y más rápido, por lo que se pueden procesar muchas radiografías en poco tiempo. Para evaluar si el incremento de la dificultad en el preprocesado con el programa de ITK-SNAP se corresponde con una clara mejoría de cara al análisis, se eligieron 35 radiografías de pacientes que no fallecieron y realizadas con aparatos de la marca "Agfa", y se extrajeron las regiones de interés de las dos formas descritas.

[3] Este programa se puede descargar de forma gratuita desde: http://www.itksnap.org/

Se realizó el preprocesado de los valores de gris, transformando las radiografías según la `Photometric Interpretation` (ver Sección 2.3.2), eliminando los valores atípicos (ver Sección 2.3.3), y aplicando la normalización descrita en la Sección 2.3.1, y se extrajeron las curvas de densidad correspondientes a cada pulmón, utilizando un estimador tipo núcleo Gaussiano, y la regla del pulgar (*rule-of-thumb*) para la estimación del parámetro ventana (Figura 2.8).

Vemos que existe una clara diferencia en las curvas de densidad entre los pulmones derechos e izquierdos de los individuos en los recortes rectangulares (Figura 2.8, (a) y (b)), probablemente debido a la presencia del corazón, mientras que las diferencias son mucho menores en el caso del recorte de la región pulmonar (Figura 2.8, (c) y (d)), por lo que optamos por realizar el recorte de las regiones de interés con el programa ITK-SNAP.

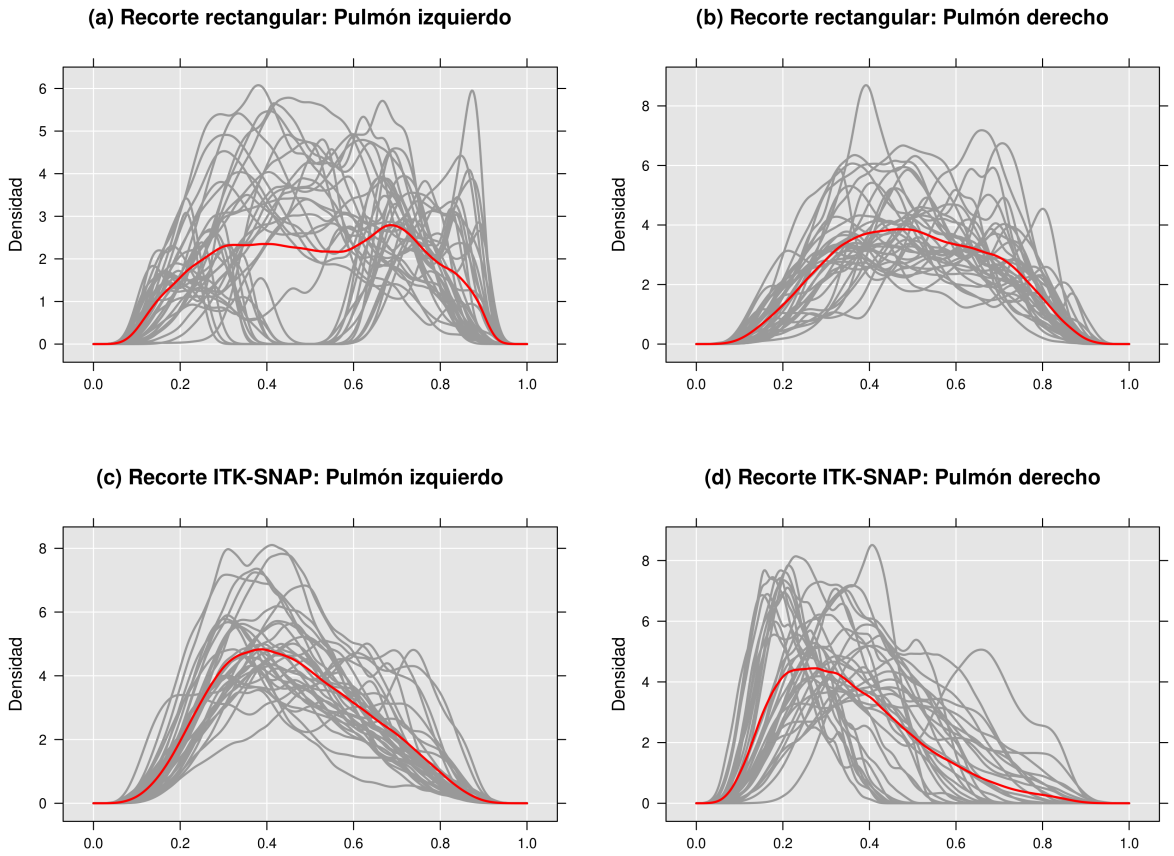
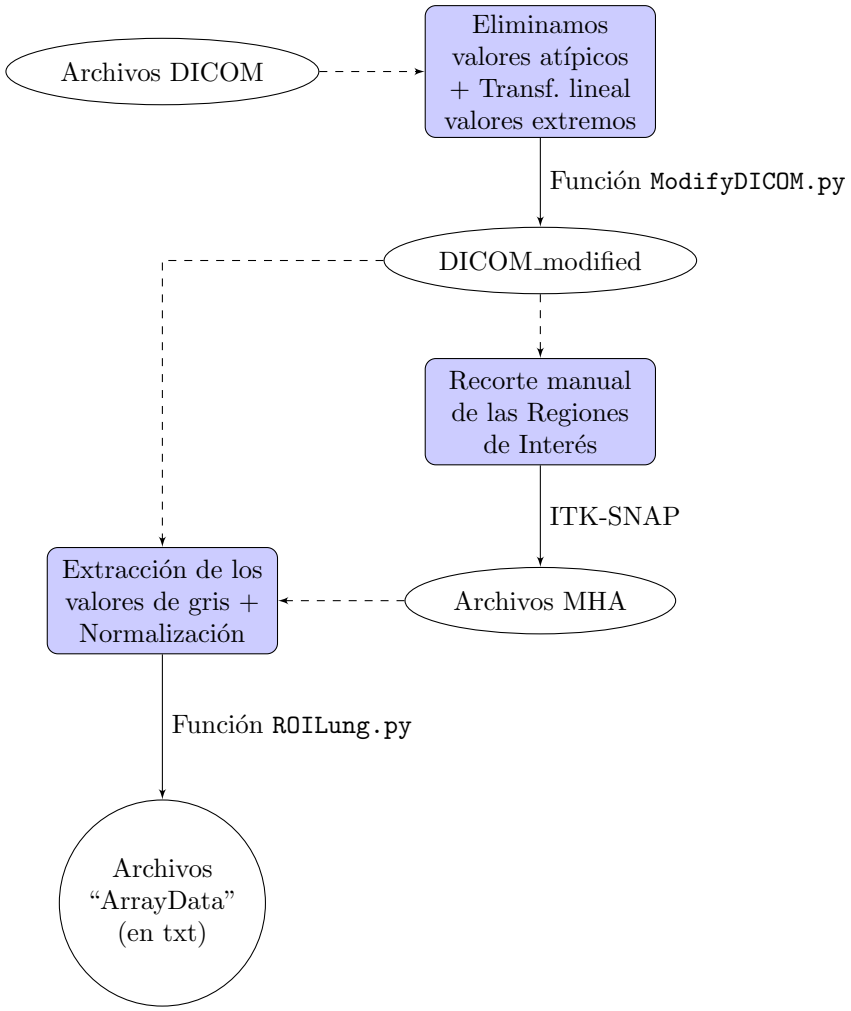


Figura 2.8: Curvas de densidad para pulmón izquierdo y derecho según los dos tipos de recortes; se presenta en rojo la curva media de las densidades. En (a) y (b) están recogidas las curvas de densidad para los recortes rectangulares para los pulmones izquierdo y derecho, respectivamente; en (c) y (d) están recogidas las curvas de densidad para los recortes con el programa ITK-SNAP para los pulmones izquierdo y derecho, respectivamente.

2.5. Resumen del procesado de las radiografías

En el siguiente diagrama se encuentra un resumen de los pasos seguidos para la obtención de los archivos finales ("ArrayData") a partir de las radiografías en su formato original, DICOM.



Capítulo 3

Textura: *coarseness*

En una imagen digital, la textura mide la homogeneidad entre píxeles adyacentes; es, junto al propio valor del píxel, una de las características principales de la imagen. Dicha textura se puede describir en función de distintas características (el contraste, la orientación, la regularidad, etc.), siendo una de las más importantes la rugosidad (*coarseness* en inglés), es decir, si los elementos de una imagen son gruesos o finos. Amadasun y King (1989) proponen 5 medidas diferentes de la textura, entre las que se encuentra la *coarseness*, a la que ellos mismos consideran la propiedad fundamental de la textura. Una imagen con textura gruesa sería aquella en la que los elementos que la componen son grandes, y por lo tanto hay zonas de relativamente gran tamaño con intensidades similares. Por ello, habría pequeñas diferencias entre los tonos de gris de los píxeles de la imagen y el promedio de tonos de gris de sus píxeles adyacentes: la suma de estas diferencias, ponderada por la frecuencia en la que aparece cada valor de gris, daría una indicación de lo rápidamente que se producen cambios de intensidad dentro de la imagen, y por lo tanto mostraría su rugosidad.

3.1. Cálculo

Para obtener la *coarseness* de una imagen, vamos a seguir la fórmula propuesta por Amadasun y King (1989): se genera en primer lugar la matriz llamada Neighborhood Gray-Tone Difference Matrix (**NGTDM**) de la siguiente forma:

Sea $\mathbf{M}$ la matriz de valores de gris de una imagen digital, de tamaño $m \times n$, y sea $i = f(k,l) \in \mathbb{N}$ el nivel de gris del píxel situado en la posición (k,l) de dicha matriz. Se define:

$$\hat{A}(k,l) = \frac{1}{(2d+1)^2 - 1} \left[\sum_{k_d=-d}^{d} \sum_{l_d=-d}^{d} f(k+k_d, l+l_d) \right] \tag{3.1}$$

para $(k_d, l_d) \neq (0,0)$, donde $d \in \mathbb{N}^*$ es un parámetro de distancia que determina el tamaño del vecindario: $(2d+1) \times (2d+1)$. Entonces, el valor de la matriz **NGTDM** para el tono de gris i viene dado por:

$$s(i) = \begin{cases} \sum_i |i - A_i| \,, \forall i \in N_i \\ \\ 0 \ \textit{otro caso} \end{cases} \tag{3.2}$$

donde $N_i = \{(k,l) \ / \ f(k,l) = i, \ k \in (d, m-d), \ l \in (d, n-d)\}$, y $A_i = \{\hat{A}(k,l), \forall (k,l) \ / \ f(k,l) = i\}$.

Finalmente, la *coarseness* se define como:

$$f_{coar} = \left[\epsilon + \sum_{i=0}^{max(i)} p_i s(i) \right]^{-1} \tag{3.3}$$

donde p_i es la frecuencia relativa del valor i en el vecindario de la matriz $\mathbf{M}$, y $\epsilon \in \mathbb{R}^+$ un valor positivo lo suficientemente pequeño para evitar que f tienda a infinito.

Ejemplo 1 *Consideremos la siguiente matriz de orden* 5×5*:*

$$\begin{pmatrix} 1 & 1 & 4 & 3 & 1 \\ 3 & 4 & 0 & 1 & 1 \\ 5 & 4 & 0 & 3 & 2 \\ 2 & 1 & 1 & 3 & 4 \\ 0 & 2 & 2 & 5 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 1 & 4 & 3 & 1 \\ 3 & \boxed{4} & \boxed{0} & \boxed{1} & 1 \\ 5 & \boxed{4} & \boxed{0} & \boxed{3} & 2 \\ 2 & \boxed{1} & \boxed{1} & \boxed{3} & 4 \\ 0 & 2 & 2 & 5 & 1 \end{pmatrix}$$

Si tomamos $d = 1$*, resulta un vecindario de orden* 3×3 *(resutado de excluir las filas* 1 *y* 5*, y las columnas* 1 *y* 5 *de la matriz original), en el que se tienen los niveles de gris:* $N = \{0, 1, 3, 4\}$*. Para obtener el valor de* f_{coar}*, seguimos los siguientes pasos:*

1. *Calculamos los valores de* $\hat{A}$ *para cada píxel del vecindario:*

   $$\hat{A}(2,2) = \frac{1+1+4+3+0+5+4+0}{8} = 2.25$$

   De manera análoga, se calcula $\hat{A}$ *para el resto de elementos:*

   $\hat{A}(2,3) = 2.5;\ \hat{A}(2,4) = 1.75;\ \hat{A}(3,2) = 2;\ \hat{A}(3,3) = 2.25$
   $\hat{A}(3,4) = 1.5;\ \hat{A}(4,2) = 2;\ \hat{A}(4,3) = 2.5;\ \hat{A}(4,4) = 2.25$

2. *Para cada valor de gris* $i \in N = \{0, 1, 3, 4\}$*, obtenemos el conjunto* A_i*:*

   $A_0 = \{\hat{A}(2,3), \hat{A}(3,3)\} = \{2.5, 2.25\};\ A_1 = \{\hat{A}(2,4), \hat{A}(4,2), \hat{A}(4,3)\} = \{1.75, 2, 2.5\}$
   $A_3 = \{\hat{A}(3,4), \hat{A}(4,4)\} = \{1.5, 2.25\};\ A_4 = \{\hat{A}(2,2), \hat{A}(3,2)\} = \{2.25, 2\}$

3. *Obtenemos el valor de* s *para cada píxel:*

   $$s(0) = \sum_{i=0} |i - A_i| = |0 - 2.5| + |0 - 2.25| = 4.75;\ s(1) = 3.25;\ s(3) = 2.25;\ s(4) = 3.75$$

4. *Calculamos la frecuencia relativa de cada valor de* i *en el vecindario:*

   $p_0 = 2/9;\ p_1 = 1/3;\ p_3 = 2/9;\ p_4 = 2/9$

5. *Por último, para* $\epsilon \approx 0$*, el valor de* f_{coar} *de la matriz viene dado por:*

   $$f_{coar} = \left[\epsilon + \sum_{i=0}^{max(i)} p_i s(i) \right]^{-1} = \left[\frac{2}{9} \cdot 4.75 + \frac{1}{3} \cdot 3.25 + \frac{2}{9} \cdot 2.25 + \frac{2}{9} \cdot 3.75 \right]^{-1} = 0.288$$

3.2. Interpretación y limitaciones

La *coarseness* es una medida del promedio de las diferencias entre cada valor de gris y los valores de gris de los píxeles contiguos a él: valores altos de *coarseness* se corresponden con diferencias pequeñas, es decir, con imágenes menos finas, mientras que valores bajos se corresponden con imágenes más "gruesas". Sin embargo, estos valores dependen del tamaño de vecindario elegido: una gran diferencia entre el valor de gris de un píxel y el promedio de valores de gris de los píxeles circundantes en un vecindario pequeño indica mayores cambios de intensidad que la misma diferencia en un vecindario grande. Por lo tanto, es fundamental determinar el valor óptimo de d; Amadasun y King en su artículo recomiendan tamaños pequeños, dado que permiten una mayor sensibilidad, y en la literatura es habitual tomar valores de d: 1, 3 y 5 (Varela et al. 2005), por lo que son estos los valores que consideraremos para d. Veamos un ejemplo acerca de la interpretación de la *coarseness* y de la influencia de d: en la Figura 3.1 hemos recogido cuatro imágenes, obtenidas de la base de datos de imágenes USC-SIPI [1], ordenadas por rugosidad, de texturas más finas a texturas más gruesas. En la Tabla 3.1 se encuentran recogidos los valores de *coarseness* para cada imagen con distintos d.

Tabla 3.1: Valores de *coarseness* para las imágenes de la Figura 3.1 según distintos valores de d.

	$d = 1$	$d = 3$	$d = 5$
Imagen F_1	0.008	0.005	0.004
Imagen F_2	0.009	0.006	0.006
Imagen F_3	0.010	0.006	0.006
Imagen F_4	0.013	0.007	0.006

Vemos que los valores de *coarseness* se van haciendo más similares a medida que aumenta d, debido a que tamaños de vecindario más grandes miden diferencias más alejadas, y por tanto no se perciben tan bien los cambios a pequeña escala.

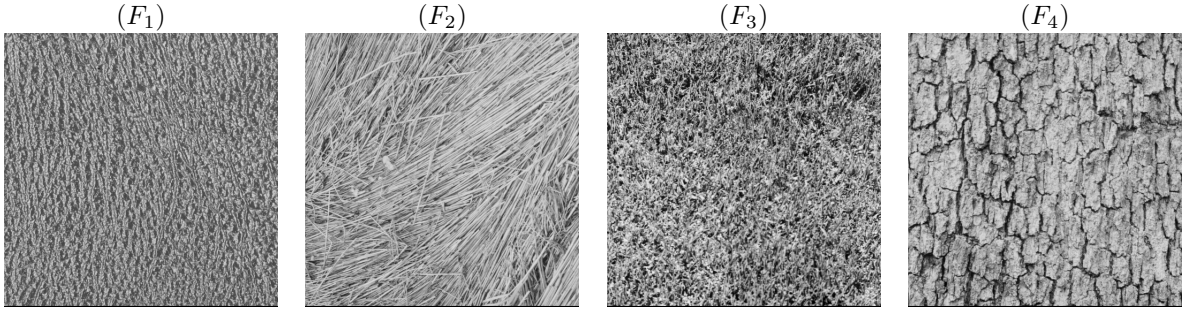


Figura 3.1: Ejemplos de imágenes con distintas rugosidades. Imágenes obtenidas de la base pública de imágenes USC-SIPI.

Esta definición de *coarseness* tiene, a nuestro juicio, dos limitaciones importantes:

- Una consecuencia inmediata de su definición es que su valor depende del número de valores de gris distintos que toma la imagen. Hemos visto en el apartado 2.3.1 que el rango de grises de los píxeles varía mucho dependiendo del aparato con el que se han realizado las radiografías. La normalización utilizada (*intensity scaling*) permite que los valores sean comparables entre sí, pero el número de valores distintos que toman sigue siendo muy diferente. Según el rango

[1] The USC-SIPI Image Database: http://sipi.usc.edu/database

de cada aparato, podemos agruparlos en 3 grupos (ver Figura 2.3): "Agfa", "Carestream", y "Otros". En la Figura 3.2 podemos ver los valores que toma la *coarseness* dependiendo de cada marca comercial; se han cogido 15 muestras aleatorias de cada aparato, con el único requisito de que los pacientes no hubieran fallecido. Se podría considerar utilizar únicamente radiografías realizadas con un aparato determinado, para eliminar estas diferencias; sin embargo, esto no es una opción óptima, dado que dependiendo de la gravedad del paciente puede ser necesario utilizar un aparato móvil, con lo cual descartar las radiografías en función de la marca puede dar lugar a graves sesgos.

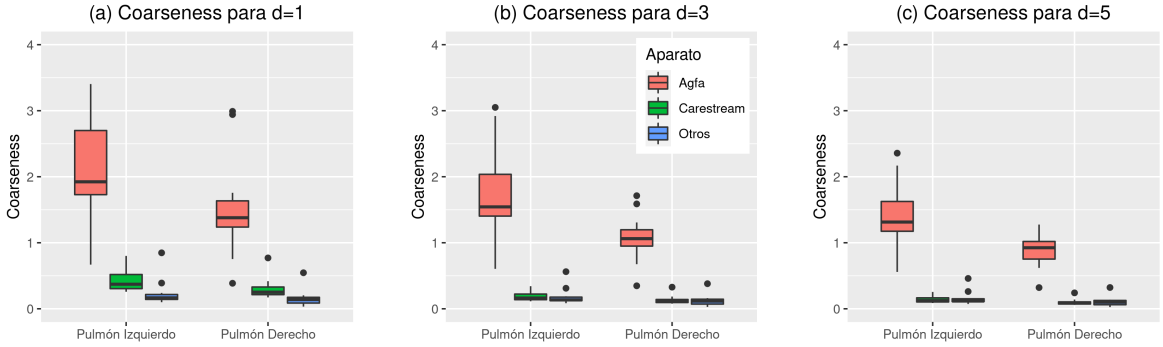


Figura 3.2: Valores de *coarseness* para radiografías tomadas con distintos aparatos, tomando 15 muestras de pacientes no fallecidos en cada grupo. Se presentan los resultados para distintos tamaños de vecindario: en (a) para $d = 1$; en (b) para $d = 3$, y en (c) para $d = 5$.

- Hemos visto que la *coarseness* estudia la media de las diferencias entre los valores de los píxeles de las imágenes. No obstante, en algunos estudios biomédicos, como puede ser el caso de las radiografías, resulta de mayor interés estudiar la variabilidad de las diferencias entre estos valores de gris. En el caso del COVID-19, una de las afectaciones radiológicas más comunes es la consolidación pulmonar (Wong et al. 2020), que se percibe en la radiografía como un área difusa y blanquecina; por lo tanto, aquellos pacientes con radiografías patológicas tendrán una menor variabilidad en los valores de gris que aquellos con radiografía normal. Por lo tanto, para estudiar la afectación pulmonar en los casos de COVID-19, medir el promedio de las diferencias no parece la mejor aproximación, y puede resultar más informativa la variabilidad de estas diferencias.

3.3. *CoarGrid*

Debido a las limitaciones observadas en la *coarseness* de Amadasun y King, proponemos una modificación a su definición original, que se presenta a continuación. El código con su implementación en `R`, `coarGrid.R`, se encuentra recogido en el Anexo B.

3.3.1. Definición

En primer lugar, con el objetivo de eliminar la relación entre el valor de la *coarseness* y el rango de grises de la imagen, redefinimos la matriz **NGTDM** de la siguiente forma: sea A_i el valor medio de gris en un entorno del píxel $i = f(k, l)$, obtenido según la ecuación (3.1). Calculamos el valor de la matriz **NGTDM** correspondiente al píxel i como:

$$s'_i = \frac{s_i}{\sum_{i \in N_i} s_i} \tag{3.4}$$

donde s_i se obtiene según la ecuación (3.2).

Con el objetivo de estudiar la variabilidad de los valores de gris dentro de la imagen digital, dividimos la matriz original $\mathbf{M}$, de tamaño $m \times n$, en submatrices contiguas (en adelante, cuadrículas) de tamaño $q \times q$, con $q \in \mathbb{N}$, $q > d$ y $q < min(m, n)$, y realizamos el cálculo de la *coarseness* para estas nuevas submatrices, siguiendo la ecuación (3.3). Es decir, para la matriz de valores de gris $\mathbf{M}$, y para un tamaño de cuadrícula q, se obtiene:

$$coar_j = \left[\epsilon + \sum_{i=0}^{max(i)} p_i \hat{s}_i \right]^{-1}, \ \forall\ j \in \{1, ..., min([m/q], [n/q])\} \tag{3.5}$$

donde [.] denota la función parte entera. Una vez obtenidos los valores de *coar* para todas las cuadrículas, obtenemos la *coarGrid* de la matriz original como el rango intercuartílico (IQR) de todos ellos:

$$coarGrid = IQR(coar_j),\ j \in \{1, ..., min([m/q], [n/q])\} \tag{3.6}$$

Ejemplo 2 *Consideremos la siguiente matriz* 10×10*; si tomamos un tamaño de cuadrícula de* 5×5*, nos quedamos con* 4 *submatrices, de la siguiente forma:*

$$\mathbf{M} = \begin{pmatrix} 5 & 2 & 0 & 0 & 3 & 3 & 5 & 1 & 3 & 2 \\ 5 & 0 & 1 & 2 & 4 & 5 & 1 & 2 & 3 & 5 \\ 4 & 2 & 1 & 1 & 1 & 0 & 2 & 5 & 3 & 3 \\ 0 & 3 & 1 & 2 & 5 & 1 & 5 & 1 & 0 & 3 \\ 3 & 5 & 4 & 4 & 2 & 5 & 1 & 0 & 4 & 0 \\ 1 & 3 & 3 & 0 & 4 & 1 & 3 & 5 & 0 & 2 \\ 4 & 0 & 1 & 0 & 2 & 2 & 3 & 2 & 4 & 5 \\ 5 & 4 & 2 & 3 & 2 & 2 & 1 & 2 & 1 & 3 \\ 4 & 1 & 2 & 4 & 4 & 1 & 4 & 3 & 1 & 1 \\ 4 & 4 & 5 & 0 & 0 & 5 & 4 & 4 & 4 & 2 \end{pmatrix} \longrightarrow \left(\begin{array}{|ccccc|ccccc|} \hline 5 & 2 & 0 & 0 & 3 & 3 & 5 & 1 & 3 & 2 \\ 5 & 0 & 1 & 2 & 4 & 5 & 1 & 2 & 3 & 5 \\ 4 & 2 & 1 & 1 & 1 & 0 & 2 & 5 & 3 & 3 \\ 0 & 3 & 1 & 2 & 5 & 1 & 5 & 1 & 0 & 3 \\ 3 & 5 & 4 & 4 & 2 & 5 & 1 & 0 & 4 & 0 \\ \hline 1 & 3 & 3 & 0 & 4 & 1 & 3 & 5 & 0 & 2 \\ 4 & 0 & 1 & 0 & 2 & 2 & 3 & 2 & 4 & 5 \\ 5 & 4 & 2 & 3 & 2 & 2 & 1 & 2 & 1 & 3 \\ 4 & 1 & 2 & 4 & 4 & 1 & 4 & 3 & 1 & 1 \\ 4 & 4 & 5 & 0 & 0 & 5 & 4 & 4 & 4 & 2 \\ \hline \end{array} \right)$$

Sean: $\mathbf{M_1}$ *la submatriz de* $\mathbf{M}$ *de orden* 5×5 *formada por las* 5 *primeras filas y columnas de* $\mathbf{M}$*;* $\mathbf{M_2}$ *la submatriz formada por las* 5 *primeras filas y las* 5 *últimas columnas;* $\mathbf{M_3}$ *la submatriz formada por las* 5 *últimas filas y las* 5 *primeras columnas, y* $\mathbf{M_4}$ *la submatriz formada por las últimas* 5 *filas y* 5 *columnas. Considerando* $d = 1$*, obtenemos el valor de* $coar_j\ \forall j \in \{1, 2, 3, 4\}$ *de la siguiente forma:*

1. *Obtenemos los conjuntos* A_i *de cada valor de píxel* i *para cada submatriz, de la misma forma que en el ejemplo 1, siguiendo la ecuación (3.1):*

   $\mathbf{M_1}$: $A_0 = \{2.5\};\ A_1 = \{1, 1.5, 2.75, 2.125\};\ A_2 = \{1.875, 1.375, 2.375\};\ A_3 = \{2.5\}$

   $\mathbf{M_2}$: $A_0 = \{2.375\};\ A_1 = \{2.875, 2.5\};\ A_2 = \{2.5, 2.875\};\ A_3 = \{3, 2.75\};\ A_5 = \{1.875, 2.125\}$

   $\mathbf{M_3}$: $A_0 = \{2.875, 2.125\};\ A_1 = \{3.75, 1.875\};\ A_2 = \{1.875, 2.875\};\ A_3 = \{2.125\};\ A_4 = \{2.375, 2.25\}$

   $\mathbf{M_4}$: $A_1 = \{2.375, 2.625, 2.5\};\ A_2 = \{2.375, 2.375\};\ A_3 = \{2.25, 2.625\};\ A_4 = \{2.75, 2.5\}$

2. *Para cada submatriz, calculamos* s'_i*, siguiendo la ecuación (3.4):*

   $\mathbf{M_1}$: $\hat{s}_0 \approx 0.333;\ \hat{s}_1 = 0.450;\ \hat{s}_2 = 0.150;\ \hat{s}_3 \approx 0.067$

   $\mathbf{M_2}$: $\hat{s}_0 \approx 0.178;\ \hat{s}_1 \approx 0.252;\ \hat{s}_2 \approx 0.103;\ \hat{s}_3 \approx 0.019;\ \hat{s}_5 \approx 0.449$

   $\mathbf{M_3}$: $\hat{s}_0 \approx 0.360;\ \hat{s}_1 \approx 0.261;\ \hat{s}_2 \approx 0.072;\ \hat{s}_3 \approx 0.063;\ \hat{s}_4 \approx 0.243$

   $\mathbf{M_4}$: $\hat{s}_1 \approx 0.493;\ \hat{s}_2 \approx 0.082;\ \hat{s}_3 \approx 0.123;\ \hat{s}_4 \approx 0.301$

3. *Calculamos la frecuencia relativa de cada valor de gris para cada submatriz:*

   $\mathbf{M_1}:\ p_0 = 1/9;\ p_1 = 4/9;\ p_2 = 1/3;\ p_3 = 1/9$
   $\mathbf{M_2}:\ p_0 = 1/9;\ p_1 = 2/9;\ p_2 = 2/9;\ p_3 = 2/9;\ p_5 = 2/9$
   $\mathbf{M_3}:\ p_0 = 2/9;\ p_1 = 2/9;\ p_2 = 2/9;\ p_3 = 1/9;\ p_4 = 2/9$
   $\mathbf{M_4}:\ p_1 = 1/3;\ p_2 = 2/9;\ p_3 = 2/9;\ p_4 = 2/9$

4. *Obtenemos, para cada submatriz, el valor de la coarseness, siguiendo la ecuación (3.5):*

   $\mathbf{M_1}:\ coar_1 = p_0 \cdot \hat{s}_0 + p_1 \cdot \hat{s}_1 + p_2 \cdot \hat{s}_2 + p_3 \cdot \hat{s}_3 \approx 0.294$
   $\mathbf{M_2}:\ coar_2 = p_0 \cdot \hat{s}_0 + p_1 \cdot \hat{s}_1 + p_2 \cdot \hat{s}_2 + p_3 \cdot \hat{s}_3 + p_5 \cdot \hat{s}_5 \approx 0.203$
   $\mathbf{M_3}:\ coar_3 = p_0 \cdot \hat{s}_0 + p_1 \cdot \hat{s}_1 + p_2 \cdot \hat{s}_2 + p_3 \cdot \hat{s}_3 + p_4 \cdot \hat{s}_4 \approx 0.215$
   $\mathbf{M_4}:\ coar_4 = p_1 \cdot \hat{s}_1 + p_2 \cdot \hat{s}_2 + p_3 \cdot \hat{s}_3 + p_4 \cdot \hat{s}_4 \approx 0.277$

5. *Por último, obtenemos el valor de la coarGrid siguiendo la ecuación (3.6):*

$$coarGrid = IQR(coar_1, coar_2, coar_3, coar_4) \approx 0.069$$

3.3.2. Determinación de parámetros óptimos

La definición de la *coarGrid* depende de dos parámetros fundamentales: el tamaño de cuadrícula (q) y el tamaño de vecindario (d). Para el tamaño de vecindario se aplican las mismas consideraciones comentadas en el apartado 3.2, por lo que de nuevo vamos a estudiar valores de d pequeños, $d \in \{1, 3, 5\}$. Con respecto al tamaño de cuadrícula, ésta depende tanto del tamaño de la imagen como del valor elegido de d. Si se opta por cuadrículas muy grandes, el valor de la *coarGrid* estará más cercano a la media de las diferencias de gris de los píxeles, por lo que consideramos valores pequeños de q en relación al tamaño de las imágenes, pero lo suficientemente grandes como para poder manejar los tamaños de vecindario elegidos; en este caso, $q \in \{15, 20, 25, 30\}$.

Dado que uno de los principales objetivos de esta redefinión es poder comparar imágenes que tomen un número de valores de gris muy diferentes para así poder comparar radiografías tomadas con distintos aparatos, vamos a buscar la combinación de q y d que nos permita minimizar las diferencias entre ellos. En la Figura 3.3 se pueden ver los valores de *coarGrid* para distintos valores de q y de d:

1. Cuadrícula 15×15: ninguno de los valores estudiados para d unifica los valores de *coarGrid* entre los distintos aparatos para tamaños de cuadrícula $q = 15$ (test de Kruskal-Wallis con *p-valor* significativo para todos los valores de d y para ambos pulmones).

2. Cuadrícula 20×20: para tamaños de cuadrícula $q = 20$, el valor óptimo de vecindario es $d = 1$ (test de Kruskal-Wallis con *p-valor* significativo para todos los valores de d y para ambos pulmones, excepto para $d = 1$).

3. Cuadrícula 25×25: para tamaños de cuadrícula $q = 25$, el valor óptimo de vecindario es $d = 3$ (test de Kruskal-Wallis con *p-valor* significativo para todos los valores de d y para ambos pulmones, excepto para $d = 3$).

4. Cuadrícula 30×30: para tamaños de cuadrícula $q = 30$, el valor óptimo de vecindario es $d = 5$ (test de Kruskal-Wallis con *p-valor* significativo para todos los valores de d y para ambos pulmones, excepto para $d = 5$).

Disponemos entonces de 3 combinaciones óptimas de tamaño de cuadrícula y de vecindario, en el sentido de que minimizan las diferencias observadas en los valores de gris para los aparatos de diferentes marcas: $q = 20$ y $d = 1$, $q = 25$ y $d = 3$, y $q = 30$ y $d = 5$. Se puede ver en la Figura 3.3, subfiguras (d), (h) y (l), que la *coarGrid* toma valores similares con las 3 combinaciones óptimas; por lo tanto, y

dado que el tiempo computacional se incrementa al aumentar los valores de d y de q, elegimos como parámetros óptimos $q = 20$ y $d = 1$, por ser la combinación menos costosa computacionalmente.

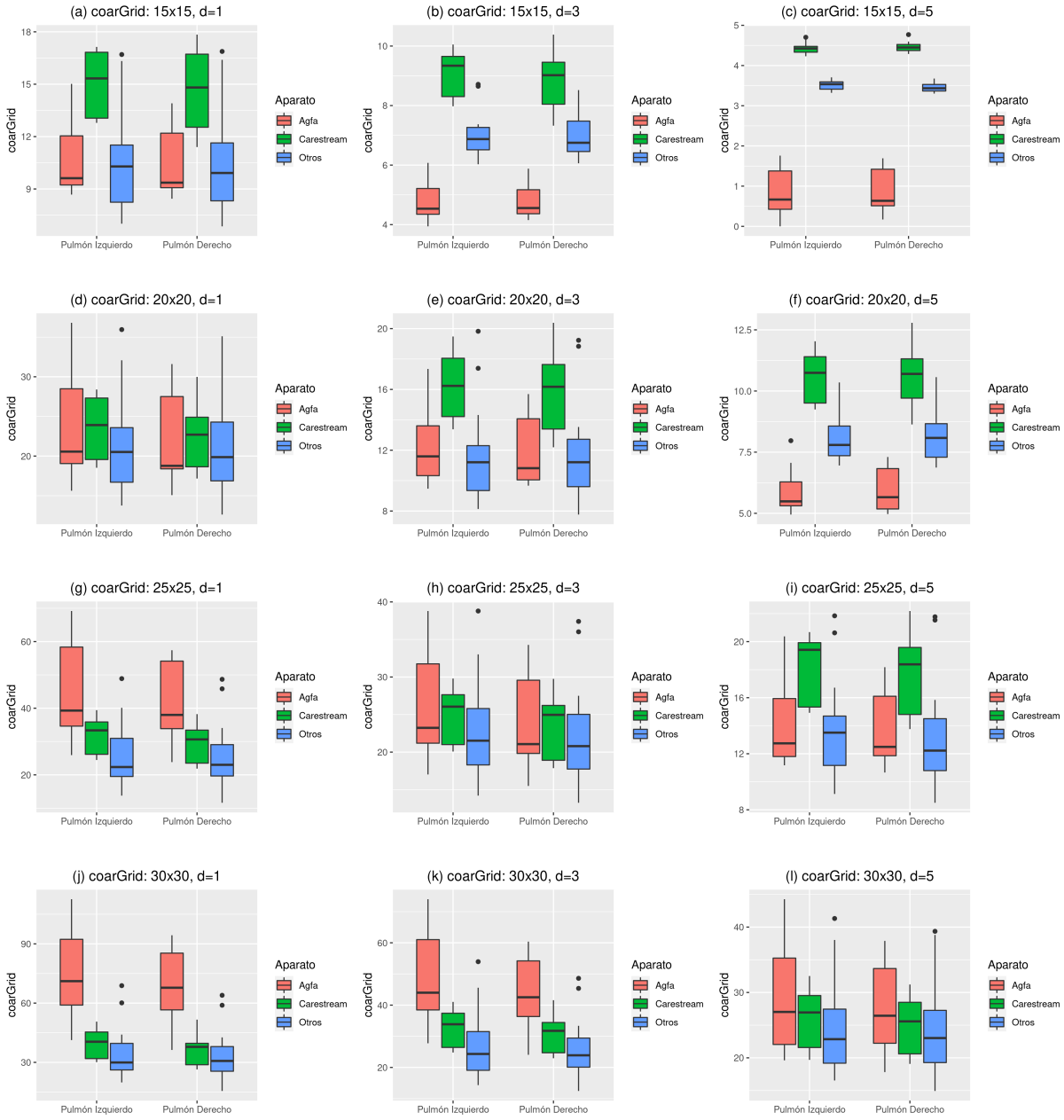


Figura 3.3: Valores de *coarGrid* para los distintos aparatos, según diferentes tamaños de cuadrícula q y de vecindario d: (a) para $q = 15$ y $d = 1$, *p-valor*< 0.05; (b) para $q = 15$ y $d = 3$, *p-valor*< 0.05; (c) para $q = 15$ y $d = 5$, *p-valor*< 0.05; (d) para $q = 20$ y $d = 1$, *p-valor* no significativo; (e) para $q = 20$ y $d = 3$, *p-valor*< 0.05; (f) para $q = 20$ y $d = 5$, *p-valor*< 0.05; (g) para $q = 25$ y $d = 1$, *p-valor*< 0.05; (h) para $q = 25$ y $d = 3$, *p-valor* no significativo; (i) para $q = 25$ y $d = 5$, *p-valor*< 0.05; (j) para $q = 30$ y $d = 1$, *p-valor*< 0.05; (k) para $q = 30$ y $d = 3$, *p-valor*< 0.05; (l) para $q = 20$ y $d = 5$, *p-valor* no significativo. Los *p-valores* se obtuvieron con el test de Kruskal-Wallis.

Capítulo 4

Aplicación

4.1. Metodología estadística

Los modelos de regresión logística son unos de los modelos más utilizados en los estudios biomédicos, ya que las variables de respuesta binaria son muy comunes en este ámbito (presencia o ausencia de una determinada condición o enfermedad, fallecimiento sí/no, etc.). Dado que en este trabajo estamos interesados en estudiar la probabilidad de *exitus* de los pacientes de COVID-19, estos son los modelos que vamos a aplicar.

4.1.1. Regresión logística multivariante

Consideremos como variable respuesta la variable binaria $Y \in \{0, 1\}$, donde 1 denota la ocurrencia del evento que se pretende estudiar, y sea $X = \{X_1, X_2, \ldots, X_k\}$ el conjunto de predictores. Con el objetivo de estimar la probabilidad de que ocurra el evento en función de las covariables disponibles, P, se construye el siguiente modelo:

$$P = \mathbb{P}(Y = 1) = \left(1 + e^{-\beta X}\right)^{-1} \tag{4.1}$$

donde: $\beta = \{\beta_0, \beta_1, \beta_2, \ldots, \beta_k\}$ es el conjunto de los parámetros de la regresión, que se estiman a partir del método de máxima verosimilitud. Si en la ecuación (4.1) despejamos βX, se obtiene la denominada función *logit*:

$$logit(\mathbb{P}(Y = 1)) = logit(P) = \log\left(\frac{P}{1 - P}\right) = \beta X = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k \tag{4.2}$$

Sustituyendo ahora P por su valor $\mathbb{P}(Y = 1)$ en la ecuación (4.2), y aplicando la exponencial, obtenemos la $Odds(Y)$ (el cociente entre la probabilidad de que ocurra el evento y la probabilidad de que no ocurra) en función del conjunto de covariables X:

$$Odds(Y) := \frac{\mathbb{P}(Y = 1)}{\mathbb{P}(Y = 0)} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k} \tag{4.3}$$

Así, la *Odds-Ratio* entre un individuo A, con predictores $X^A = \{X_1^A, X_2^A, \ldots, X_k^A\}$, y un individuo B, con predictores $X^B = \{X_1^B, X_2^B, \ldots, X_k^B\}$, viene dada por:

$$\frac{Odds(A)}{Odds(B)} = \frac{e^{\beta_0 + \beta_1 X_1^A + \beta_2 X_2^A + \ldots + \beta_k X_k^A}}{e^{\beta_0 + \beta_1 X_1^B + \beta_2 X_2^B + \ldots + \beta_k X_k^B}} = e^{\beta_0 + \sum_{i=1}^{k} \beta_i (X_i^A - X_i^B)} \tag{4.4}$$

Suavizado de covariables continuas

Hasta ahora hemos asumido que las covariables continuas del modelo tenían un efecto lineal sobre la variable respuesta, pero esto no siempre es correcto, pues existen casos en los que la relación entre la covariable y la respuesta es no lineal. Para modelar este tipo de relación se pueden utilizar funciones como los *splines* polinómicos, que son funciones polinómicas definidas en intervalos y conectadas entre sí. Los extremos de estos intervalos se denominan nodos, y van a ser una de las características principales del suavizado, junto con el orden del polinomio. Estos *splines* pertenecen al grupo de *splines* de regresión, y pueden ajustarse utilizando el método de mínimos cuadrados, determinando previamente el número de nodos (Durbán 2008, Harrell 2015).

Entre los *splines* polinómicos más utilizados en la práctica se encuentran los *splines* cúbicos, pues permiten obtener funciones "suaves" en los puntos de unión entre intervalos. La función de *spline* cúbica para k nodos, $\{a_1, a_2, ..., a_k\}$, viene dada por:

$$f(X) = \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \alpha_3 X^3 + \alpha_4 (X - a_1)_+^3 + \alpha_5 (X - a_2)_+^3 + \ldots + \alpha_k (X - a_k)_+^3$$

Donde:

$$(X - a_i)_+ = \begin{cases} X - a_i & si \quad X > a_i \\ \\ 0 & si \quad X \leq a_i \end{cases}$$

Vemos que para k nodos es necesario estimar $3 + k$ parámetros de la regresión, α, además del intercepto; es decir, cuanto más incrementamos el número de nodos mayor es la complejidad de la función de suavizado. En la práctica es habitual utilizar entre 3 y 5 nodos (ver Harrell 2015).

4.1.2. Evaluación de un modelo logístico

Para evaluar un modelo logístico, disponemos de medidas de bondad de ajuste (que nos permiten calcular la diferencia o la distancia entre los valores predichos con el modelo y los valores reales), las medidas de calibración (que nos permiten comparar los valores obtenidos con los valores observados), y las medidas de discriminación (que sirven para evaluar la capacidad del modelo para separar a los individuos que cursan el evento de los que no).

Medidas de bondad de ajuste

- R^2 **de Nagelkerke**: cuantifica qué parte de la varianza de la variable respuesta está explicada por el modelo, y viene dado por:

  $$R_N^2 = \frac{1 - e^{-LR/n}}{1 - e^{2LL_0/n}}$$

  Donde $LR = -2(LL_0 - LL)$, siendo LL_0 el logaritmo de la verosimilitud del modelo nulo (es decir, considerando todos los coeficientes de la regresión igual a 0 excepto el intercepto), y LL el logaritmo de la verosimilitud del modelo, que viene dado por:

  $$LL = \sum_{i=1}^{n} (Y_i \log(P_i) + (1 - Y_i) \log(1 - P_i))$$

  Siendo n el número total de individuos, Y_i la variable respuesta observada para el individuo i, y P_i la probabilidad estimada de que ocurra el evento para el individuo i, $P_i = \mathbb{P}(Y_i = 1)$.

- **Puntuación de Brier**: este parámetro mide la precisión de la predicción, calculando las diferencias cuadráticas entre la probabilidad estimada y el evento observado; es decir:

  $$Brier = \frac{1}{n} \sum_{i=1}^{n} (Y_i - P_i)^2$$

Donde, de nuevo, n es el número total de individuos, Y_i la variable respuesta observada para el individuo i, y $P_i = \mathbb{P}(Y_i = 1)$.

Medidas de calibración

Lo más habitual para evaluar la calibración de un modelo es dibujar el gráfico de calibración, esto es, representar gráficamente los valores observados de la probabilidad de que ocurra el evento frente a los valores predichos. Dado que la probabilidad del evento no es directamente observable, esta se ha de estimar mediante técnicas de suavizado, en concreto, la regresión local o *loess*.

En el caso de que las probabilidades predichas sean exactamente iguales a las observadas, el gráfico sería una recta de pendiente 1.

Medidas de discriminación

La capacidad de discriminación de un modelo se suele medir a través de la representación de la curva ROC (*Receiver Operating Characteristic*, o Característica Operativa del Receptor) y del cálculo del área bajo la curva ROC (AUC).

Dado un modelo regresión logística podemos determinar un punto de corte según el cual clasificamos a los individuos como positivos o negativos. Entonces, para cada punto de corte se puede obtener la *sensibilidad* y la *especifidad* del modelo según:

$$Sensibilidad = \frac{VP}{VP + FN};\ Especificidad = \frac{VN}{VN + FP}$$

Donde: VP denota los verdaderos positivos (es decir, individuos clasificados como positivos y que presentan realmente el evento), VN los verdaderos negativos (individuos clasificados como negativos y que no presentan el evento), FN los falsos negativos (individuos clasificados como negativos que sí presentan el evento), y FP los falsos positivos (individuos clasificados como positivos que no presentan el evento).

Así, la curva ROC no es más que la representación gráfica de la *sensibilidad* frente a la *especificidad* para cada posible punto de corte. El área bajo esta curva ROC nos da una medida de la capacidad de discriminación del modelo: cuanto más alta sea (más próxima a 1), más cerca estarán la *sensibilidad* y la *especificidad* de 1, y por tanto mejor discrimina, mientras que valores bajos (próximos a 0.5) indican que el modelo tiene poca capacidad de discriminación.

4.2. Resultados

4.2.1. Análisis descriptivo

La edad media de los 228 pacientes estudiados es de 66.38 ± 15.41, siendo el 42.98 % de ellos mujeres. Hubo 43 casos de *exitus*, el 18.86 %. En la Tabla 4.1 podemos ver las características clínicas de los pacientes. El total de radiografías disponibles es de 607, con una mediana de 3 por paciente (rango intercuartílico: 2-3.25), siendo el máximo de 8. De estas radiografías, el 86.49 % (525) se realizaron con aparatos de la marca 'Agfa', el 2.64 % (16) con 'Carestream', y el 10.87 % (66) con aparatos de otras marcas.

Tomando la primera radiografía disponible de cada paciente (es decir, la radiografía con la fecha más cercana a la fecha de RT-PCR positiva o a la fecha de inicio de los síntomas, después de eliminar las de mala calidad), podemos ver en la Figura 4.1 que el valor de la *coarGrid* no varía en función del sexo, de la EPOC, de la diabetes (test de U de Mann-Whitney en todos los casos no significativo), o del tabaquismo (test de Kruskal-Wallis: $p > 0.05$), y tampoco se observa una relación entre la *coarGrid*

Tabla 4.1: Características clínicas de los pacientes del estudio. La edad (en años) se expresa como $\mu \pm \sigma$, y el resto de variables como frecuencia absoluta y porcentaje.

	Total ($n = 228$)	No *Exitus* ($n = 185$)	*Exitus* ($n = 43$)
Edad	66.38 ± 15.41	63.64 ± 14.74	78.14 ± 12.5
Sexo (Mujer)	98 (42.98 %)	86 (46.49 %)	12 (27.91 %)
Tabaquismo			
Ex-Fumador	49 (21.49 %)	38 (20.54 %)	11 (25.58 %)
Fumador	9 (3.95 %)	9 (4.86 %)	0 (0 %)
EPOC	18 (7.89 %)	10 (5.41 %)	8 (18.6 %)
Diabetes	53 (23.25 %)	35 (18.92 %)	18 (41.86 %)

y la edad (test de correlación de Spearman: $p > 0.05$, Figura 4.2).

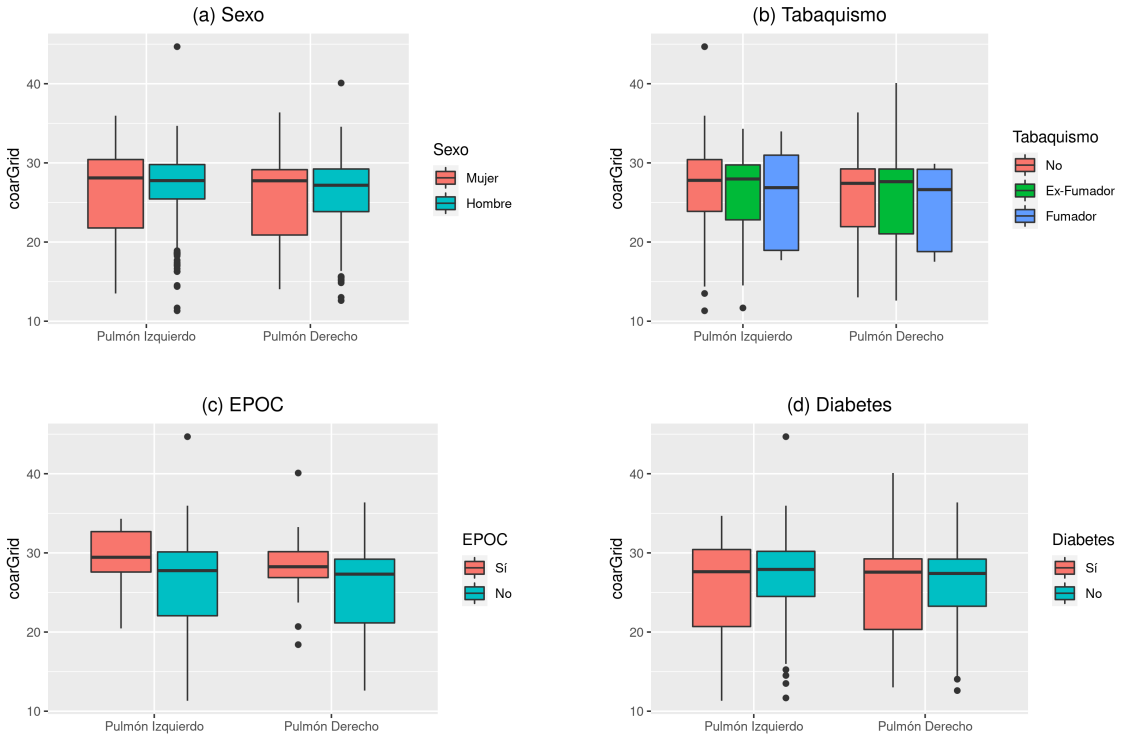


Figura 4.1: Relación entre la *coarGrid* y: (a) el sexo, (b) el tabaquismo, (c) la enfermedad pulmonar obstructiva crónica o EPOC, y (d) la diabetes, para cada pulmón. El test de U de Mann-Whitney y el de Kruskal-Wallis (en el caso de la variable "Tabaquismo") son no significativos en todos los casos ($p > 0.05$).

Dado que la correlación entre la *coarGrid* de los pulmones izquierdo y derecho es muy elevada (Spearman $\rho \approx 0.89$), y los valores del pulmón izquierdo pueden llegar a recoger un poco más de ruído debido a la presencia del corazón, se ha optado por realizar todos los cálculos con el valor de *coarGrid* del pulmón derecho.

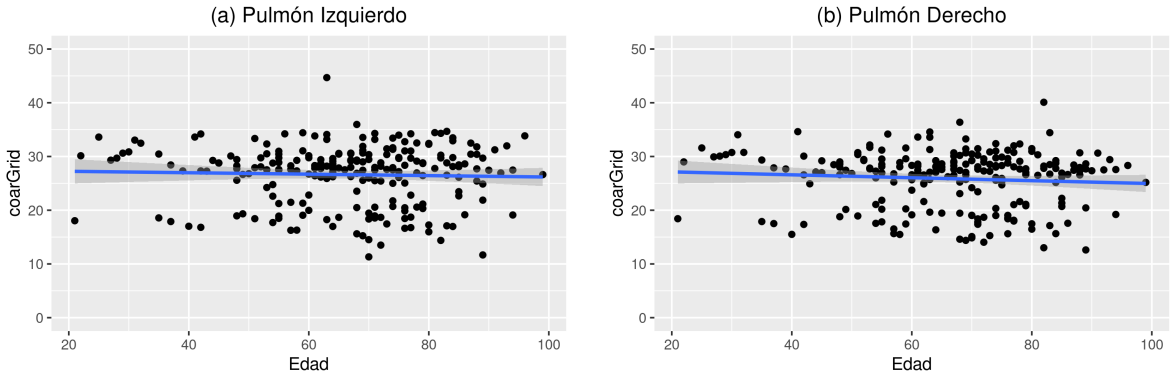


Figura 4.2: Relación entre la *coarGrid* y la edad, (a) para el pulmón izquierdo, y (b) para el pulmón derecho. El test de correlación de Spearman es no significativo en ambos casos ($p > 0.05$).

4.2.2. Modelo de predicción para la mortalidad

El objetivo de este trabajo es construir un modelo que nos permita identificar aquellos pacientes de COVID-19 con mayor riesgo de mortalidad, y en especial si la variable de interés descrita en este estudio, la *coarGrid*, es un factor de riesgo. Para estudiar esto, se han ajustado varios modelos de regresión logística multivariante, con respuesta el *exitus*, y covariables la *coarGrid* junto con las variables clínicas descritas en el apartado anterior. Para la estimación de estos modelos se ha utilizado la función `lrm` del paquete `rms` (Harrell 2021) de `R`, que sigue el método de maximización de la verosimilitud parcial para la estimación de los parámetros de la regresión. Se evaluó la posible no linealidad de las covariables continuas mediante el uso de *splines* cúbicos, utilizando para ello la función `rcs` del mismo paquete; dado que en esta función es necesario introducir manualmente el número de nodos, se creó el modelo inicial considerando las covariables continuas tanto con efecto lineal como no lineal con distinto número de nodos ($k = 3, 4, 5$), y se eligió la combinación que mejoraba el AIC, siguiendo la propuesta de Harrell (2015).

El procedimiento seguido para llegar al modelo final óptimo es el siguiente:

1. Partimos de un modelo inicial en el que introducimos como covariables todas las variables disponibles: la edad, el sexo, el tabaquismo, la EPOC, la diabetes, y el valor de la *coarGrid* del pulmón derecho de la primera radiografía del paciente.

2. Se genera un nuevo modelo descartando la covariable con el *p-valor* menos significativo, y se aplica un test de razón de verosimilitud (*likelihood ratio test*) para determinar si su contribución al modelo es estadísticamente significativa. En el caso de que no lo sea, se descarta. El test de razón de verosimilitud se aplicó utilizando la función `lrtest`.

3. Se repite el proceso hasta que todas las covariables del modelo contribuyen significativamente al mismo (*p-valor* < 0.05).

Para cada modelo, se obtuvieron las medidas de bondad de ajuste descritas en la sección 4.1.2: la´ R^2 de Nagelkerke y la puntuación de Brier, y se corrigieron mediante *bootstrap*, utilizando la función `validate` con un resampleado de 200 muestras (Miller et al. 1991).

En la Tabla 4.2 se encuentran las estimaciones obtenidas para el modelo inicial (R^2 de Nagelkerke corregido: 0.291, puntuación de Brier corregida: 0.128), y en la Tabla 4.3 las del modelo final (R^2 de Nagelkerke corregido: 0.289, puntuación de Brier corregida: 0.123). Las variables que permanecen en el modelo final son la edad (con efecto lineal), el sexo y la *coarGrid* (con efecto lineal); en la Figura 4.3 se pueden ver el gráfico del logaritmo de la *Odds-Ratio* para las covariables continuas del modelo final. Vemos que por cada incremento de un año en la edad, y fijadas las demás covariables, la probabilidad de *exitus* se multiplica por 1.1; ser mujer resulta un factor protector, ya que en los hombres la probabilidad de morir se multiplica por 2.533. Por último, vemos que los valores altos de *coarGrid* se corresponden con un menor riesgo de *exitus*: cada incremento en una unidad de la *coarGrid* reduce la probabilidad de que el paciente fallezca, multiplicándose por 0.914.

Tabla 4.2: Estimación de las covariables para el modelo inicial, donde SE denota el error estándar, OR la *Odds-Ratio*, y el IC95 %(OR) el intervalo de confianza al 95 % para la *Odds-Ratio*.

	β	SE(β)	OR	IC95 %(OR)	*p-valor*
Intercepto	-5.730	1.669	-	-	0.0006
Edad	0.085	0.019	1.089	$(1.049, 1.131)$	< 0.0001
Sexo (Ref: Mujer)	0.928	0.459	2.530	$(1.029, 6.217)$	0.0431
Tabaquismo					
Ex-Fumador	-0.427	0.578	0.653	$(0.210, 2.027)$	0.4604
Fumador	-6.871	22.634	0.001	$(5.6e^{-23}, 1.9e^{16})$	0.7615
EPOC (Ref: No)	1.316	0.708	1.144	$(0.931, 14.937)$	0.0631
Diabetes (Ref: No)	0.553	0.418	3.729	$(1.738, 3.942)$	0.1855
coarGrid	-0.105	0.037	0.900	$(0.836, 0.968)$	0.0048

Tabla 4.3: Estimación de las covariables para el modelo final, donde SE denota el error estándar, OR la *Odds-Ratio*, y el IC95 %(OR) el intervalo de confianza al 95 % para la *Odds-Ratio*.

	β	SE(β)	OR	IC95 %(OR)	*p-valor*
Intercepto	-6.638	1.606	-	-	< 0.0001
Edad	0.095	0.019	1.100	$(1.060, 1.141)$	< 0.0001
Sexo (Ref: Mujer)	0.929	0.417	2.533	$(1.119, 5.736)$	0.0258
coarGrid	-0.090	0.036	0.914	$(0.851, 0.981)$	0.0122

La capacidad de discriminación del modelo es bastante alta; podemos ver en la Figura 4.4 (a) que el área bajo la curva ROC es elevada (AUC=0.819, DeLong IC95 %=$(0.752, 0.885)$), y que el modelo asigna mayores probabilidades de *exitus* a los pacientes que efectivamente han fallecido (Figura 4.4 (b)).

Por último, en la Figura 4.5 podemos ver el gráfico de calibración del modelo final, usando un remuestreo de 200 muestras *bootstrap* y un suavizado con regresión *loess* para la estimación de las probabilidades, cuyos valores se obtuvieron con la función `calibrate` del paquete `rms`. Vemos que las predicciones del modelo son muy similares a las probabilidades ideales, por lo que podemos concluir que el modelo está bien calibrado.

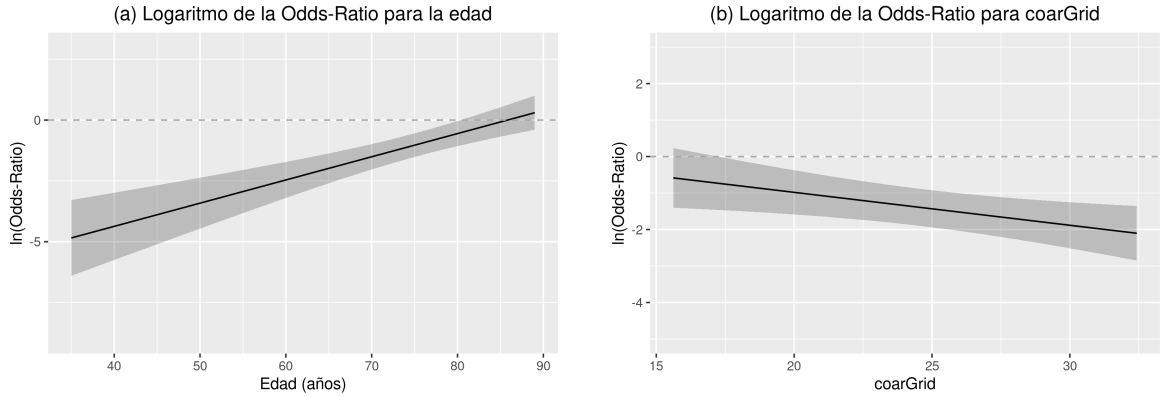


Figura 4.3: Logaritmo neperiano de la *Odds-Ratio* para el modelo final de las covariables continuas: (*a*) *Odds-Ratio* de la edad, fijada la *coarGrid* en su mediana y el sexo en "Hombre", y (*b*) *Odds-Ratio* de la *coarGrid*, fijada la edad en su mediana y el sexo en "Hombre".

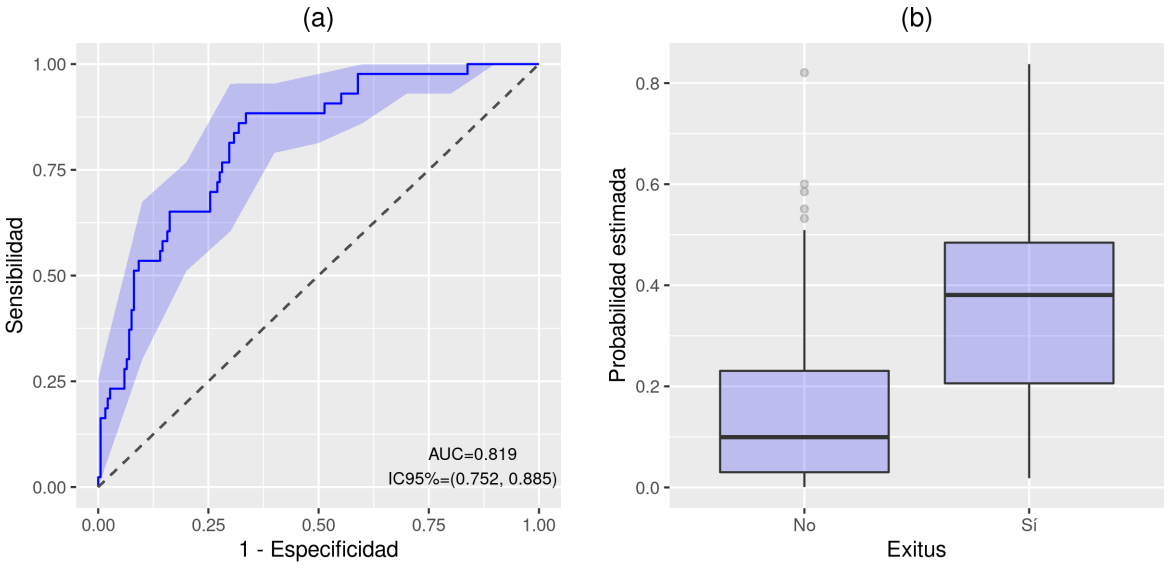


Figura 4.4: Capacidad de discriminación del modelo final: (*a*) gráfico de la curva ROC del modelo, con el intervalo de confianza al 95 %; (*b*) distribución de la probabilidad asignada por el modelo a cada paciente en relación a su estado real.

Visto que el modelo es una buena aproximación, construimos un *score* que nos permita evaluar el riesgo de *exitus* de un paciente a partir de las tres variables del modelo (sexo, edad y *coarGrid*). Para obtenerlo, usamos la función `nomogram`, que nos permite asociar una puntuación a cada incremento de las variables. La puntuación obtenida con este *score*, recogida en la Tabla 4.4, nos permite extraer la probabilidad de que un paciente sobreviva en función de su sexo, su edad, y del valor de la *coarGrid*. Así, por ejemplo, un paciente de sexo femenino, de 70 años y con un valor de *coarGrid* de 25, tendría aproximadamente 92 puntos; con esta puntuación, la probabilidad de que sobreviva es de 0.902, mientras que si fuera un hombre esta probabilidad se reduce a 0.774.

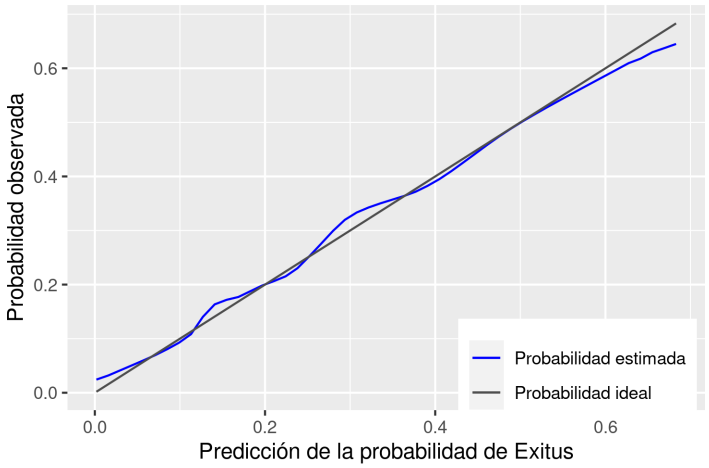


Figura 4.5: Gráfico de calibración del modelo final.

Tabla 4.4: Puntuación asociada a los distintos valores de las variables del modelo. A partir de la suma de los puntos de cada variable), se puede obtener la probabilidad de *exitus*, $P(exitus)$.

Sexo	**Puntos**	**Edad**	**Puntos**	***coarGrid***	**Puntos**	Total	$P(exitus)$
Mujer	0	20	0	5	53.2	0	$9.7e^{-05}$
Hombre	12	30	12.5	10	47.3	20	$4.5e^{-04}$
		40	25.0	15	41.4	40	0.002
		50	37.5	20	35.5	60	0.009
		60	50.0	25	29.6	80	0.042
		70	62.5	30	23.7	100	0.166
		80	75.0	35	17.7	120	0.478
		90	87.5	40	11.8	140	0.808
		100	100	45	5.9	160	0.951
				50	0		

Capítulo 5

Discusión

En este trabajo se ha presentado una forma de extraer información sobre el estado de un paciente a través del análisis de sus radiografías. Para ello, se recogen todos los pasos necesarios para la extracción de la información relevante: en primer lugar, en el apartado 2.2 se han introducido las características básicas de una imagen digital, y el formato en el que habitualmente se encuentran las radiografías. En el apartado 2.3 se han expuesto los problemas observados en el manejo de las radiografías, principalmente relacionados con los valores de los píxeles (presencia de valores atípicos, codificación de los valores extremos, diferencias en los valores de gris según los distintos aparatos, ...), y se han propuesto formas para resolver estos problemas. En el apartado 2.4 se han expuesto dos formas para extraer los valores de gris de las regiones de interés (en este caso los pulmones), y se ha valorado cuál se ajustaba más a las necesidades del presente estudio, en términos de introducir la menor cantidad de ruido posible. Una vez obtenidos los valores de gris de las regiones de interés, y siguiendo la propuesta de Amadasun y King (1989), se ha calculado en el apartado 3 el valor del parámetro definido como *coarseness* de las radiografías, para el estudio de la textura de la región pulmonar. En el apartado 3.2 se han valorado las limitaciones observadas en este parámetro, y en el apartado 3.3 se ha propuesto una modificación en su definición para evitar dichas limitaciones: la *coarGrid*. Recordemos que esta nueva definición depende de 2 parámetros: el tamaño de cuadrícula, q, y el tamaño de vecindario, d. En el apartado 3.3.2 se han explorado los valores óptimos para estos parámetros, realizando la comparación entre diversas radiografías y eligiendo aquellos que minimizaban las diferencias observadas entre ellas. Por último, en el apartado 4 se ha estudiado la relevancia de este parámetro para predecir la probabilidad de *exitus* de los pacientes de COVID-19. Para ello hemos construido un modelo de regresión logística multivariante de la siguiente forma: generamos un modelo inicial en el que introducimos como covariables la *coarGrid* y todas las variables clínicas disponibles, y fuimos descartando una a una las covariables que no contribuían significativamente al modelo. Así, el modelo final obtenido consta de únicamente 3 covariables: la edad, el sexo, y la *coarGrid*. El modelo así construido tiene una buena capacidad de discriminación (recordemos que el AUC del modelo era de 0.819); por tanto, añadimos también un *score* que permite calcular la probabilidad de *exitus* de un paciente de forma rápida y sencilla. En los apéndices A y B se encuentran recogidos todos los scripts generados para el preprocesado de las radiografías, la extracción de los valores de gris, y el cálculo de la *coarGrid*.

Limitaciones y trabajo futuro

La principal limitación del estudio radica en la dificultad de la extracción de los datos, en particular la selección de las regiones de interés, dado que al ser un proceso manual limita mucho el tamaño muestral que se puede manejar, y hace que no resulte práctico en el caso de querer implementarlo para la clínica. El modelo predictivo ajustado da mucho más peso a la edad del paciente que a la *coarGrid*, por lo que quizás no sea justificable en la práctica el incremento en la dificultad que supondría la extracción de este parámetro. Esto podría resolverse utilizando, por ejemplo, técnicas de *deep-learning* que reali-

cen la selección de las regiones de interés de forma automática, pero la optimización e implementación de este proceso no es algo trivial; en los últimos años se han incrementado considerablemente el número de estudios que aplican este tipo de técnicas para la extracción de las ROI (ver, por ejemplo, Sivaramakrishnan et al. 2018, Xu et al. 2019, Suresh and Mohan 2020), pero no existe todavía un estándar para ello, y sigue siendo necesario partir de grandes cantidades de datos para ajustar y optimizar el proceso.

Entre las posibles mejoras del estudio se encontrarían, en primer lugar, ampliar el tamaño muestral y las covariables de estudio: se podrían recoger más variables clínicas de los pacientes, como por ejemplo otro tipo de comorbilidades, tratamientos, etc., e incluso se podría estudiar si existen diferencias en el pronóstico de los pacientes dependiendo de la cepa o variante de SARS-CoV-2 con la que están infectados. Podría resultar también interesante estudiar otras características de las radiografías: hemos visto que, aunque la rugosidad o *coarseness* se considera la característica principal de la textura, esta se puede describir de numerosas formas (Amadasun y King 1989, Haralick et al. 1973). Se podría estudiar si la combinación de distintos parámetros puede mejorar el modelo de predicción de la mortalidad.

También podría resultar interesante estudiar técnicas para eliminar el "ruido" presente en las radiografías (costillas, vasos sanguíneos, elementos ajenos como marcapasos, intubación, etc.). Existen estudios en los que se plantean distintas formas para detectar y eliminar este ruido (ver, por ejemplo, Katsuragawa et al. 1988, van Ginneken et al. 2002); se podría estudiar el efecto que cada técnica puede tener en el cálculo de la *coarGrid*, y valorar si su aplicación resulta conveniente.

Por último, dado que muchos de los pacientes disponen de varias radiografías tomadas en distintos momentos del desarrollo de la enfermedad, y el tiempo entre el diagnóstico y el *exitus* es conocido, otra posibilidad sería la aplicación de técnicas de *joint-modelling* para el análisis de la supervivencia de estos pacientes (ver, por ejemplo, Rizopoulos 2012). Estos modelos se están utilizando cada vez más en la investigación biomédica para el estudio de la supervivencia en diversas de enfermedades (Roustaei et al. 2018, Long y Mills 2018, Sheikh et al. 2020), dado que permiten predecir el tiempo al evento introduciendo en el modelo covariables que cambian en ese tiempo, como puede ser el caso de la *coarGrid*: es esperable que se aprecie un deterioro en las radiografías de los pacientes que no sobreviven, y sería interesante estudiar si la *coarGrid* es capaz de identificar dicho deterioro. Sin embargo, para poder aplicar esta metodología lo óptimo sería disponer de más radiografías por paciente (el 25 % de los pacientes incluidos en este estudio únicamente disponían de 2 radiografías o menos), y sería recomendable incrementar el tamaño muestral.

Apéndice A

Scripts Python

A.1. Función `ModifyDICOM.py`

```
#!/usr/bin/python3
#---------------------------
import sys
import pydicom
import csv
import numpy as np
import pathlib
import math
import os
import cv2
import glob
#---------------------------
## Directorios:
# Primer argumento: directorio donde están almacenados las DICOM:
dicomDIR = sys.argv[0]
# Segundo argumento: directorio donde queremos guardar el output:
outputDCM = sys.argv[1]
try:
        os.stat(outputDCM)
except:
        os.mkdir(outputDCM)
#---------------------------
## Función para transformar los blancos en negros en DICOM:
def inversionDICOM(matriz):
        ## Creamos una matriz auxiliar para guardar el output:
        matrizNew=np.zeros(shape=(matriz.shape[0],matriz.shape[1]))
        minimo = int(np.nanmin(matriz))
        maximo = int(np.nanmax(matriz))
        m = -1
        n = maximo + minimo
        for i in range(0, matriz.shape[0]):
                for j in range(0, matriz.shape[1]):
                        if (np.isnan(matriz[i,j])==False):
                                matrizNew[i,j]= m*matriz[i,j] + n
```

```
                                ## Guardamos los datos en la matriz auxiliar:
                                matrizNew[i,j] = round(matrizNew[i,j])
                        else:
                                matrizNew[i,j] = np.nan
        return matrizNew
#--------------------------
with open(outputDir+'RX_info.csv', 'w', newline='') as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(["FileCode", "ID", "Date", "Time", "RXType", "Modality",
         "PhotometricInterpretation", "Manufacturer", "Model", "Minimo", "Maximo"])
        for files in os.listdir(dicomDIR):
                #--------------------------
                # Para cada imagen, extraemos la información pertinente y la
                # escribimos en un csv:
                fName=pathlib.PurePath(dicomDIR,files)
                imgFull= pydicom.filereader.dcmread(fName)
                ID=files.split("_")[1]
                Date=str(imgFull.StudyDate)
                if (hasattr(imgFull, "AcquisitionTime")):
                        aux=str(imgFull.AcquisitionTime)
                elif (hasattr(imgFull, "ContentTime")):
                        aux=str(imgFull.ContentTime)
                else:
                        aux="NA"
                Time=aux.split(".")[0]
                if (hasattr(imgFull, "ViewPosition")):
                        RXType=str(imgFull.ViewPosition)
                else:
                        RXType="NA"
                Modality=str(imgFull.Modality)
                if (hasattr(imgFull, "PhotometricInterpretation")):
                        photo=str(imgFull.PhotometricInterpretation)
                else:
                        photo="NA"
                if (hasattr(imgFull, "Manufacturer")):
                        manufacturer=str(imgFull.Manufacturer)
                else:
                        manufacturer="NA"
                if (hasattr(imgFull, "ManufacturerModelName")):
                        model=str(imgFull.ManufacturerModelName)
                else:
                        model="NA"
                #--------------------------
                ## Transformación de los archivos DICOM:
                # Leemos la DICOM correspondiente:
                imgData = imgFull.pixel_array
                newdata = np.copy(imgData)
                newdata = newdata.astype(float)
                # Buscamos los outliers:
                q1 = np.nanquantile(newdata,0.000001)
                q9 = np.nanquantile(newdata,0.999999)
```

```
                outmin = np.where(newdata<=q1)
                outmax = np.where(newdata>=q9)
                ## Cambiamos los outliers inferiores por el valor mínimo sin ellos:
                newdata[outmin] = np.nan
                newdata = np.nan_to_num(newdata, copy=True, nan=np.nanmin(newdata))
                ## Cambiamos los outliers superiores por el valor máximo sin outliers:
                newdata[outmax] = np.nan
                newdata = np.nan_to_num(newdata, copy=True, nan=np.nanmax(newdata))
                imgData = newdata
                # Sacamos el signo de la transformación (monochrome1=inverse,
                 monochrome2=identity):
                if photo=="MONOCHROME1":
                        imgData = inversionDICOM(np.copy(imgData))
                else:
                        imgData = np.copy(imgData)
                ## Escribimos los valores de la dicom:
                Minimo = int(np.nanmin(imgData))
                Maximo = int(np.nanmax(imgData))
                writer.writerow([files, ID, Date, Time, RXType, Modality, photo,
                 manufacturer, model, Minimo, Maximo])
                ## Escribimos la dicom modificada:
                imgData = imgData.astype(dtype="uint16")
                imgFull.PixelData = imgData.tobytes()
                imgFull.save_as(outputDCM + files + "_modified.dcm")
        #--------------------------
newcsv.close()
#--------------------------
```

A.2. Función ROILung.py

```
#!/usr/bin/python3
#--------------------------
# Importamos las librerías:
import sys
from medpy.io import load
import pydicom
import pathlib
import numpy as np
import os
#--------------------------
## Directorios:
# Primer argumento: directorio donde están los MHA
MHADir = sys.argv[0]
# Segundo argumento: directorio donde están las DICOM modificadas
DICOMDir = sys.argv[1]
# Tercer argumento: archivo csv con los datos de la DICOM
csvname = sys.argv[2]
csvfile = csv.reader(open(csvname, newline=''))
csvf = list(csvfile)
# Cuarto argumento: directorio donde escribimos el output
```

```
outputArray = sys.argv[3]
try:
        os.stat(outputArray)
except:
        os.mkdir(outputArray)
#--------------------------
## Bucle para obtener los arrays:
for k in range(1,len(csvf)):
        ## Leemos la dicom:
        fName = pathlib.PurePath(DICOMDir + csvf[k][0] + "_modified.dcm")
        imgFull = pydicom.filereader.dcmread(fName)
        imgData = imgFull.pixel_array
        ## Leemos las etiquetas:
        label_data, label_header = load(MHADir + csvf[k][0])
        # Modificamos label_data para que tenga 2 dimensiones:
        labelNew = label_data.reshape((label_data.shape[0], -1), order='F')
        # Hay que hacer la traspuesta de la matriz de etiquetas:
        labelFinal = np.transpose(labelNew)
        # Tenemos una matriz de 0's, 1's y 2's: queremos separarla en dos matrices:
        # una que tenga los 1's y nan's, y otra que tenga 2's y nan's:
        # Pulmón derecho:
        labelRL = np.copy(labelFinal.astype(float))
        labelRL[labelRL!=1] = np.nan
        # Pulmón izquierdo:
        labelLL = np.copy(labelFinal.astype(float))
        labelLL[labelLL!=2] = np.nan
        labelLL = labelLL/2
        # Hacemos el producto de la imagen original con la imagen de la etiqueta:
        imgNewRL = imgData * labelRL
        imgNewLL = imgData * labelLL
        # Normalizamos los valores de las matrices:
        minimo = int(csvf[k][9])
        maximo = int(csvf[k][10])
        imgNormRL = (imgNewRL - minimo) / (maximo - minimo)
        imgNormLL = (imgNewLL - minimo) / (maximo - minimo)
        # Escribimos los archivos:
        np.savetxt(outputArray + csvf[k][0] + "_RL.txt", imgNewRL)
        np.savetxt(outputArray + csvf[k][0] + "_LL.txt", imgNewLL)
#--------------------------
```

A.3. Función `ROIRectangular.py`

```
#!/usr/bin/python3
#--------------------------
# Importamos las librerías:
import sys
import pydicom
import csv
import numpy as np
```

```
import pathlib
from PIL import Image
import math
import os
import cv2
import glob
#---------------------------
## Directorios:
# Primer argumento: directorio donde están almacenados los txt con las coordenadas:
coordDIR = sys.argv[0]
# Segundo argumento: directorio donde están las DICOM modificadas
dicomDIR = sys.argv[1]
# Tercer argumento: archivo csv con los datos de la DICOM
csvname = sys.argv[2]
csvfile = csv.reader(open(csvname, newline=''))
csvf = list(csvfile)
# Cuarto argumento: directorio donde queremos guardar los arrays:
outputArrayData = sys.argv[3]
try:
        os.stat(outputArrayData)
except:
        os.mkdir(outputArrayData)
#---------------------------
## Función auxiliar para rotar la región de interés:
def rotateROI(valores, dimension):
        nc = int(dimension[0]/5)
        nf = int(dimension[1]/5)
        ## Tomamos los puntos A0, A1, A2 y A3:
        x = [np.abs(float(valores[1])), np.abs(float(valores[4])),
         np.abs(float(valores[7])), np.abs(float(valores[10]))]
        y = [np.abs(float(valores[2])), np.abs(float(valores[5])),
         np.abs(float(valores[8])), np.abs(float(valores[11]))]
        ## Calculamos la pendiente de la recta que une A0A1:
        m = (y[1]-y[0])/(x[1]-x[0])
        ## Obtenemos el ángulo de giro:
        alpha = math.atan(m)
        ## Calculamos las distancias: A0A1, A0A3, A1A2, A2A3:
        d01 = ((x[1]-x[0])**2 + (y[1]-y[0])**2)**(1/2)
        d23 = ((x[3]-x[2])**2 + (y[3]-y[2])**2)**(1/2)
        d03 = ((x[3]-x[0])**2 + (y[3]-y[0])**2)**(1/2)
        d12 = ((x[2]-x[1])**2 + (y[2]-y[1])**2)**(1/2)
        ## Vectores auxiliares:
        x_new = list(x)
        y_new = list(y)
        ## Obtenemos el rectángulo con la largura y anchura máximas a partir de
        # los puntos del trapecio:
        if d01<=d23:
                ## A0' es la intersección entre la recta de pendiente m que pasa
                # por A0 y la recta de pendiente -1/m que pasa por A3:
                x_new[0] = (y[0] - y[3] + (-1/m)*x[3] - m*x[0])/((-1/m) - m)
                y_new[0] = y[3] + (-1/m)*(x_new[0] - x[3])
```

```
                ## A1' es la intersección entre la recta de pendiente m que pasa
                # por A1 y la recta de pendiente -1/m que pasa por A2:
                x_new[1] = (y[1] - y[2] + (-1/m)*x[2] - m*x[1])/((-1/m) - m)
                y_new[1] = y[2] + (-1/m)*(x_new[1] - x[2])
                if d03>=d12: # Mantenemos fijo A3
                        ## A2' es la intersección entre la recta de pendiente m que
                        # pasa por A3 y la recta de pendiente -1/m que pasa por A2:
                        x_new[2] = (y[3] - y[2] + (-1/m)*x[2] - m*x[3])/((-1/m) - m)
                        y_new[2] = y[2] + (-1/m)*(x_new[2] - x[2])
                else: # Mantenemos fijo A2
                        ## A3' es la intersección entre la recta de pendiente m que
                        # pasa por A2 y la recta de pendiente -1/m que pasa por A3:
                        x_new[3] = (y[2] - y[3] + (-1/m)*x[3] - m*x[2])/((-1/m) - m)
                        y_new[3] = y[3] + (-1/m)*(x_new[3] - x[3])
        else: # Mantenemos fijos A0 y A1:
                ## A2' es la intersección entre la recta de pendiente m que pasa
                # por A2 y la recta de pendiente -1/m que pasa por A1:
                x_new[2] = (y[2] - y[1] + (-1/m)*x[1] - m*x[2])/((-1/m) - m)
                y_new[2] = y[1] + (-1/m)*(x_new[2] - x[1])
                ## A3' es la intersección entre la recta de pendiente m que pasa
                # por A2 y la recta de pendiente -1/m que pasa por A0:
                x_new[3] = (y[2] - y[0] + (-1/m)*x[0] - m*x[2])/((-1/m) - m)
                y_new[3] = y[0] + (-1/m)*(x_new[3] - x[0])
        ## Obtenemos el punto medio del rectángulo (centro inicial):
        m02 = (y_new[2]-y_new[0])/(x_new[2]-x_new[0])
        m13 = (y_new[3]-y_new[1])/(x_new[3]-x_new[1])
        cX = (y_new[0] - y_new[1] + m13*x_new[1] - m02*x_new[0])/(m13 - m02)
        cY = y_new[1] + m13*(cX - x_new[1])
        ## Le restamos el punto medio de la imagen (traslación):
        X_ = cX - nc/2
        Y_ = cY - nf/2
        ## Rotamos los puntos:
        centerX_ = X_*math.cos(alpha) + Y_*math.sin(alpha)
        centerY_ = Y_*math.cos(alpha) - X_*math.sin(alpha)
        ## Obtenemos el punto medio, el ancho y el largo del rectángulo
        # de interés como porcentajes:
        centerX = (centerX_ + nc/2)/nc
        centerY = (centerY_ + nf/2)/nf
        width = np.max([d01, d23])/nc
        length = np.max([d03, d12])/nf
        return([centerX, centerY, width, length, alpha])
#--------------------------
## Bucle para extraer las regiones de interés:
for k in range(1,len(csvf)):
        ## Leemos los archivos con las coordenadas:
        txtfile = open(csvf[k][0], "r")
        txt = txtfile.read().split()
        txtfile.close()
        ## Leemos la DICOM correspondiente:
        fName = pathlib.PurePath(dicomDIR,csvf[k][0])
        imgFull = pydicom.filereader.dcmread(fName)
```

```
        imgData = imgFull.pixel_array
        ## Si el ROI es ya un rectángulo cogemos los valores:
        if len(txt)==5:
                centerX = np.abs(float(txt[1]))
                centerY = np.abs(float(txt[2]))
                width = np.abs(float(txt[3]))
                length = np.abs(float(txt[4]))
                alpha = "NA"
                valorMedian = -1
        ## Si es un trapecio, obtenemos el ángulo de giro y los valores:
        else:
                valoresRotados = rotateROI(txt, imgData.shape)
                centerX = float(valoresRotados[0])
                centerY = float(valoresRotados[1])
                width = float(valoresRotados[2])
                length = float(valoresRotados[3])
                alpha = 180*valoresRotados[4]/math.pi
        ## En caso de que alpha no sea NA, giramos la imagen:
        if alpha!="NA":
                ## Obtenemos la matriz de datos girada:
                M = cv2.getRotationMatrix2D((imgData.shape[0]/2,
                 imgData.shape[1]/2),alpha,1)
                imgDataRot = cv2.warpAffine(np.copy(imgData),M,(imgData.shape[1],
                 imgData.shape[0]))
                imgData = imgDataRot.astype(float)
                ## Cambiamos los 0 por nan:
                imgData[imgData==0] = np.nan
        ## Extraemos el rango máximo de filas y columnas:
        fmax = imgData.shape[0]
        cmax = imgData.shape[1]
        ## Obtenemos las coordenadas de las esquinas del rectángulo:
        if alpha=="NA":
                c1 = round(cmax*(centerX - width/2))
                c2 = round(cmax*(centerX + width/2))
                f1 = round(fmax*(centerY - length/2))
                f2 = round(fmax*(centerY + length/2))
        else:
                c1 = round(fmax*(centerX - width/2))
                c2 = round(fmax*(centerX + width/2))
                f1 = round(cmax*(centerY - length/2))
                f2 = round(cmax*(centerY + length/2))
        ## Nos aseguramos de no salir fuera de la imagen:
        if c1 > cmax:
                c1 = cmax
        elif c1 < 0:
                c1 = 0
        if c2 > cmax:
                c2 = cmax
        elif c2 < 0:
                c2 = 0
        if f1 > fmax:
```

```
                f1 = fmax
        elif f1 < 0:
                f1 = 0
        if f2 > fmax:
                f2 = fmax
        elif f2 < 0:
                f2 = 0
        ## Submatriz con los datos:
        imgSubData = np.copy(imgData[f1:f2,c1:c2])
        ## Extraemos el rango máximo de filas y columnas:
        fmax=imgSubData.shape[0]
        cmax=imgSubData.shape[1]
        ## Dividimos el rectángulo en 3 regiones (descartando el tercio del medio):
        cc = round(cmax/3)
        widthROI = width/3
        minimo = int(csvf[k][9])
        maximo = int(csvf[k][10])
        for cuadrado in ["RL", "LL"]:
                if cuadrado == "RL":
                        ROI = np.copy(imgSubData[0:fmax,0:cc])
                        centerXROI = centerX - widthROI
                        centerYROI = centerY
                elif cuadrado == "LL":
                        ROI = np.copy(imgSubData[0:fmax,(2*cc):cmax])
                        centerXROI = centerX + widthROI
                        centerYROI = centerY
                # Normalizamos los valores de la matriz:
                newROI = (ROI - minimo) / (maximo - minimo)
                ## Escribimos el array con los datos para cada ROI:
                np.savetxt(outputArrayData + csvf[k][0] + "_" + cuadrado +
                 "_DICOM.txt", ROI)
#--------------------------
```

Apéndice B

Scripts R: Función `coarGrid.R`

La forma de usar esta función, una vez cargada en `R` con el comando `source`, sería: `coarGrid(imagen,q,d)`, donde: `imagen` denota la matriz de valores de gris, `q` es el tamaño de cuadrícula, y `d` el tamaño de vecindario.

```
#-------------------------------------------
calculaSnorm <- function(matriz,d){
        if((nrow(matriz)-2*d-1)<0 | (ncol(matriz)-2*d-1)<0) s <- NULL
        if((nrow(matriz)-2*d-1)>=0 & (ncol(matriz)-2*d-1)>=0){
                x <- unique(as.vector(matriz))
                s <- c()
                for(i in x){
                        li <- which(matriz==i, arr.ind=T)
                        li <- li[li[,1]>d & li[,1]<=(nrow(matriz)-d) & li[,2]>d &
                         li[,2]<=(ncol(matriz)-d), ]
                        if(is.null(nrow(li))) li <- as.data.frame(t(li))
                        ai <- c()
                        if(nrow(li)!=0){
                                for(k in 1:nrow(li)){
                                        aux <- matriz[(li[k,1]-d):(li[k,1]+d),
                                         (li[k,2]-d):(li[k,2]+d)]
                                        aux[(nrow(aux)/2)+0.5,
                                        (ncol(aux)/2)+0.5] <- NA
                                        ai <- c(ai, sum(aux,na.rm=T))
                                }
                        }
                        si <- abs(i-ai/(((2*d+1)^2)-1))
                        if(length(si)==0) s <- rbind(s, c(i,NA))
                        if(length(si)!=0) s <- rbind(s, c(i,sum(si, na.rm=T)))
                }
        }
        s[,2] <- s[,2]/sum(s[,2],na.rm=T)
        return(s)
}
#-------------------------------------------
coarsenessNorm <- function(matriz,d){
        matriz <- as.matrix(matriz)
        if(F%in%is.nan(matriz)){
```

```
        s <- calculaSnorm(matriz,d)
        if(is.null(s)) return("Error: d demasiado grande")
        if(!is.null(s)){
                s <- s[!is.na(s[,2]),]
                if(class(s)=="numeric" || (class(s)!="numeric" & nrow(s)<5)){
                        return(NA)
                }
                if(class(s)!="numeric" & nrow(s)>=5){
                        n <- (nrow(matriz)-2*d)*(ncol(matriz)-2*d)
                        matriz_aux <- matriz[(d+1):(nrow(matriz)-d),
                         (d+1):(ncol(matriz)-d)]
                        x <- unique(as.vector(matriz_aux))
                        x <- x[!is.nan(x)]
                        x <- x[order(x)]
                        p <- cbind(x, table(as.vector(matriz_aux))/n)
                        p <- p[p[,1]%in%s[,1],]
                        s <- s[order(s[,1]),]
                        p <- p[order(p[,1]),]
                        return(1/(1.e-20+sum(s[,2]*p[,2])))
                }
        }
}
if(F%in%is.nan(matriz)==F) return(NA)
}
#-------------------------------------------
coarGrid <- function(imagen, q, d){
if(d>=q) return("Error: q debe ser menor que d")
listaNoNans <- which(!is.na(imagen),arr.ind=T)
lung <- imagen[min(listaNoNans[,1]):max(listaNoNans[,1]),
 min(listaNoNans[,2]):max(listaNoNans[,2])]
if(q>min(c(nrow(lung), ncol(lung)))) return("Error: q demasiado grande")
li <- seq(1,nrow(lung)-q,q)
lj <- seq(1,ncol(lung)-q,q)
coars.aux <- c()
for(i in li){
        ifelse(i==li[length(li)] & (i+q-1)<nrow(lung), ki <- nrow(lung)-i, ki <- q-1)
        for(j in lj){
                ifelse(j==lj[length(lj)] & (j+q-1)<ncol(lung),
                 kj <- ncol(lung)-j, kj <- q-1)
                aux <- coarsenessNorm(lung[i:(i+ki),j:(j+kj)],d)
                coars.aux <- c(coars.aux, aux)
        }
}
coars.aux <- coars.aux[!is.na(coars.aux)]
return(IQR(coars.aux))
}
```

Bibliografía

[1] Amadasun M, King R (1989) Texural Features Corresponding to Texural Properties. IEEE Transactions on Systems, Man, and Cybernetics 19:1264-1274.

[2] Bhaskaran K, Bacon S, Evans SJW, Bates CJ, Rentsch CT, MacKenna B, Tomlinson L, Walker AJ, Schultze A, Morton CE, Grint D, Mehrkar A, Eggo RM, Inglesby P, Douglas IJ, McDonald HI, Cockburn J, Williamson EJ, Evans D, Curtis HJ, Hulme WJ, Parry J, Hester F, Harper S, Spiegelhalter D, Smeeth L, Goldacre B (2021) Factors associated with deaths due to COVID-19 versus other causes: population-based cohort analysis of UK primary care data and linked national death registrations within the OpenSAFELY platform. The Lancet Regional Health - Europe 6.

[3] Cavallo AU, Troisi J, Forcina M, Mari PV, Forte V, Sperandio M, Pagano S, Cavallo P, Floris R, Garaci F (2021) Texture Analysis in the Evaluation of Covid-19 Pneumonia in Chest X-Ray Images: A Proof of Concept Study. Current Medical Imaging 17.

[4] Deepa VN, Krishna N, Kumar HG (2018) Feature Extraction and Classification of X-Ray Lung Images Using Haralick Texture Features. En: Bhattacharyya P, Sastry H, Marriboyina V, Sharma R (ed) Smart and Innovative Trends in Next Generation Computing Technologies. Springer, Singapore, pp 899-907.

[5] Delrue L, Gosselin R, Ilsen B, Van Landeghem A, de Mey J, Duyck P (2011) Difficulties in the Interpretation of Chest Radiography. En: Coche E, Ghaye B, de Mey J, Duyck P (ed) Comparative Interpretation of CT and Standard Radiography of the Chest. Springer, Berlin, Heidelberg, pp 27-49.

[6] Durbán M (2008) Splines con penalizaciones (P-splines): teoría y aplicaciones. En: María Dolores Ugarte (ed). Universidad Pública de Navarra.

[7] Elezkurtaj S, Greuel S, Ihlow J, Michaelis EG, Bischoff P, Kunze CA, Sinn BV, Gerhold M, Hauptmann K, Ingold?Heppner B, Miller F, Herbst H, Corman VM, Martin H, Radbruch H, Heppner FL, Horst D (2021) Causes of death and comorbidities in hospitalized patients with COVID-19. Scientific Reports 11:4263.

[8] Eze P, Parampalli U, Evans R, Liu D (2020) Evaluation of the Effect of Steganography on Medical Image Classification Accuracy. Journal of Applied Bioinformatics & Computational Biology 9:4.

[9] Franco L, Tahoces PG, Martínez-Mera JA (2013) Visualization software for CT: fan/cone beam and metrology applications. Procedia Engineering 63:779-785.

[10] González R, Woods RE, Eddins SL (2003) Digital Image Processing Using MATLAB. Prentice-Hall, Inc, USA.

[11] Gude-Sampedro F, Fernández-Merino C, Ferreiro L, Lado-Baleato O, Espasandín-Domínguez J, Hervada X, Cadarso CM, Valdés L (2020) Development and validation of a prognostic model based on comorbidities to predict Covid-19 severity. A population-based study. International Journal of Epidemiology 50:64-74.

[12] Hakim B, Basari (2019) Tuberculosis detection analysis using texture features on CXRs image. AIP Conference Proceedings 2092.

[13] Haralick RM, Shanmugam K, Dinstein I (1973) Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics 6:610-621.

[14] Harrell FE Jr (2015) Regression Modeling Strategies With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. Springer, New York.

[15] Harrell FE Jr (2021) rms: Regression Modeling Strategies. R package version 6.2-0. https://CRAN.R-project.org/package=rms. Accedido 09 de junio de 2021.

[16] Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X, Cheng Z, Yu T, Xia J, Wei Y, Wu W, Xie X, Yin W, Li H, Liu M, Xiao Y, Gao H, Guo L, Xie J, Wang G, Jiang R, Gao Z, Jin Q, Wang J, Cao B (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The Lancet 395:497-506.

[17] Hussain L, Nguyen T, Li H, Abbasi AA, Lone KJ, Zhao Z, Zaib M, Chen A, Duong TQ (2020) Machine-learning classification of texture features of portable chest X-ray accurately classifies COVID-19 lung infection. BioMedical Engineering OnLine 19:88.

[18] Karmakar P, Teng SW, Zhang D, Liu Y, Lu G (2017) Improved Tamura Features for Image Classification Using Kernel Based Descriptors. International Conference on Digital Image Computing: Techniques and Applications (DICTA) 1-7.

[19] Katsuragawa S, Doi K, MacMahon H (1988) Image feature analysis and computer-aided diagnosis in digital radiography: Detection and characterization of interstitial lung disease in digital chest radiographs. Medical Physics 15:311.

[20] Khuzi AM, Besar R, Wan Zaki WMD (2008) Texture Features Selection for Masses Detection In Digital Mammogram. En: Abu Osman NA, Ibrahim F, Wan Abas WAB, Abdul Rahman HS, Ting HN (ed) 4th Kuala Lumpur International Conference on Biomedical Engineering. Springer, Berlin, Heidelberg, pp 629-632.

[21] Kimpe T, Tuytschaever T (2007) Increasing the Number of Gray Shades in Medical Display Systems?How Much is Enough? Journal of Digital Imaging 20:422-432.

[22] Kociolek M, Strzelecki M, Szymajda S (2018) On the influence of the image normalization scheme on texture classification accuracy. Signal Processing: Algorithms, Architectures, Arrangements, and Applications 152-157.

[23] Lee D, Choi S, Lee H, Kim D, Choi S, Kim HJ (2017) Quantitative evaluation of anatomical noise in chest digital tomosynthesis, digital radiography, and computed tomography. Journal of Instrumentation 12:T04006-T04006.

[24] Levine MD (1985) Vision in man and machine. McGraw-Hill College, USA.

[25] Long JD, Mills JA (2018) Joint modeling of multivariate longitudinal data and survival data in several observational studies of Huntington?s disease. BMC Medical Research Methodology 18:138.

[26] Lu R, Zhao X, Li J, Niu P, Yang B, Wu H, Wang W, Song H, Huang B, Zhu N, Bi Y, Ma X, Zhan F, Wang L, Hu T, Zhou H, Hu Z, Zhou W, Zhao L, Chen J, Meng Y, Wang J, Lin Y, Yuan J, Xie Z, Ma J, Liu WJ, Wang D, Xu W, Holmes EC, Gao GF, Wu G, Chen W, Shi W, Tan W (2020) Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. The Lancet 395:565-574.

[27] Mildenberger P, Eichelberg M, Martin E (2002) Introduction to the DICOM standard. European Radiology 12:920-927.

[28] Miller ME, Hui SL, Tierney WM (1991) Validation techniques for logistic regression models. Statistics in Medicine 10:1213-1226.

[29] Pawley JB (2006) Points, Pixels, and Gray Levels: Digitizing Image Data. Handbook of Biological Confocal Microscopy. Springer Science+Business Media, LLC, New York.

[30] Remuzzi A, Remuzzi G (2020) COVID-19 and Italy: what next? The Lancet 395:1225-1228.

[31] Rizopoulos D (2012). Joint Models for Longitudinal and Time-to-Event data with Applications in R. CRC Press, Boca Raton.

[32] Roustaei N, Ayatollahi SMT, Zare N (2018) A Proposed Approach for Joint Modeling of the Longitudinal and Time-To-Event Data in Heterogeneous Populations: An Application to HIV/AIDS?s Disease. BioMed Research International.

[33] Sheikh T, Ibrahim JG, Gelfond JA, Sun W, Chen MH (2020) Joint modelling of longitudinal and survival data in the presence of competing risks with applications to prostate cancer data. Statistical Modelling 21:72-94.

[34] Shephard CT (2003) Radiographic image production and manipulation. McGraw-Hill Columbus, Ohio.

[35] Sivaramakrishnan R, Antani S, Candemir S, Xue Z, Abuya J, Kohli M, Alderson P, Thoma G (2018) Comparing deep learning models for population screening using chest radiography. Medical Imaging 2018: Computer-Aided Diagnosis.

[36] Sprawls P Jr (1993) Physical Principles of Medical Imaging. Aspen Pub, Inc, USA.

[37] Sun X, Shi L, Luo Y, Yang W, Li H, Liang P, Li K, Mok VCT, Chu WCW, Wang D (2015) Histogram-based normalization technique on human brain magnetic resonance images from different acquisitions. BioMedical Engineering OnLine 14:73.

[38] Suresh S, Mohan S (2020) ROI-based feature learning for efficient true positive prediction using convolutional neural network for lung cancer diagnosis. Neural Computing and Applications 32:15989-16009.

[39] Tabik S, Gómez-Ríos A, Martín-Rodríguez JL, Sevillano-García I, Rey-Area M, Charte D, Guirado E, Suárez JL, Luengo J, Valero-González MA, García-Villanova P, Olmedo-Sánchez E, Herrera F (2020) COVIDGR Dataset and COVID-SDNet Methodology for Predicting COVID-19 Based on Chest X-Ray Images. IEEE Journal of Biomedical and Health Informatics 24:3595-3605.

[40] Thepade SD, Bang SV, Chaudhari PR, Dindorkar MR (2020) Covid19 Identification from Chest X-ray Images using Machine Learning Classifiers with GLCM Features. Electronic Letters on Computer Vision and Image Analysis 19:85-97.

[41] van Ginneken B, Katsuragawa S, ter Haar Romeny BM, Doi K, Viergever MA (2002) Automatic Detection of Abnormalities in Chest Radiographs Using Local Texture Analysis. IEEE Transactions on Medical Imaging 21:139-149.

[42] Varela C, Tahoces PG, Méndez AJ, Souto M, Vidal JJ (2007) Computerized detection of breast masses in digitized mammograms. Academic Radiology 2:959-966.

[43] Wang S, Zha Y, Li W, Wu Q, Li X, Niu M, Wang M, Qiu X, Li H, Yu H, Gong W, Bai Y, Li L, Zhu Y, Wang L, Tian J (2020) A Fully Automatic Deep Learning System for COVID-19 Diagnostic and Prognostic Analysis. European Respiratory Journal 57.

[44] Weszka JS, Dyer CR, Rosenfeld A (1976) A Comparative Study of Texture Measures for Terrain Classification. IEEE Transactions on Systems, Man, and Cybernetics 6:269-285.

[45] Whitcher B, Schmid VJ, Thornton A (2011) Working with the DICOM and NIfTI Data Standards in R. Journal of Statistical Software 44:1-28.

[46] Wong HYF, Lam HYS, Fong AHT, Leung ST, Chin TWY, Lo CSY, Lui MMS, Lee JCY, Chiu KWH, Chung TWH, Lee EYP, Wan EYF, Hung IFN, Lam TPW, Kuo MD, Ng MY (2020) Frequency and Distribution of Chest Radiographic Findings in Patients Positive for COVID-19. Radiology 296:E72-E78.

[47] Xu M, Qi S, Yue Y, Teng Y, Xu L, Yao Y, Qian W (2019) Segmentation of lung parenchyma in CT images using CNN trained with the clustering algorithm generated dataset. BioMedical Engineering OnLine 18.

[48] Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G (2006) User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. NeuroImage 31:1116-1128.

[49] Zhang C, Yang G, Cai C, Xu Z, Wu H, Guo Y, Xie Z, Shi H, Cheng G, Wang J (2020) Development of a quantitative segmentation model to assess the effect of comorbidity on patients with COVID?19. European Journal of Medical Research 25.