# Proyecto Final de Bioingeniería

*Título:* Sistema de soporte a la toma de decisiones para la detección de opacidades pulmonares en radiografías de tórax mediante el uso de redes neuronales convolucionales

*Alumna:* Eugenia BERRINO - 57.202

*Tutora:* Candelaria MOSQUERA

*Entrega:* Febrero de 2022

# Índice

# Índice de figuras

# Índice de tablas

# Siglas

**AIC** *Akaike's information criterion*), criterio de información de Akaike.

**ANN** *artificial neural network.*

**AP** anteroposterior.

**AUC** *area under the curve*, área debajo de la curva.

**BB** *bounding boxes.*

**CNN** redes neuronales convolucionales.

**DICOM** *Digital Imaging and Communication On Medicine.*

**DL** *deep learning.*

**DS** *dataset.*

**FN** falsos negativos.

**FP** falsos positivos.

**GT** *ground truth.*

**HIBA** Hospital Italiano de Buenos Aires.

**IA** inteligencia artificial.

**IoU** *intersection over union.*

**L** lateral.

**mAP** *mean average precision.*

**ML** *machine learning.*

**PA** posteroanterior.

**pIASHIBA** Programa de Inteligencia Artificial en Salud del Hospital Italiano de Buenos Aires.

**ReLU** *rectified linear unit.*

**RMSProp** *root mean square propagation.*

**ROC-AUC** *receiver operating characteristic curve* - área debajo de la curva).

**RxTx** radiografía de tórax.

**SGD** *stochastic gradient descent.*

**SSD** sistemas de soporte a la toma de decisiones.

**TL** *transfer learning.*

**VN** verdaderos negativos.

**VP** verdaderos positivos.

# 1. Resumen

La radiografía de tórax es una técnica diagnóstica ampliamente utilizada en todo el mundo debido a que permite obtener representaciones confiables del cuerpo de los pacientes de manera no invasiva, a tiempos cortos, sin la necesidad de preparaciones especiales, con riesgos residuales aceptables y a un costo significativamente menor que otros estudios de imagen. Sin embargo, la interpretación de este estudio de imagen es compleja y presenta una gran variabilidad interobservador.

Los profesionales de salud capacitados para la interpretación de la radiografía son los médicos especialistas en imágenes, que se encargan de preparar el informe del estudio, para que otros profesionales de la salud puedan utilizarlos para la atención.

Dada la creciente demanda mundial en los servicios de diagnóstico por imágenes, muchas veces no es posible informar todas las radiografías en tiempo, aumentando el volumen de radiografías sin informe de especialista. En estos casos, las interpretaciones deban ser realizadas por médicos de otras especialidades, menos capacitados para la tarea.

Tomando como base el teorema fundamental de la informática biomédica enunciado por Charles Friedman, el cual postula que el trabajo de un profesional en conjunto con una fuente de información es mejor que el mismo profesional trabajando sin ella, se desarrolló un sistema de soporte a la toma de decisiones médicas para la detección de opacidades pulmonares en radiografías de tórax basado en redes neuronales convolucionales. El sistema es capaz de filtrar las imágenes inválidas previamente para garantizar predicciones sobre el tipo de imagen correcta, detectar presencia o ausencia de opacidades pulmonares de manera binaria y, en caso de hallar opacidades, mostrar en la imagen su ubicación. El sistema fue diseñado para ser implementado en la central de emergencias para asistir a médicos no especialistas en imágenes del Hospital Italiano de Buenos Aires (HIBA) y por tanto, se utilizaron imágenes propias del hospital para validar el sistema. Estas últimas pruebas, demostraron la capacidad del mismo para clasificar correctamente el 95 % de las imágenes de un total de 1284 casos.

# 2. Introducción

## 2.1. Contexto

Este proyecto se encuentra enmarcado en el Programa de inteligencia artificial (IA) en Salud (pIASHIBA) del Departamento de Informática en Salud del Hospital Italiano de Buenos Aires (DIS-HIBA) y fue realizado en conjunto con este Departamento y el de Diagnóstico por Imágenes del hospital.

El Hospital Italiano de Buenos Aires (HIBA) es un centro de salud con internación general de notoria trayectoria en la mejora de la salud de pacientes con diferentes niveles de complejidad de atención. Desde pacientes de alto riesgo con terapia intensiva especializada, hasta ambulatorios en el sector de consultorios. A su vez, cuenta con un sector de guardia y un departamento dedicado exclusivamente a imágenes médicas con una gran afluencia de gente. Una problemática actual del servicio de Diagnóstico por Imágenes es que la cantidad de estudios solicitados posee una tendencia creciente haciendo cada vez más complejo el procesamiento en tiempo y forma de cada uno de estos estudios. Por otra parte, si bien el servicio obtiene imágenes bajo diferentes modalidades, el mayor porcentaje se encuentra representado por radiografías y de la totalidad de este tipo de estudios, el 50 % son de tórax. Actualmente, en promedio, se realizan en HIBA 150.000 radiografías de tórax al año, lo que equivale a 400 radiografías por día.

Para poder almacenar esta cantidad de imágenes de manera diaria, el HIBA cuenta con sistemas de información en salud que permiten el guardado de archivos digitales en un formato específicamente diseñado para tal fin conocido como *Digital Imaging and Communication On Medicine* (DICOM), el cual no solo almacena la imagen en sí, sino que también los metadatos necesarios para realizar su interpretación. Sin embargo, en algunos equipos de rayos X el registro de estos metadatos depende del técnico radiólogo y esto hace que dichos estudios presenten errores en ciertos campos de metadatos. Uno de estos campos es la región anatómica de la que se obtiene el estudio y otro la proyección de la imagen. Esto sumado a que las imágenes individuales se encuentran agrupadas en series y que estos metadatos se encuentran registrados a este nivel de granularidad, trae como consecuencia que existan errores o datos faltantes en los campos de metadatos.

El pIASHIBA fue creado en el 2018 con el objetivo de investigar las posibles aplicaciones de inteligencia artificial en salud, desarrollar herramientas de utilidad clínica e implementarlas como parte del flujo clínico para mejorar la calidad de atención. Dentro del marco de este mismo programa y también en conjunto con el Departamento de Diagnóstico por imágenes del hospital, se encuentra en desarrollo desde ese

mismo año T-Rx, un sistema capaz de detectar hallazgos patológicos en radiografías de tórax. Si bien T-Rx ha demostrado ser de gran utilidad para la detección de los hallazgos que se propone, una de sus principales desventajas es que selecciona las imágenes a procesar mediante los metadatos presentes en los estudios.

## 2.2. Planteo del Problema

La radiografía de tórax es una técnica diagnóstica ampliamente utilizada en todo el mundo debido a que permite obtener representaciones confiables del cuerpo de los pacientes de manera no invasiva, a tiempos cortos, sin la necesidad de preparaciones especiales, con riesgos residuales aceptables y a un costo significativamente menor que otros estudios de imagen.

Sin embargo, la interpretación de dichas radiografías no es una tarea sencilla. Esto se debe principalmente a la gran cantidad de estructuras anatómicas que se localizan en el tórax y a la consecuente superposición de las mismas en la placa. [1]. Los profesionales de salud capacitados en interpretarlas y por tanto los encargados de escribir los informes correspondientes, son médicos especialistas en imágenes. Esta modalidad de imagen no resulta atractiva y no es explorada por las nuevas generaciones de médicos especialistas en diagnóstico por imágenes, que prefieren orientarse hacia métodos modernos de mayor complejidad y mejor rédito económico, como la tomografía computada o la resonancia magnética. Por lo tanto, la problemática del volumen cada vez mayor de radiografías simples a analizar es intensificado por la escasez de médicos especialistas con dedicación y experiencia en su interpretación. Esto ocasiona que en muchos casos las interpretaciones deban ser realizadas por médicos de otras especialidades, menos capacitados para esta tarea. [2] [3]

Para realizar un análisis automático de estas imágenes se debe comenzar con la identificación de la región anatómica y proyección de la misma. Estas descripciones, cargadas manualmente por técnicos radiólogos, son almacenadas como metadatos de la imagen en la cabecera del archivo DICOM. Los principales problemas de este proceso son que alrededor del 5 % de las imágenes que son parte de un estudio de radiografías de tórax no posee descripción y que en alrededor del 27 % de los estudios de radiografías de tórax se incluyen también radiografías de otras regiones anatómicas y se cataloga a todo el conjunto como “Radiografía de Tórax”. En consecuencia, muchas veces los metadatos de las imágenes no son confiables, lo que trae dificultades cuando se desea integrar sistemas automáticos de asistencia al diagnóstico o de *triage*, que recurren a los metadatos de la imágenes para filtrarlas y seleccionarlas, ya que estos no son confiables.

Para resolver el problema anterior, se desarrolló un sistema basado en redes neurona-

les convolucionales con la capacidad de determinar características de una radiografía automáticamente a partir de la imagen, sin recurrir a sus correspondientes metadatos. En primer lugar, se buscó discriminar si una radiografía es de tórax o de otra región anatómica, y en segundo lugar, clasificar la proyección de una radiografía de tórax en posteroanterior (PA), anteroposterior (AP) o lateral (L).

Las opacidades pulmonares son el tipo de hallazgos más prevalente en radiografías de tórax, debido a que identifican la ausencia de oxígeno en una región pulmonar, característica que se relaciona con muchos posibles diagnósticos. Como consecuencia, este tipo de hallazgo es el más prevalente en radiografías de tórax y su correcta identificación tiene la capacidad de mejorar considerablemente el servicio de atención médica brindado.

Como solución y tomando como base el teorema fundamental de la informática biomédica enunciado por Charles Friedman [4], se desarrolló un sistema de soporte a la toma de decisiones médicas para la detección de opacidades pulmonares en radiografías de tórax. El mismo es capaz de detectar presencia o ausencia de opacidades pulmonares de manera binaria y, de poseer opacidades, de mostrar en la imagen su ubicación.

El sistema final propuesto, se presenta en la Fig. 1. La sección de filtrado se encuentra compuesta con un filtro bietápico. La primer etapa recibe como entrada una radiografía de cualquier región anatómica, y es capaz de clasificar las imágenes en tórax u otros con un *F1-Score* de 0.93. La segunda etapa recibe las imágenes clasificadas como tórax y las clasifica en PA, AP y L con un *F1-score* de 0.83. Luego, las imágenes clasificadas como PA, son introducidas en un algoritmo de detección de opacidades pulmonares. El mismo detecta la presencia de estas opacidades con un *mean average precision* (mAP) de 0.31. Finalmente, los datos de salida del detector de opacidades, en particular, la cantidad de detecciones y la máxima confianza asociada a alguna de dichas detecciones, son dadas como entrada a un modelo de regresión logística, encargado de binarizar las detecciones. Esta etapa posee un *area under the curve*, área debajo de la curva (AUC) de 0.98 y de *precision - recall* de 0.92. Esta salida binaria le permite al sistema funcionar como un soporte a la toma de decisiones. Las métricas obtenidas en la etapa de filtrado se obtuvieron a partir de un conjunto de 106 radiografías del hospital, mientras que las de detecciones de opacidades y binarización se obtuvieron a partir de 1248 imágenes disponibilizadas por HIBA.

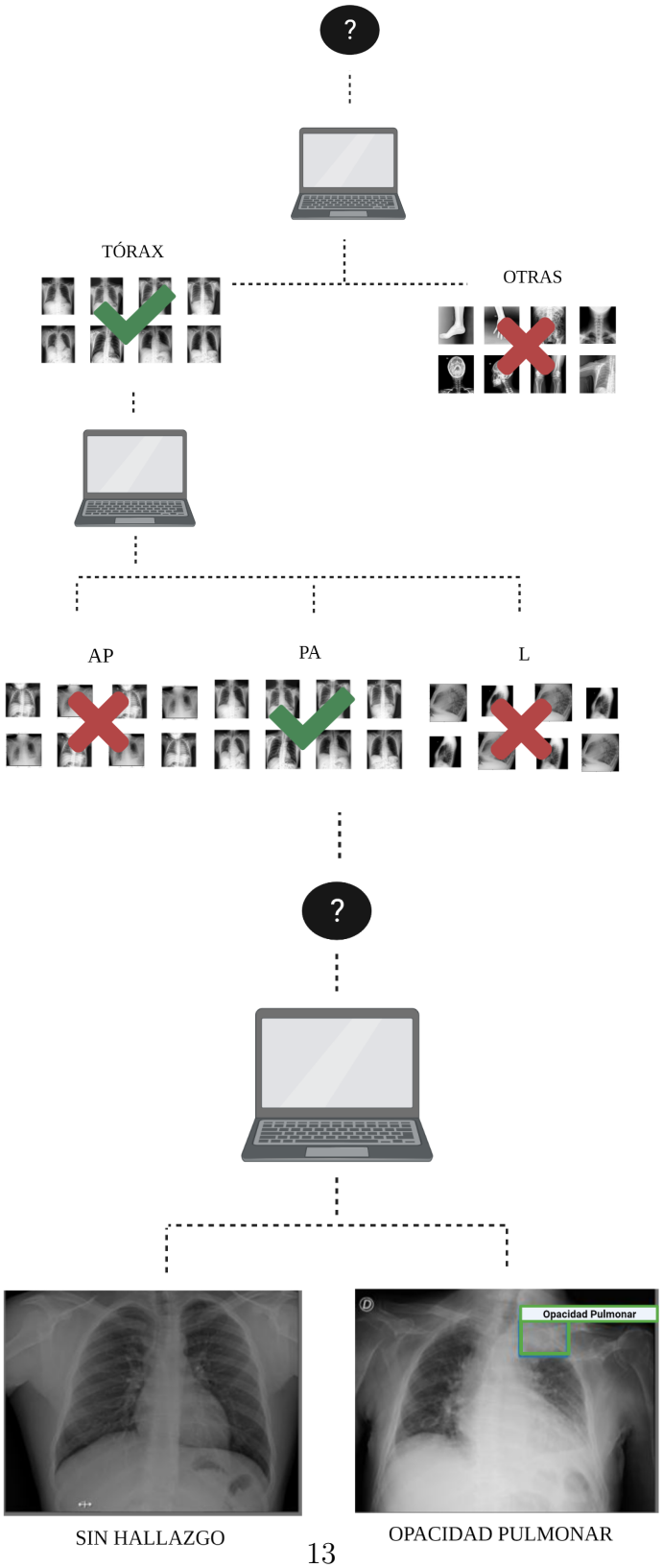




Figura 1: SSD para la Detección de Opacidades Pulmonares

Una importante ventaja del sistema es que el mismo se encuentra diseñado de forma modular. Es decir, existe la posibilidad de implementar diferentes partes por separado, lo que permitiría su uso en diferentes centros de salud, con diferentes necesidades. Además, el sistema de filtrado puede resultar de gran utilidad para otros algoritmos previamente implementados, como por ejemplo T-Rx, los cuales ven disminuida su efectividad por la presencia de imágenes de otras regiones corporales u otras proyecciones. Una última propuesta de aplicación para el sistema de detección de opacidades pulmonares con su posterior regresión logística, es el armado de un sistema de priorización de atenciones, o sistema de *triage*. Sin embargo, para la implementación del mismo debería ponerse especial atención en la forma de mitigar posibles falsos negativos y dicha descripción se encuentra por fuera de los alcances de este proyecto.

El resto del trabajo se encuentra estructurado en tres secciones: Marco Teórico, Desarrollo y Conclusión. En el marco teórico se realiza una breve introducción de radiología, opacidades pulmonares e inteligencia artificial para aplicaciones médicas. Luego, en la sección de desarrollo se presentan tanto los materiales y métodos así como los resultados para cada uno de los experimentos que guiaron la elección de arquitecturas e hiperparámetros y que dieron forma al sistema finalmente propuesto. En las conclusiones se resumen las decisiones tomadas como consecuencia de cada experimento, se plantean los puntos fuertes del trabajo así como también las limitaciones y perspectivas a futuro. Finalmente, existe para aquel lector interesado una sección de anexos con información más detallada sobre los materiales y métodos empleados.

# 3. Marco Teórico

## 3.1. Radiología

La obtención radiografías se produce como consecuencia de la emisión de un haz colimado de rayos X sobre el paciente. Los rayos X son producidos en un tubo de rayos mediante un fenómeno conocido como radiación de frenado. Estos rayos tienen la capacidad de penetrar tejidos e interactuar con ellos. Como consecuencia, parte de los rayos son desviados y parte absorbidos. La verdadera razón por la que puede generarse una imagen es que diferentes tejidos interactúan de diferente manera. Están aquellos como el tejido óseo que absorbe fuertemente la radiación y por ende se ve en la imagen muy radiopaco (blanco) y aquellos como el tejido pulmonar, prácticamente compuesto por aire, que casi no interactúa con este tipo de radiación y como consecuencia se considera que es muy radiolúcido (negro). Sin embargo, la formación de la imagen final depende también de otro elemento clave, el detector, encargado de medir los rayos X una vez atravesado el cuerpo del paciente. A modo de resumen y para facilitar la comprensión, se presenta en la Fig. 2 una imagen que ilustra el procedimiento de adquisición de imágenes utilizando esta técnica.

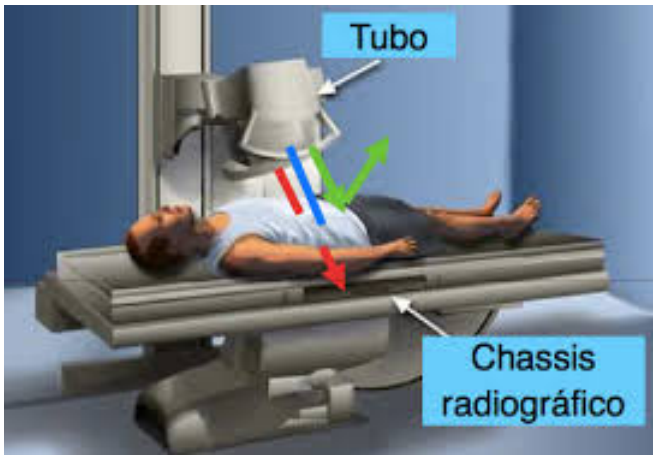


Figura 2: Obtención de Radiografías. Imagen tomada de [5]

### 3.1.1. Radiografía de Tórax

La radiografía de tórax (RxTx) es uno de los métodos de obtención de imágenes más utilizados debido a su capacidad diagnóstica sobre una variedad de órganos y estructuras corporales a las que se tiene acceso. Dada su posición anatómica y las características de este tipo de adquisición, es posible observar las siguientes estructuras: el corazón, los pulmones, los vasos sanguíneos, el árbol bronquial, vértebras y

caja torácica. Como consecuencia, existe una gran cantidad de hallazgos que pueden ser detectados. Más aún en los casos de hallazgos múltiples, existe una gran cantidad de variabilidad entre estos estudios. [6] [1]

Una clasificación muy usual que suele realizarse en las radiografías de tórax es utilizar como criterio la proyección o vista de la misma. Existen tres tipos principales: posteroanterior (PA), anteroposterior (AP) y lateral (L). Tanto la PA como la AP son consideradas radiografías frontales, y se diferencian en la posición relativa del paciente, la fuente de rayos X y el detector. En las radiografías de tórax AP, la fuente de rayos X se posiciona de frente al paciente, mientras que en las PA, dicha posición la ocupa el detector. Si bien ambos tipos de radiografías pueden obtenerse con el paciente de pie, generalmente las PA se obtienen de dicha manera, mientras que en las AP el paciente se coloca en posición supina (Ver Fig. 3). Las radiografías L son aquellas que se obtienen con el paciente de perfil tanto a la fuente como al detector de rayos.[7]

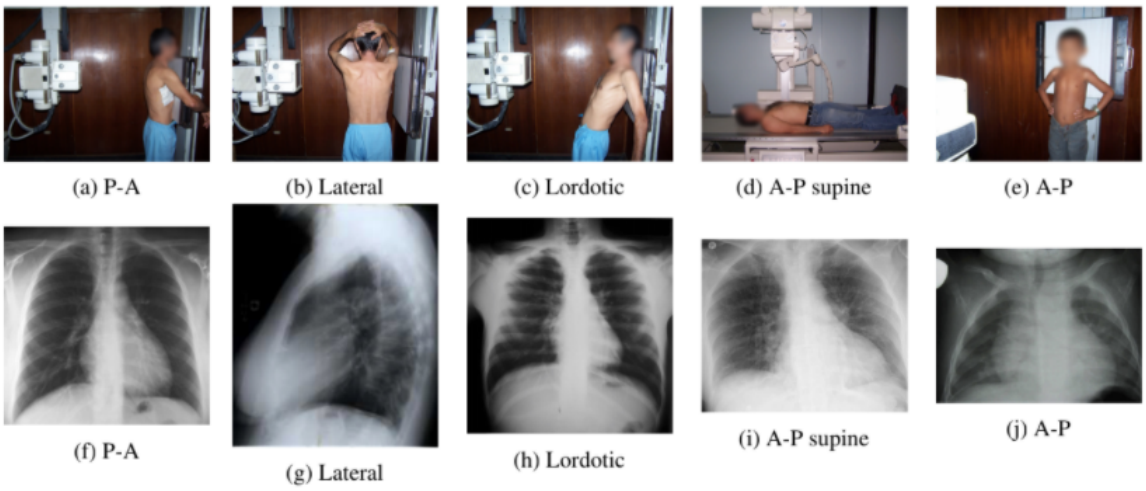


Figura 3: VISTAS RADIOGRÁFICAS CONTENIDAS EN PADCHEST. IMAGEN TOMADA DE [8]

Al momento de analizar una radiografía de tórax, el primer paso que realiza un profesional de salud capacitado es la identificación del tipo de imagen y de proyección. Una vez determinados dichos parámetros es que se propone a observar una a una las diferentes estructuras corporales y finalmente, de considerar la presencia de hallazgos radiográficos, a clasificarlos. [1]

### 3.1.2. Almacenamiento y Transmisión de Imágenes Médicas

Actualmente, en HIBA, las imágenes médicas son almacenadas y transmitidas gracias al sistema PACS (Picture Archive and Comunication System). El mismo se encuentra basado en el Protocolo DICOM (Digital Image Comunication on Medicine).

Para DICOM, toda la información del mundo real, es un objeto con sus correspondientes propiedades o atributos. Estos objetos se encuentran relacionados entre sí según la entidad que representan. En primer lugar, codifican como objetos los pacientes. En segundo lugar, se definen como objetos los estudios. En tercer lugar, se encuentran los objetos que componen a los estudios, que se conocen como series de imágenes. Las series de imágenes agrupan en teoría todas aquellas imágenes obtenidas en un estudio adquiridas bajo una misma modalidad, de una región corporal y de una orientación anatómica corporal o proyección. Y por último se encuentran las imágenes individuales dentro de una serie. En la Fig. 4 se presenta un resumen del modelo de DICOM.

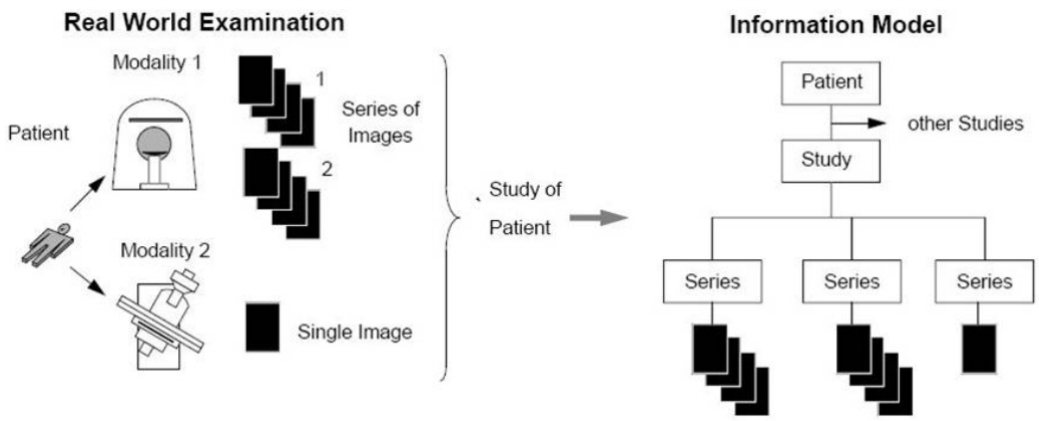


Figura 4: MODELO DE INFORMACIÓN DICOM. IMAGEN TOMADA DE [9]

### 3.1.3. Opacidades Pulmonares

Entendemos por Opacidades Pulmonares a "Cualquier opacidad u opacidades focales o generalizadas anormales en los campos pulmonares (definición generalizada que incluye pero no se encuentra limitada a consolidación, cavidad, fibrosis, nódulo, masa, calcificación, engrosamiento intersticial, etc.)."[10]

Más coloquialmente, se trata de un término muy general que agrupa todos aquellos hallazgos radiográficos en los que algún pulmón o región de este se encuentre más radiopaco que otro, como consecuencia de la presencia de alguna sustancia capaz de

interactuar en mayor medida con los rayos X que un tejido pulmonar normal. Esta sustancia podría ser algún fluido biológico como por ejemplo sangre, agua, bacterias o incluso células tumorales.

Se pueden clasificar como opacidades pulmonares los siguientes tipos de hallazgos:

- **Atelectasia:** "Colapso de una parte del pulmón debido a un decremento en la cantidad de aire en los alveolos, resultando en una disminución del volumen pulmonar e incremento en su densidad."
- **Consolidación:** "Cualquier proceso patológico que ocasiona la presencia de fluidos, pus, sangre, células (incluidas las tumorales) u otras sustancias en los alveolos, y que dan lugar a opacidades mal definidas ya sean lobares, difusas o multifocales."
- **Nódulo/Masa:** "Cualquier espacio que ocupe una lesión, ya sea unitaria o múltiple."

## 3.2. Inteligencia Artificial

Tal como lo muestra la Fig. 5, la inteligencia artificial (IA) es un concepto amplio, considerablemente abarcativo y que en su definición incluye en un primer nivel a *machine learning* (ML), y este último incluye a otro que se conoce como *deep learning* (DL).

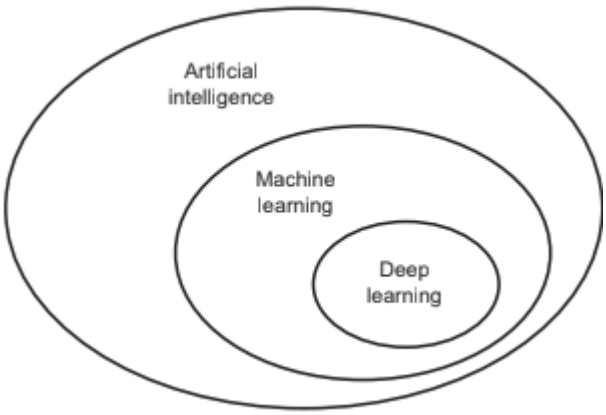


Figura 5: INTELIGENCIA ARTIFICIAL, MACHINE LEARNING Y DEEP LEARNING. IMAGEN TOMADA DE [11]

El concepto de IA surgió en los años 1950s, junto con el interrogante de si las máquinas podrían ser capaces de realizar tareas intelectuales, que hasta el momento se

encontraban reservadas exclusivamente para humanos. Se dice que esta definición es amplia y abarcativa porque puede incluir por ejemplo, la programación de una cantidad de secuencias de pasos lógicos que permitan resolver una tarea de tipo "intelectual".[12]

## 3.3. *Machine Learning*

El concepto ML, se encuentra incluido dentro de IA, pero resulta mucho más específico, ya que pretende responder el interrogante de si es posible que una máquina resuelva una tarea sin la necesidad de programarla para ello, es decir, sin la indicación explícita de todos los pasos lógicos que debe seguir. Este concepto plantea un nuevo paradigma de programación, el cual se ilustra en la Fig. 6.

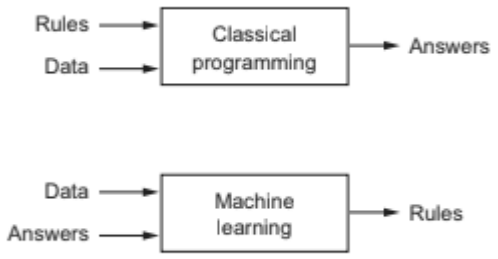


Figura 6: Diferencias el paradigma de programación clásica y el propuesto por ML. imagen tomada de [11]

Observando la figura, se puede concluir que en el caso del paradigma de ML, se requieren como entradas al modelo datos y respuestas, generalmente en grandes cantidades, y a partir de las mismas, es posible que el sistema deduzca las reglas para la solución de un problema en particular. Es debido a esta diferencia fundamental que se dice que los modelos de ML no se programan explícitamente, sino que se entrenan.

Ahora bien, la forma en la que son presentados dichos resultados permite clasificar los modelos de ML según su tipo de aprendizaje: supervisado, no supervisado, semi-supervisado y aprendizaje por refuerzo. En el primer caso, que es el adoptado a lo largo de todo el presente proyecto, los resultados se presentan de manera explícita, tal cual como se pretende que sea la salida del modelo, mientras que en el aprendizaje no supervisado se obtiene algún resultado que permite inferir, sin tener datos directos, sobre la tarea realizada.

El último punto que es necesario tener en cuenta respecto al paradigma de ML, es que al tener como entrada datos y sus respuestas, es decir, ejemplos de la tarea

a resolver, muchas veces estos datos componen tan solo una muestra del universo de casos que se pretenden que el algoritmo resuelva. Como consecuencia, en ML se suelen utilizar técnicas estadísticas para buscar patrones en los datos presentados y realizar inferencias a partir de ellos. Sin embargo, para que estos algoritmos alcancen un desempeño aceptable, muchas veces se requieren una gran cantidad de datos de entrada, y una potencia computacional acorde. Además, debe ponerse especial cuidado que la muestra utilizada para el entrenamiento sea representativa del universo en el cual se pretende que el sistema sea capaz de resolver su tarea. [13]

Para poder comprender en mayor profundidad cómo funciona un modelo de aprendizaje supervisado en ML, resulta útil observar la Fig. 7, que se presenta a continuación.

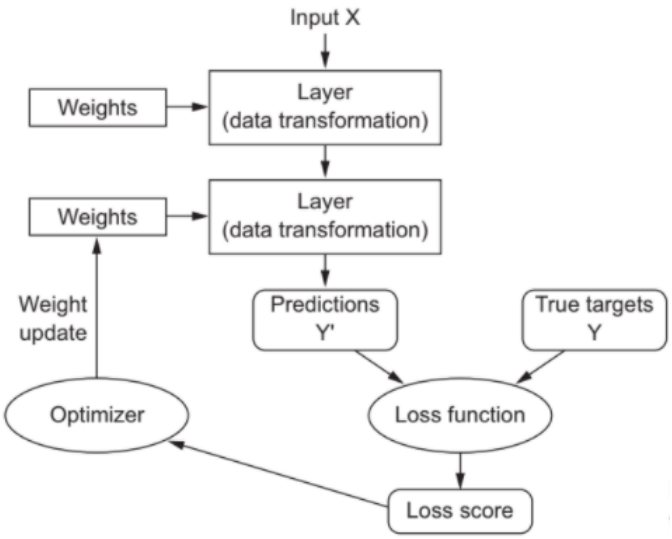


Figura 7: Representación de un modelo de aprendizaje supervisado de ML. imagen tomada de [11]

En la Fig. 7 es posible observar en primer lugar el *input* 'X' que son los datos que se le a presentan al modelo. Luego, estos datos sufren transformaciones parametrizadas por ciertos pesos o coeficientes, llamados parámetros.

En un comienzo, dichos parámetros se definen de manera aleatoria y el modelo no tiene la capacidad de resolver correctamente la tarea para la que fue creado. Allí es donde entra el concepto de aprendizaje: se toman las predicciones a la salida del modelo, y se comparan con las definidas como *ground truth* (GT) a través de la función de costo. De esta función de obtiene un *score* o métrica que es dada como entrada al optimizador. El optimizador, busca minimizar esta función de costo con

el objetivo de que los resultados del modelo sean lo más similares a su GT posible. Debido a que la función de costo suele ser compleja, se utilizan métodos numéricos para su optimización los cuales son de naturaleza iterativa. Como *output* de cada una de las iteraciones, se ajustan los pesos del modelo, y se calcula nuevamente la función de costo. Luego de varias iteraciones, de estar correctamente parametrizado el optimizador, se converge a un mínimo local de la función de costo.[12].

## 3.4. Deep Learning

La diferencia clave entre ML y DL es que *deep learning* realiza representaciones de los datos en una gran cantidad de capas sucesivas, mientras que ML se enfoca en representar la información en una o dos capas como mucho (ver Fig. 7). [12].

## 3.5. Redes Neuronales

Y finalmente, dentro del DL, existen modelos de procesamiento de los datos que se conocen como *artificial neural network* (ANN), o simplemente redes neuronales. Para comprender el funcionamiento de la ANN, conviene comenzar por el concepto de perceptrón.

### 3.5.1. Perceptrones

El funcionamiento de un perceptrón se puede observar en la Fig. 8: existen múltiples entradas afectadas por coeficientes, que en este caso reciben el nombre de pesos sinápticos, y luego, dichas entradas son adicionadas, lo que se observa como unión sumadora. Finalmente, estos datos son transformados por una función de activación.

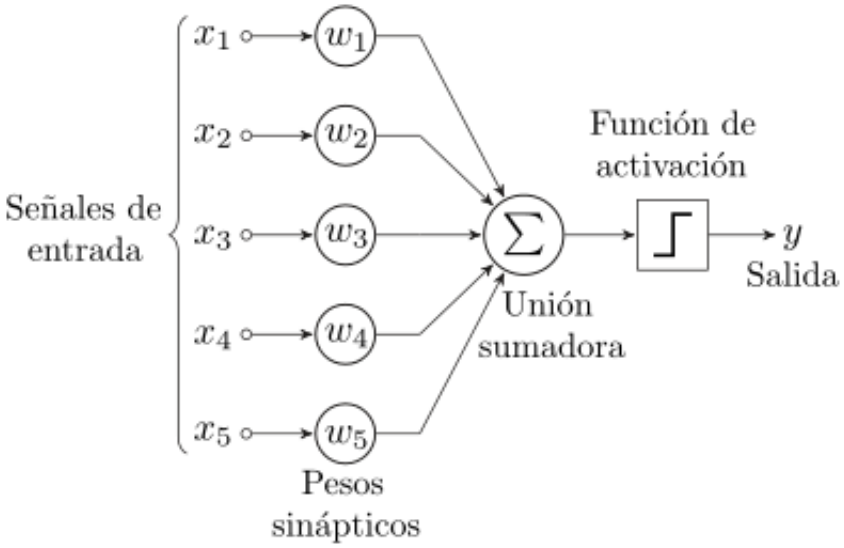


Figura 8: Perceptrón

### 3.5.2. Redes Neuronales

Las ANN funcionan como una estructura de preceptrones multicapa, debido a que dichos modelos se encuentran compuestos por una gran cantidad de perceptrones, en un patrón que se repite en las diferentes capas: *input*, capas ocultas y *output* del modelo y en paralelo según la cantidad de neuronas para una dada capa. Esto mismo se puede observar de manera gráfica en la imagen que se presenta a continuación (Fig. 9).

Por lo tanto, se tiene un modelo que una vez aplicado un patrón a la entrada de la red, el mismo se propaga capa a capa hasta llegar a producir una salida. Ahora bien, la pregunta que surge es, si se trata de un algoritmo de ML, ¿Cómo es que se produce el entrenamiento? La respuesta a dicho interrogante se encuentra en un algoritmo conocido como *backpropagation* que lo que hace es propagar el gradiente hacia atrás desde la capa de salida a todas las anteriores estimando el aporte de cada uno de los pesos al error cometido.

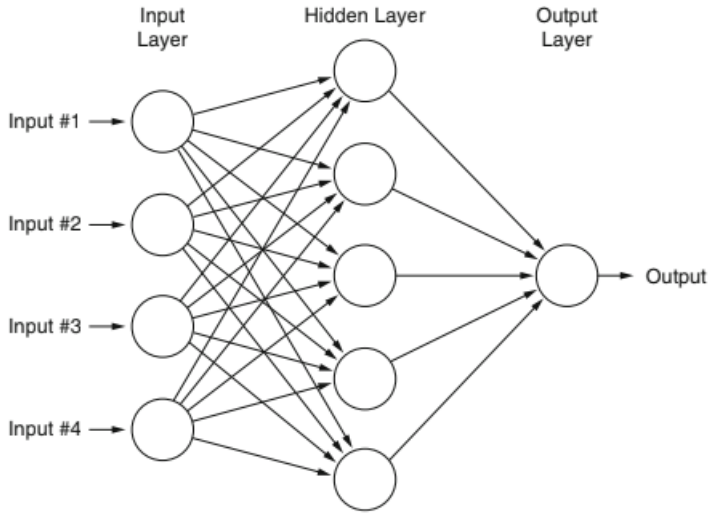


Figura 9: Capas Fully Connected [14]

## 3.6. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son un tipo de redes neuronales que se utilizan mayoritariamente para el procesamiento de imágenes. Esto se debe a que trabajan con *kernels* o filtros y por tanto, poseen la capacidad de procesar mosaicos de imágenes, teniendo en cuenta su distribución espacial, en lugar de tratarlos como píxeles aislados. Esto permite encontrar relaciones con grupos de píxeles que se encuentran en la vecindad uno del otro. En consecuencia, se obtienen resultados por mucho superiores a otros enfoques al utilizar este tipo de modelos para el tratamiento de imágenes.

## 3.7. Visión Computacional

Existe una manera de clasificar a la inteligencia artificial según el tipo de tarea que tiene por objetivo resolver. En base a este criterio, la IA se clasifica en Dominios. El presente trabajo se realiza dentro del Dominio conocido como Visión Computacional, el cual tiene por objetivo imitar la visión humana, y resolver tareas que dependan de ella. Ejemplos de dichas tareas son:

- **Clasificación:** Se asignan las imágenes a diferentes clases, mediante una etiqueta global. Las clases se encuentran definidas y el algoritmo devuelve una probabilidad de pertenencia a cada clase. Si se trata de una clasificación binaria, luego es posible definir un umbral para determinar la pertenencia a una

clase. En cambio, si la clasificación es multiclase, se toma la máxima probabilidad como pertenencia a dicha clase.

- **Detección de Objetos:** Se asignan etiquetas a hallazgos locales. Se indica mediante el uso de *bounding boxes* (BB), que no son más que rectángulos que enmarcan una región de la imagen, la presencia de un objeto perteneciente a una determinada clase. Esto implica que para una única imagen es posible de detectar más de un objeto e incluso que es posible detectar objetos pertenecientes a diferentes clases. A cada uno de estos hallazgos además de asignarle una región espacial de la imagen, se le asigna un nivel de confianza asociado a dicha detección.
- **Segmentación:** Se asignan etiquetas mediante el uso de máscaras. Por lo tanto, la clasificación es a nivel píxel. Es posible detectar elementos pertenecientes a más de una máscara utilizando diferentes colores. En este caso nuevamente es posible que una única imagen posea más de un elemento y perteneciente a diferentes clases.

Este trabajo se centrará tanto en la clasificación de imágenes como en la detección de objetos dentro de ellas y es por ello que las siguientes secciones se detienen con mayor detalle en los conceptos necesarios para comprender cómo se busca resolver esas tareas.

### 3.7.1. Clasificación

Los principales componentes de las CNN son los siguientes:

- Capas convolucionales
- Funciones de activación
- Capas de *pooling*
- Una capa de *flattening*
- Capas *fully connected*

#### 3.7.1.1. Capas Convolucionales

Matemáticamente, las convoluciones se definen de la siguiente manera

$$(f * g)_{(t)} = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{1}$$

Llevado al mundo discreto y para el caso particular de imágenes, se aplica tal como lo muestra la Fig.10:

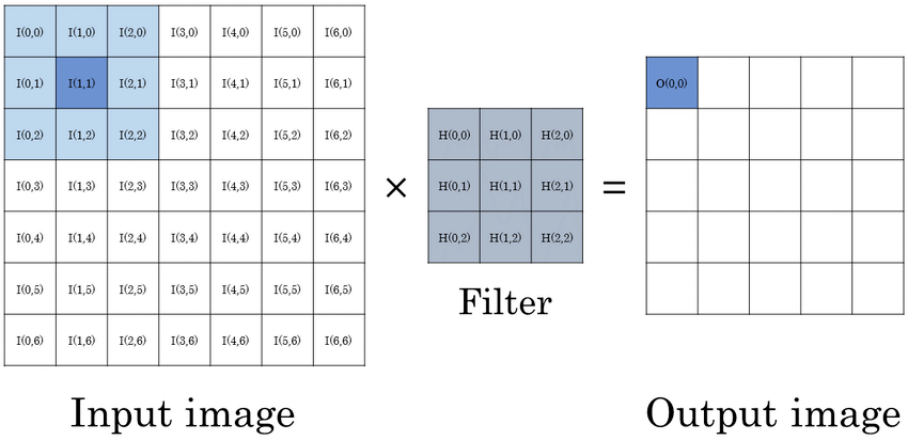


Figura 10: Convolución en Imágenes. Imagen obtenida de [15]

En este caso, se cuenta con una imagen de entrada, que para la primera capa es la imagen que queremos clasificar, y se cuenta con un filtro, que es una pequeña matriz a la cual se le definen las dimensiones de antemano y cuyos coeficientes son aprendidos por el modelo.

La convolución entonces tiene lugar cuando los elementos de la imagen de entrada son multiplicados uno a uno con los del filtro, sumados y colocados en la imagen de salida en la posición central del filtro. Esto ocasiona, tal como se puede ver en la imagen de salida, una disminución en las dimensiones dadas por la forma del filtro que se aplica. Sin embargo, de querer mantener las dimensiones iniciales, se podría suponer que la imagen de entrada se extiende más allá de sus límites, por ejemplo mediante el agregado de ceros, lo que se conoce como *padding same*.

Otro hiperparámetro (parámetro que no se entrena con el modelo) factible de ser modificado y afectar las dimensiones de salida de la imagen se conoce como *stride*, es decir, cada cuantos píxeles de la imagen original se detiene el centro del filtro.

Para poner todo lo anterior en términos matemáticos y comprender exactamente como afectan dichos hiperparámetros al modelo y en particular a las dimensiones de las imágenes, es que se presenta la siguiente ecuación, dónde $H_i$ representa la altura en píxeles de la imagen pero podría también ser el ancho, $H_0$ representa la dimensión a la entrada, $H_1$ la dimensión a la salida, P representa el *padding*, f el tamaño del filtro en dicha dimensión y S el *stride*.

$$H_1 = \frac{H_0 + 2 * P - f}{S} + 1 \tag{2}$$

Una última cuestión a tener en cuenta referida a las capas convolucionales y que escapa a la Fig. 10 es la profundidad de la imagen. Para poder realizar una convolución, de contar con una imagen con más de un canal de profundidad, es necesario que el filtro que se aplique iguale a dicha dimensión de la imagen. Por lo tanto, la convolución transformará a la imagen con profundidad en una imagen de un único canal de profundidad. Pero en general, en un modelo de CNN, por cada capa se aplican varios filtros a la imagen de entrada, lo que resulta en una dimensión de profundidad de salida igual a la cantidad de filtros aplicados a la imagen.

#### 3.7.1.2. Funciones de Activación

Las funciones de activación se utilizan con el objetivo de transformar los resultados a la salida de las etapas convolucionales, para mejorar el entrenamiento.

Un tipo de función de activación ampliamente utilizada por las mejoras que ha introducido a la hora de entrenar modelos, es conocida como ReLU o *rectified linear unit*, que se describe a continuación:

$$f(x) = \begin{cases} 0 & si \quad x \leq 0 \\ \\ x & si \quad x > 0 \end{cases} \tag{3}$$

Esta función termina por ende, llevando todos los píxeles negativos de la imagen a cero y el resto los mantiene en su valor original.

Existen otras funciones de activación también ampliamente utilizadas. Entre ellas se encuentra la *Leaky Relu*, la cual se presenta en la Fig.11. Esta función es similar a la anterior, pero en lugar de mantener todos los píxeles menores a cero nulos, los multiplica por un valor constante determinado en forma de hiperparámetro. De esta manera, terminan importando los valores negativos pero en una proporción ajustable.

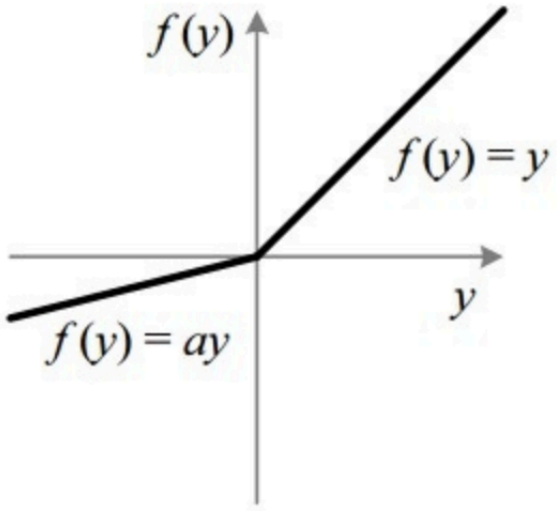


Figura 11: *Leaky Relu*

Los tipos de funciones de activación anteriormente mencionados se suelen utilizar en capas intermedias, es decir, en las conocidas como *hidden layers*. Esto se debe a que en dichas capas no es necesario devolver una función de densidad de probabilidades. Sin embargo, en la capa final de cualquier modelo de clasificación, se vuelve sumamente importante dicha característica. En este caso, se suelen diferenciar los modelos de clasificación binaria, de los de clasificación multiclase. Para el primero de los casos, se suele utilizar la función de activación que se conoce como sigmoidea, la cual se describe en la ecuación 4 y se muestra de manera gráfica en la Fig. 12.

$$\sigma(x) = \frac{1}{1 + e^{(-x)}} \tag{4}$$

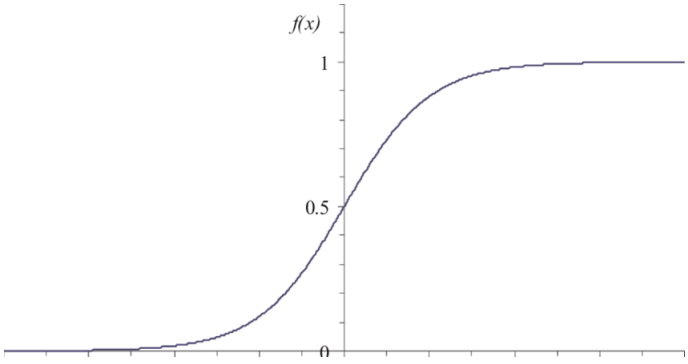


Figura 12: Función de Activación Sigmoidea

Para el segundo de los casos, se suele utilizar en la capa final del modelo de redes neuronales una función de activación conocida como *softmax*. Esta función mapea los resultados obtenidos como consecuencia de la convolución a una distribución de probabilidades, mediante la expresión que se presenta en la ecuación 5.

$$f_i(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \tag{5}$$

La función de entrada es un vector K-dimensional, cuyos valores pertenecen al conjunto de los números reales, y se encuentran representados en la ecuación por el $z_i$. La sumatoria en le denominador de la división genera el efecto normalizador y como para todos los casos el denominador no varía y el numerador es menor que el denominador, se termina obteniendo una función de distribución de probabilidad.

#### 3.7.1.3. Pooling

En la capa de *pooling* se escoge entre ciertos píxeles, un valor que represente al conjunto. Es decir, se pierde información en este paso, pero se realiza de tal manera de captar la información más importante del conjunto. Esto se puede realizar de diferentes formas, como por ejemplo aplicando *max pooling*, es decir, conservar el mayor valor del grupo o *average pooling*, tomar el promedio de los píxeles del grupo.

En general, el más utilizado en CNN es el *max pooling*, ya que le otorga al modelo la capacidad de detectar los *features* mas salientes, como por ejemplo los bordes.

El hecho de aplicar *pooling* permite disminuir el tamaño de las imágenes con las que se trabaja, lo cual convierte el proceso en mucho más eficiente, y con menos parámetros entrenables. Esto resulta una ventaja porque a mayor cantidad de estos parámetros, mayor riesgo de *overfitting*, es decir, menor capacidad de generalización del modelo.

#### 3.7.1.4. *Flattening*

Una vez pasadas las imágenes por varias capas convolucionales seguidas de sus correspondientes *poolings*, se suele hacer un proceso de *flattening* de las imágenes, que es tomar la matriz que corresponde a las imágenes y colocar todos los píxeles en forma de vector, perder la representación matricial y dejar el modelo listo para ingresar a capas densas o *fully connected*.

Para el ingreso a la próxima capa es que es de especial importancia que las imágenes posean las mismas dimensiones.

#### 3.7.1.5. *Fully Connected*

Finalmente, luego de las sucesivas representaciones obtenidas mediante convolu-

ciones rectificadas, de la obtención de las características más representativas y de acomodar un cambio de la forma en la que se representan las imágenes, se ingresan a las capas densas o *fully connected*, que es el paso por una red neuronal artificial, como ya fue descripta en la sección 3.5.2.[16]

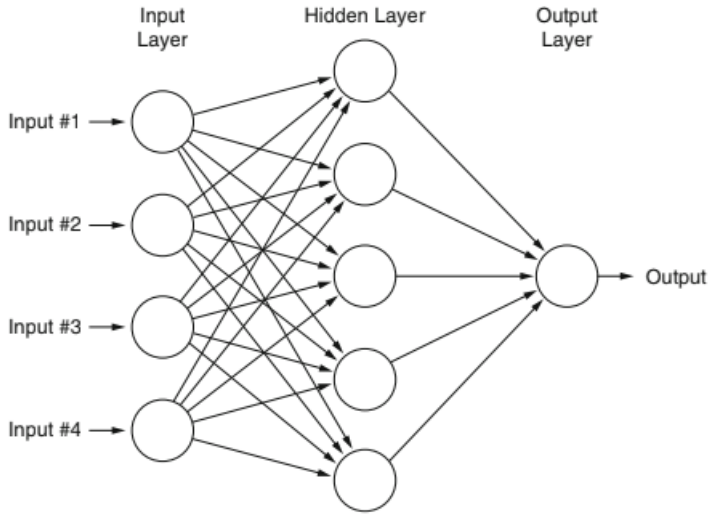


Figura 13: Capas *Fully connected* [14]

## 3.8. Detección de Objetos

### 3.8.1. Localización de Objetos

El concepto localización de objetos se refiere a la acción de detectar la presencia o ausencia de un elemento dentro de una dada imagen y de existir presencia, de enmarcar dicho elemento por medio del uso de un BB, tal como se muestra en la Fig. 14 . La diferencia entre la detección de objetos y la localización de objetos radica en que el primer grupo es capaz de detectar la presencia de múltiples objetos en una misma imagen y asignarle una probabilidad de pertenencia a una clase a cada uno de los objetos por separado, mientras que la localización de objetos no es más que una clasificación global de la imagen más el agregado del dato de dónde se encuentra posicionado el objeto que determina la presencia a dicha clase.

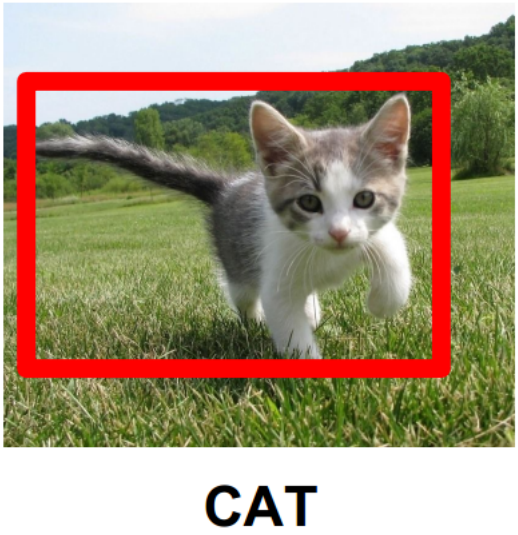


Figura 14: Ejemplo Localización de Objetos

En cuanto a las diferencias con la tarea de clasificación, la resolución de este tipo de tareas implica como salida del algoritmo no simplemente un vector que indique la probabilidad de pertenencia a cada una de las clases, sino que 4 parámetros extra que son $b_x$, $b_y$, las coordenadas del centro del BB, y $b_w$ y $b_h$, el ancho y alto del BB. Otra diferencia con las tareas de clasificación es que en los casos de localización de objetos siempre debe existir una clase en la que no se detecte ningún tipo de objeto. Por ejemplo, si se quisieran detectar gatos y perros, además debería existir una clase llamada por ejemplo fondo o vacío. Entonces, el vector de salida debería estar compuesto por un primer elemento que podría considerarse como pertenencia a la clase fondo, que lo que hace es darle valor o no al resto de los parámetros. Si el primer coeficiente de salida es cercano a cero, el resto de los coeficientes de salida carecen de valor, ya que esto indica que no existe un objeto detectable en dicha imagen. Si por el contrario dicho coeficiente posee un valor cercano a la unidad, esto indica la presencia de un objeto, y entonces los otros coeficientes de salida cobran relevancia, ya que definen el tipo de objeto y su extensión dentro de la imagen de entrada.

### 3.8.2. Detección de Objetos

Como ya fue mencionado previamente, la detección de objetos es una tarea que permite detectar más de un elemento en una única imagen. Una forma en la que puede lograrse esta tarea es mediante un algoritmo conocido como *sliding window detection*. El mismo requiere de una red convolucional como aquellas utilizadas para clasificación de imágenes que únicamente contengan un objeto que se quiera detectar

y que el mismo ocupe prácticamente la totalidad de la imagen. Una vez entrenada dicha red, lo que hace el algoritmo es ir seccionando porciones de la imagen y clasificando cada una de ellas. Esta selección se realiza mediante "ventanas" separadas por un determinado *stride*, de ahí el nombre del algoritmo. Además, para garantizar la detección independientemente de la escala en la que se encuentre el objeto, se realiza el proceso de selección mediante ventanas cambiando el tamaño de las mismas. Una gran desventaja que posee este algoritmo es que consume muchos recursos de procesamiento y no resulta eficiente en cuanto a tiempos. Esto es consecuencia de que se debe hacer pasar por una red neuronal cada una de las ventanas de selección manual. Y por si fuera poco, cuanto menor el *stride*, es decir, mejor la detección, mayor cantidad de pasadas por la red de clasificación y por lo tanto, menor eficiencia aún.

Debido a la mencionada desventaja es que se desarrolló una implementación convolucional del algoritmo *sliding window detection*. La misma permite procesar en paralelo las detecciones de las diferentes ventanas, disminuyendo mucho los tiempos de implementación. Sin embargo, este algoritmo sigue teniendo dos grandes desventajas. En primer lugar, la detección se da en tamaños fijos, no permitiéndole al algoritmo adaptarse a cualquier escala en la que se presente el objeto. En segundo lugar, de encontrarse más de un objeto en una misma ventana, es imposible la correcta identificación de ambos.

Como solución a este último problema, surgieron *anchor boxes*. Asociado a cada una de las celdas se describen BBs con diferentes tamaños iniciales, representativos de la forma esperada de objetos a detectar. Cuando se detecta un objeto en una imagen, asociado a una cierta ventana, se calcula el IoU entre la detección y cada *anchor box*, y se asigna dicha detección al *anchor box* con mayor resultado.

## 3.9. *Transfer Learning*

Los entrenamientos en los que se usa *transfer learning* (TL) son aquellos que parten de un modelo pre-entrenado, con el objetivo de cumplir una dada tarea distinta a la que en esta instancia se desea resolver. Es decir, a utilizar los pesos pre-entrenados por un modelo, tomarlos como punto de partida para seguir entrenando, hasta obtener valores óptimos para la nueva tarea que se desea resolver.

En particular, este enfoque suele utilizarse mucho para aplicaciones de *deep learning*, para resolver tareas en el campo de Visión Computacional y *Natural Language Processing(NLP)*, ya que suele implicar mejoras en tiempo de procesamiento y *performance* del modelo.

## 3.10. Arquitecturas

Tal como fue mencionado previamente, *deep learning* consiste en sucesivas capas de procesamiento de los datos. Estas sucesivas capas se encuentran lejos de ser homogéneas. Debido a que el trabajo aquí presentado se realiza con CNN, los diferentes bloques son los descriptos en la sección correspondiente. La forma de concatenar los mismos, así como la selección de ciertos hiperparámetros es lo que da origen a una variedad de arquitecturas. Estas arquitecturas se encuentran desarrolladas para resolver un tipo de tarea en específico. Sin embargo, al momento de resolver un problema en particular, muchas veces se desconoce *a priori* la arquitectura que arrojará los mejores resultados. Por lo tanto, resulta de utilidad comparar arquitecturas implementadas sobre el propio *dataset*.

A continuación, se mencionan las diferentes arquitecturas implementadas a lo largo de todo el trabajo. En primer lugar, se explican la VGG16, ResNet50 e Inception V3, las cuales se utilizan para resolver tareas de clasificación y luego, se explica el funcionamiento de YOLO V5, arquitectura implementada para la detección de objetos.

### 3.10.1. VGG16

VGG es una arquitectura de red neuronal propuesta en el año 2014 por K. Simonyan y A. Zisserman en su *paper* titulado *"Very Deep Convolutional Networks for Large-scale Image Recognition."* [17]

Esta arquitectura recibió mucha atención en su momento ya que frente al famoso problema de clasificación de ImageNet, el cual consiste en clasificar más de 14 millones de imágenes en 1000 clases, se posicionó en el *top* 5 de mejores resultados y dicha arquitectura es considerablemente simple en comparación con las otras analizadas en este trabajo.

En la figura 15 [18], podemos ver de manera gráfica la arquitectura VGG - 16. El *input*, en este caso son las imágenes radiográficas, las cuales previamente fueron submuestreadas para obtener imágenes de 224x224. En cuanto a la profundidad de la imagen, si bien las radiografías no son de color, es decir, podrían reducirse a un único canal, se decidió repetir los valores en los tres canales por dos motivos: 1) Para poder utilizar la arquitectura ya implementada en Keras y 2) Para poder implementar ciertos pesos ya preentrenados, es decir, utilizar *transfer learning* (TL) y comparar con modelos sin TL.

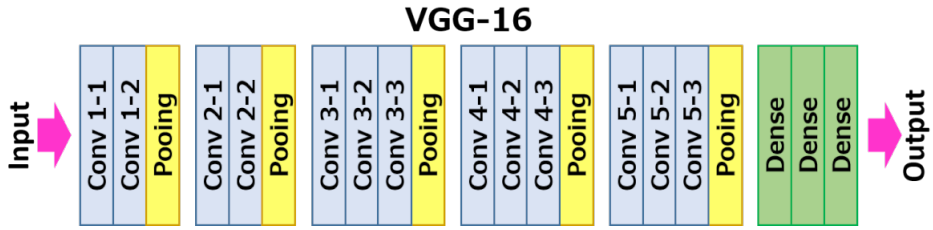


Figura 15: Arquitectura VGG16

En VGG - 16, las imágenes que ingresan al modelo pasan por 5 bloques convolucionales. El primero y el segundo de dichos bloques se encuentran conformados por dos capas convolucionales, cada una seguida por una función de activación ReLU, y luego de la segunda capa convolucional sigue un *max pooling*. Los tres últimos bloques convolucionales son idénticos a los primeros, pero suman una capa convolucional más previa al *max pooling*. En cuanto a características de las capas convolucionales, se suelen utilizar filtros pequeños, *stride* de 1 tanto en alto como en ancho y *padding same* (para que entre las capas convolucionales el tamaño se mantenga).

Luego, se realiza un *flattening* y se hacen pasar a través de tres capas densas. Las dos primeras capas constan de 4096 neuronas cada una y la tercera, por ser la capa de salida, es la que debe adaptarse para permitir la resolución de las tareas propuestas.[19]

### 3.10.2. ResNet50

La arquitectura ResNet lleva su nombre en referencia a *Residual Network*. La misma se hizo conocida cuando surgió como la arquitectura ganadora en la competencia de ImageNet de 2015.

Esta arquitectura introdujo el concepto de *skip connections*, gracias al cual es posible entrenar mas de 150 capas de profundidad de manera más sencilla, sin caer en *vanishing gradient*. *Vanishing gradient* es un fenómeno que ocurre al actualizar los pesos. Como se parte del resultado de la función de costo y de allí se comienzan a actualizar los pesos hacia las primeras capas con la siguiente fórmula,

$$\omega_{new} = \omega_{old} - \eta \cdot \frac{\partial L}{\partial \omega_{old}} \tag{6}$$

a medida que se recalculan los pesos de las primeras capas, la actualización de los pesos es cada vez menor como consecuencia de la derivada en cadena.

El concepto de *skip connections* lo que hace es sumar el input de una capa a la salida de otra tal como se muestra en la Fig. 16, creando "atajos" entre ciertas capas, lo que resulta en menos pérdida al momento de propagar el gradiente por este camino.


Figura 16: *Skip connections* introducidas en ResNet

En cuanto a la ResNet 50, la misma lleva dicho nombre debido a que se encuentra compuesta por 50 capas, las cuales se muestran en la Fig. 17.

Al ingresar a la ResNet, las imágenes ingresan a una capa convolucional, con un filtro de 2x2, *padding* de 3 y *stride* de 2. El efecto de una capa convolucional sobre una imagen se encuentra dado por la Ec. 2 presentada anteriormente. Por lo tanto, teniendo en cuenta que las imágenes a la entrada poseen 128x128x3, la salida de esta capa resulta de 64x64, tal como se muestra a continuación.

$$H_1 = \frac{128 + 2 \cdot 3 - 7}{2} + 1 = 64 \tag{7}$$

En cuanto a la profundidad, como fue mencionado anteriormente se obtiene a partir de la cantidad de filtros que se hagan pasar por las imágenes. En este caso son 64.

Por lo tanto, si se quisiera calcular la cantidad total de parámetros para esta primera capa convolucional, se debe hacer el siguiente cálculo: $7x7x3x64 + 64$, dónde $7x7x3$ son las dimensiones del filtro, los 64 multiplicando son la cantidad de filtros y los otros 64 son los bias.

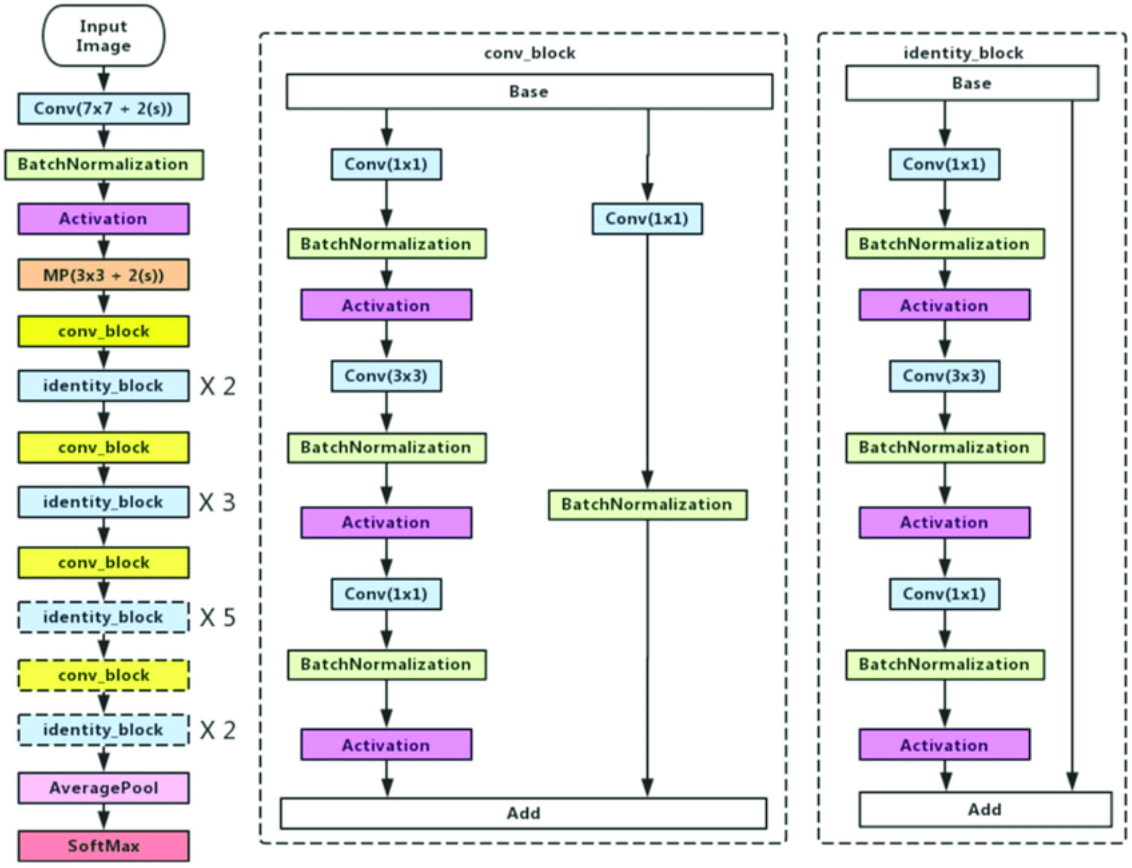


Figura 17: ResNet50

Luego de la capa convolucional, la sigue una capa de *batch normalization* en la cual se normalizan los valores de salida, restándoles la media y dividiéndolos por el desvío, y luego se los multiplica por un valor arbitrario y se les suma otro. Estos cuatro parámetros son entrenables para el modelo e independientes de cada filtro. Por lo tanto, en esta capa surgen $64 \cdot 4 = 256$ nuevos parámetros entrenables.

Luego, se pasan los valores obtenidos por una capa de activación ReLU, y a continuación por una capa de *max pooling*, con un filtro de $3x3$, un *padding* de 1 y un *stride* de 2. Con estos valores, a la salida de la capa se tiene una imagen de $32x32x64$, con la diferencia de que en las capas de *pooling* no hay parámetros entrenables, ya que simplemente es un submuestreo.

A continuación, las imágenes comienzan a pasar por una serie de bloques, entre los que se diferencian los bloques convolucionales de los bloques de identidad. Estos bloques también se encuentran explicados en la Fig. 17. Lo que tienen en común ambos bloques es que incorporan el concepto explicado anteriormente de *residual networks*.

Es decir, sumar la entrada del bloque a la salida de la función de activación. Ahora bien, en lo que se diferencian es en cómo se realiza dicha suma. En los casos en que las dimensiones a la entrada y la salida de mantienen y la suma puede hacerse de manera directa, nos encontramos frente a un *identity block*, mientras que en los casos en que es necesario adaptar las dimensiones para poder sumarlas, se suele poner en el medio una convolución de 1x1, con un *stride* apropiado, para conseguir llevar las dimensiones de entrada a las de salida.

Luego de la gran cantidad de capas que componen los bloques mencionados anteriormente, se realiza un *average pooling* y una capa densa con una función de activación. En este caso, el diagrama muestra una *softmax*, capaz de realizar clasificaciones multiclase, pero bien podría ser una sigmoidea en el caso en que la cantidad de clases a la salida fueran dos.

### 3.10.3. *Inception*

Previo a la aparición de las arquitecturas *Inception*, las estrategias más utilizadas consistían simplemente en apilar capas sobre capas. Por el contrario, esta familia de arquitecturas se diseñó cuidadosamente con el objetivo de obtener mejoras en cuanto a la velocidad de entrenamiento y a los resultados obtenidos.

Uno de los primeros problemas que se quisieron resolver, es la capacidad de detección de la red independientemente de la porción de imagen que ocupe el objeto que resulta en la clasificación. La solución de *Inception* para este problema consiste en paralelizar las representaciones de diferentes imágenes mediante bloques como el que se muestra a continuación:

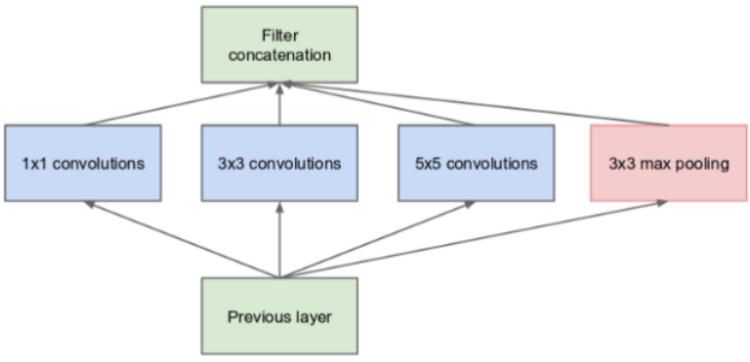


Figura 18: Procesamiento en paralelo Inception

A partir de dichos bloques se construye una arquitectura con una gran cantidad de capas, que a su vez tiene la particularidad de presentar lo que se conocen como

*auxiliary classifiers*, clasificadores que arrojan los resultados de capas intermedias. Los mismos son de utilidad para garantizar que las sucesivas representaciones en las diferentes capas no se encuentren tan alejadas de la tarea de clasificación que persigue el clasificador final y resultan de utilidad para atacar el problema de *vanishing gradient*.

### 3.10.4. YOLO

Este tipo de arquitectura se encuentra diseñado para detección de objetos. Es decir, dada una imagen identifica mediante *bounding boxes* (BB)s todos los objetos para los que fue entrenado el algoritmo y los clasifica.

El sistema divide la imagen de entrada en celdas de $nxn$. Si el centro de un objeto cae en una cierta celda, entonces la detección de dicho objeto es responsabilidad de esa celda. Cada celda predice BBs. Cada BB se encuentra descripta por los siguientes parámetros de salida:

- (x,y): coordenadas del centro del BB
- (w,h): ancho y altura
- c: nivel de confianza de la detección.

Esta arquitectura se basa en la implementación convolucional de la *sliding window*, e incluye la utilización de *anchor boxes*.

En la Fig.19, se observa de forma esquemática la arquitectura de YOLO. Está compuesta por 24 capas convolucionales y 2 capas densas. Las capas iniciales convolucionales realizan extracción de características, mientras que las últimas capas *fully connected*, realizan las predicciones de las características de los BBs. Existen capas convolucionales de $1x1$ que lo que buscan es reducir el espacio de características obtenidas a partir de las capas convolucionales previas.

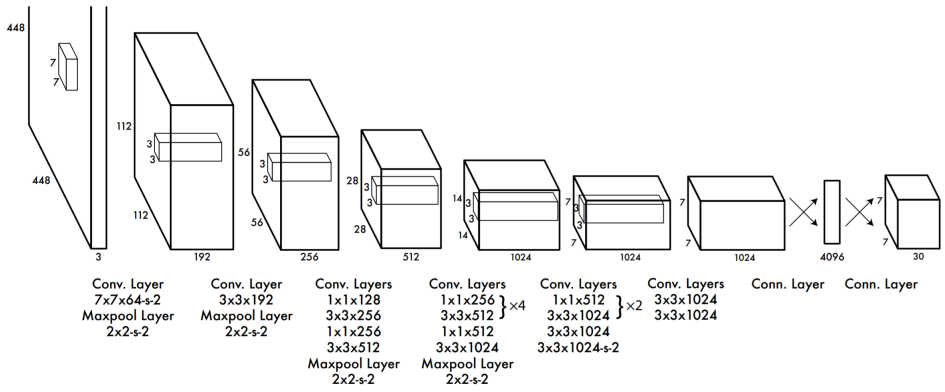


Figura 19: Arquitectura YOLO. Imagen obtenida de [20]

La arquitectura YOLO V5 se encuentra basada en la YOLO, con leves modificaciones que demostraron mejorar la capacidad de la arquitectura para resolver tareas de detección de objetos. YOLO v5 cuenta en total con 4 versiones las cuales difieren levemente en la cantidad de capas del modelo, y en la cantidad de parámetros que lo componen. Eso termina repercutiendo en la velocidad y *performance* del modelo. Dentro de estas versiones, YOLO V5m posee un tamaño intermedio, así como *performance* y velocidad de respuesta.
En cuanto a la elección de funciones de activación, YOLO V5 se basa en el uso de *Leaky ReLU* para las capas intermedias (*hidden layers*), mientras que en la capa final utiliza función sigmoidea. [21]

## 3.11. Regresión Logística

Un modelo de regresión logística puede representarse como un perceptrón con una función de activación sigmoidea. Matemáticamente, la frontera de decisión es lineal en el espacio de las entradas, y se define de la siguiente forma:

$$logit(P) = \ln(\frac{P}{1-P}) = \alpha_0 + \alpha_1 * x_1 + ... + \alpha_n * x_n \tag{8}$$

donde, $logit(P)$ es el logaritmo natural del ODDS del evento, $\alpha_i$ son los coeficientes que van a ser calculados mediante el método de máxima verosimilitud de manera de optimizar la predicción correcta, y $x_i$ representan las variables de entrada del modelo.
El método de máxima verosimilitud se basa en la idea de que la muestra obtenida, por haber ocurrido tiene una alta probabilidad de ocurrir y estima los parámetros como aquellos que maximizan la probabilidad de obtener dicha muestra. Para ello, se busca iterativamente maximizar la función de verosimilitud, que es la función de

probabilidad conjunta de la muestra.

En los resultados de la regresión logística, el modelo arroja una gran variedad de parámetros. Entre ellos se encuentra el p-valor que acompaña a cada una de las variables. El hecho de encontrar un p-valor implica que se realizó un test de hipótesis. En este caso, el test de hipótesis es conocido como test de Wald, y plantea como hipótesis las siguientes:

$$H_0 : \alpha_i = 0 \tag{9}$$
$$H_1 : \alpha_i \neq 0$$

Es decir, que para cada una de las variables $\alpha_i$ se realiza un test de hipótesis separado en el que se asume que su valor es cero, se calcula el estadístico del *test* y se obtiene la probabilidad de que dicho valor sea obtenido producto del azar dado que la hipótesis nula es verdadera. Escogiendo un nivel de confianza del 95 % y realizando un *test* a dos colas, se debe tomar como umbral un p-valor de 0.05. Si el valor obtenido es inferior al umbral, se debe rechazar la hipótesis nula, debido a que la probabilidad de obtener la muestra es demasiado pequeña y por ende se debe aceptar la hipótesis alternativa. De lo contrario, se debe rechazar la hipótesis alternativa y como consecuencia, aceptar la nula.

## 3.12. Funciones de Costo

Las funciones de costo cumplen la función de calcular la diferencia entre los valores predichos por un modelo y su *ground truth* (GT), y son las funciones sobre las que trabajan los optimizadores. Existen diferentes funciones de costo según la tarea que se desea resolver.

Para tareas de clasificación binaria, se suele utilizar el algoritmo de pérdida logarítmica (Ver Ec. 10).

$$\text{Loss} = -y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \tag{10}$$

Donde, $\hat{y}_i$ representa el $output_i$ del modelo, y $y_i$ su GT.

Para tareas de clasificación multiclase, se suele utilizar la *Categorical Cross Entropy* que busca cuantificar la diferencia entre dos distribuciones de probabilidad y se presenta en la Ec. 11.

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output} \\ \text{size}}} y_i \cdot \log \hat{y}_i \tag{11}$$

Nuevamente, $\hat{y}_i$ representa el $output_i$ del modelo, y $y_i$ su GT.

## 3.13. Optimizadores

Como ya fue mencionado previamente, los optimizadores cumplen la crucial función de actualizar los valores de los pesos a partir del resultado obtenido en la función de costo. Esto lo realizan estimando la contribución parcial de cada uno de los pesos, es decir, calculando la derivada parcial de la función de costo con respecto a cada uno de los parámetros del modelo. Ahora bien, existen múltiples optimizadores que lo realizan de diferentes maneras, y con resultados finales variados.

Uno de los optimizadores más utilizados es conocido como *stochastic gradient descent* (SGD). Este algoritmo toma aleatoriamente una observación por *batch* y le calcula la derivada parcial a cada uno de los pesos. Estos pesos, se terminan actualizando tal como lo muestra la Ec. 12

$$W_{t+1} = W_t - lr * \frac{df_{costo}}{dW} \tag{12}$$

SGD fue uno de los primeros optimizadores utilizados, al cual se le fueron adicionando una gran cantidad de mejoras con el fin de obtener los pesos óptimos y en la menor cantidad de iteraciones. Un ejemplo de dichas adaptaciones que fueron surgiendo es el agregado de un *momentum*. Intuitivamente este algoritmo acelera en función de un hiperparámetro del modelo conocido como *momentum* el descenso por el gradiente, en el caso en el que la dirección de descenso actual se parezca a la de la iteración o iteraciones anterior/es.

Una variación que surgió posteriormente es conocida con el nombre de *adaptative gradient algorithm*. En este caso, se utiliza un *learning rate* distinto para cada peso, en lugar de un mismo valor para todos ellos, tal como se hacía anteriormente. Sin embargo, para que siga resultando viable la implementación, el usuario debe escoger un solo valor inicial y a partir de este y las direcciones de optimización obtenidas en las sucesivas iteraciones, se van actualizando a su vez el resto de los valores de *learning rate*.

El siguiente optimizador es muy similar al anterior, pero en lugar de tomar los valores de los gradientes acumulados en todo el entrenamiento, se toma una ventana compuesta por las últimas n iteraciones. Este optimizador se conoce con el nombre de Adadelta.

Luego, surgió un optimizador conocido por el nombre de *root mean square propagation* (RMSProp). Este también toma un *learning rate* diferente para cada peso,pero

en este caso, intenta solucionar un problema de su predecesor que es que los valores de *learning rate* de los distintos pesos podían poseer un rango demasiado amplio. Finalmente, se desarrolló Adam, un optimizador que combina al *adaptative gradient algorithm* y RMSProp. Existe un factor de entrenamiento por parámetro, que además se ve afectado por la media del *momentum* del gradiente.[22] [23]

## 3.14. Métricas

Escoger una métrica de referencia permite comparar modelos de manera objetiva. En este apartado se mencionan las métricas utilizadas para resolver las distintas tareas presentadas en este trabajo.

Definiciones de particular interés para la definición de métricas en tareas de clasificación son las siguientes:

- **True positive (TP):** En español, verdaderos positivos (VP). Son todos aquellos elementos pertenecientes a una clase según su GT y a su vez clasificados por el modelo como pertenecientes a dicha clase.
- **False positive (FP):** o falsos positivos. Todos aquellos elementos clasificados por el modelo como pertenecientes a determinada clase, pero que según su GT no pertenecen a dicha clase.
- **False negative (FN):** Todos aquellos elementos clasificados como no pertenecientes a una determinada clase por el modelo, pero que su GT determina que si pertenecen.
- **True negative (TN):** En español, verdaderos negativos (VN). Todos aquellos elementos clasificados por el modelo como no pertenecientes a una determinada clase y que su *ground truth* determina que no pertenecen a dicha clase.

A partir de las definiciones anteriores, se calcula la sensibilidad o *recall*, tal como se presenta a continuación:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Otra medida que se utiliza para evaluar la *performance* de los modelos es el valor predictivo positivo, o también conocido como *precision*, que se presenta en la ecuación siguiente:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

A partir de los valores de *precision* (P) y *recall* (R), se puede calcular el *F1 - score* tal como se puede observar en la Ec. 15.

$$F1_{Score} = 2 \cdot \frac{P \cdot R}{P + R} \tag{15}$$

El *F1-score* suele utilizarse como métrica debido a que tiene en cuenta tanto la cantidad de falsos negativos y falsos positivos.

Las definiciones anteriores permiten inferir que se obtienen diferentes valores de *precision* y *recall* por clase, y por lo tanto de *F1-score*. Dichos valores pueden combinarse de diferentes maneras para obtener una única métrica en los casos de clasificación multiclase, como por ejemplo mediante el *F1 - macro score*, que se presenta en la Ec. 16

$$F1_{MacroScore} = \frac{\Sigma_{i=0}^{N} F1_{Score_i}}{N} \tag{16}$$

Para tareas de detección de objetos, una métrica de particular interés es el *mean average precision* (mAP). Esta se obtiene a partir del *intersection over union* (IoU), que es el cociente entre el área de solapamiento de los bounding boxes de la predicción y su *ground truth* y la unión de dichas áreas (Ver Fig. 20).

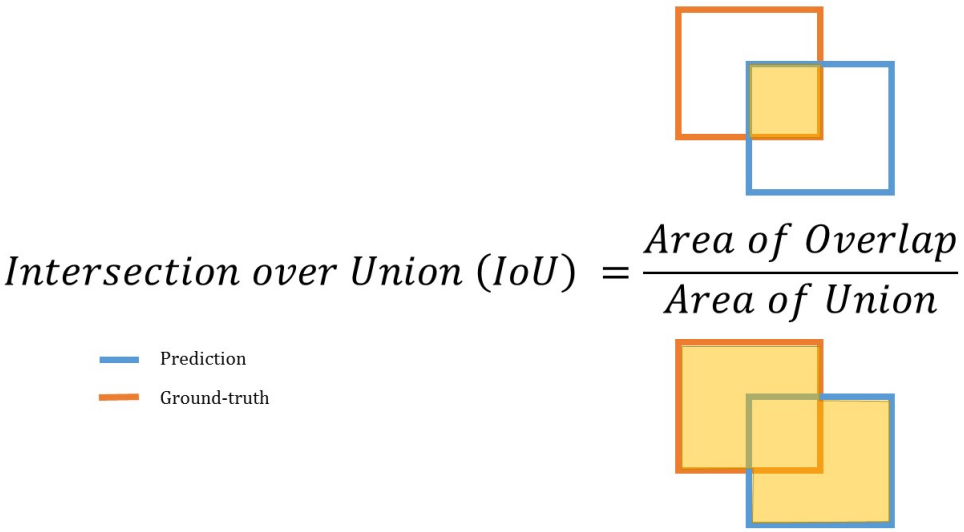


Figura 20: INTERSECTION OVER UNION (IOU)

Una vez obtenido el IoU, se escoge un umbral para la determinación de VP, VN, FP y FN y a partir de dichos valores, se calculan *precision* y *recall*, tal como en las tareas de clasificación. Finalmente, el average precision (AP), se calcula como el área debajo de la curva de *precision - recall* y el mAP, es el promedio de APs para las diferentes clases. En el caso de realizar detecciones de una sola clase, entonces el AP y el mAP son lo mismo.

El AIC (*Akaike's Information Criterion*) o criterio de información de Akaike, se utiliza como métrica para modelos de regresión logística y se calcula de la siguiente manera:

$$AIC = 2\ln(\frac{e^k}{L}) \tag{17}$$

Donde $k$ representa la cantidad de parámetros del modelo, mientras que $L$ la función de máxima verosimilitud ya optimizada. Es decir, el criterio toma en cuenta tanto la cantidad de variables incluidas en el modelo, penalizando modelos más complejos, así como también la bondad de ajuste del modelo, favoreciendo a los modelos que mejor ajuste producen. Esto permite escoger los mejores modelos, evitando el *overfitting*.

## 3.15. Validación Clínica

En la actualidad, las técnicas de *deep learning* (DL) están obteniendo resultados muy prometedores en una gran variedad de tareas. En particular, se destacan aquellas tareas de visión computacional, en las que las máquinas equiparan e incluso en algunos casos superan a los humanos más experimentados.

Sin embargo, para aplicaciones en el campo de la salud, debe tenerse especial cuidado debido a que las herramientas han demostrado no ser tan sencillamente transferibles de un centro de salud a otro. Esto es como consecuencia de la variabilidad en determinación de etiquetas, es decir, a una falta de criterio único. Es por ello, que para garantizar la utilidad de los sistemas en un entorno clínico apropiado, resulta imperioso validar los modelos con *datasets* propios del centro de salud que los va a incorporar, debido a que los mismos incorporan los criterios propios de los profesionales de salud que allí se desempeñan. [24]

## 3.16. Sistemas de Soporte a la Toma de Decisiones

Los sistemas de soporte a la toma de decisiones (SSD) son sistemas informáticos diseñados con el objetivo de asistir el proceso de toma de decisiones relativas al cuidado de la salud de un paciente, por parte de los profesionales de la salud. El teorema de la informática médica, propuesto por Charles Friedman, afirma que una persona trabajando en conjunto con una fuente de información, es mejor que la misma persona sin ayuda. Sin embargo, para ser efectivos, estos sistemas deben poseer ciertas características:

- Relevantes: deben presentar la información correcta, basada en evidencia y con el objetivo de resolver una necesidad clínica.

- Dirigidos a la persona correcta: aquella encargada de tomar la decisión que afecta a dicha necesidad clínica.
- Mediante los canales correctos: presentada en una forma práctica, con capacidad de ser incorporada sin grandes modificaciones a los flujos clínicos que complementan.
- En los formatos de intervención correctos: no deben resultar invasivos, deben presentar la información de manera clara y deben permitir a la persona encargada de la toma de decisión tener la última palabra.
- En los puntos correctos del flujo clínico: las recomendaciones deben ser presentadas en el momento justo en el que la decisión debe ser tomada. [4] [25]

# 4. Desarrollo

En este apartado se describen tanto los materiales y métodos como los resultados de la totalidad de los experimentos realizados en el marco de este proyecto final de carrera. Los mismos se encuentran divididos en dos grandes grupos: filtrado de imágenes inválidas y detección de opacidades pulmonares, y se describen en las secciones que siguen.

Dado el tamaño de los *datasets* que componen los diferentes experimentos, que ocupan una gran cantidad de memoria en disco, y dada la necesidad de contar con una placa de procesamiento gráfica (GPU) para el entrenamiento de modelos, se decidió utilizar Google Colaboratory como plataforma de trabajo. Este servicio provee servidores en la nube con acceso a GPU, que permiten el procesamiento en paralelo y por lo tanto, grandes reducciones de tiempo, sin un costo extra. En segundo lugar, teniendo en cuenta el contexto y la necesidad futura de implementación en el marco del hospital, se buscó la interoperabilidad. Es decir, la capacidad de comunicación con los sistemas de información existentes en el HIBA. Esto determinó la elección de Python como lenguaje, y particularmente de Keras como *framework* para el desarrollo de los diferentes modelos.

## 4.1. Filtrado de imágenes inválidas

El objetivo de este apartado es determinar la forma óptima de clasificar distintos tipos de radiografías. Más precisamente, separar radiografías de tórax de radiografías de otras estructuras corporales, y luego clasificar las radiografías de tórax según la vista: PA, AP o L.

Dicho objetivo se encuentra fundamentado principalmente en los flujos clínicos actuales, que resultan en tabulaciones deficientes de las imágenes. Como consecuencia, diferentes sistemas de SSD y *triage* podrían recibir como entrada imágenes que no debieran procesar. A su vez, esto ocasiona una disminución en su rendimiento, lo que termina afectando su utilidad clínica.

En particular, este algoritmo se planea integrar al sistema TRx [26] [27], que fue validado para procesar imágenes de vista PA. En consecuencia, el objetivo final de esta sección es identificar las imágenes PA en contraposición con el resto. Es por ello que esta sección se plantea en términos de filtrado y no simplemente de clasificación. Sin embargo, se decidió mantener la clasificación en las diferentes vistas de tórax de manera de ampliar las potenciales aplicaciones de este primer sistema de filtrado para contribuir a la mejora de otros algoritmos en el hospital.

Escoger una métrica de referencia permite comparar modelos de manera objetiva. La métrica de desempeño utilizada como referencia a lo largo de esta sección es el *macro F1-Score*, ya que las tareas que se realizan son de clasificación multiclase. Para los casos de clasificación en solo dos clases, se utiliza simplemente el *F1-score*.

### 4.1.1. Experimento 1: Cantidad de Etapas de Filtrado

Este primer experimento intenta responder la pregunta de cómo conviene realizar el filtrado: si mediante un único filtro, capaz de distinguir en las clases "PA", "AP", "L" y "otros" (ver Fig. 21) o mediante dos filtros consecutivos, un primer filtro con salidas "tórax" y "otros" y un segundo algoritmo, al cual ingresen las imágenes clasificadas como "tórax" y con salidas "PA", "AP" y "L" (ver Fig. 22).

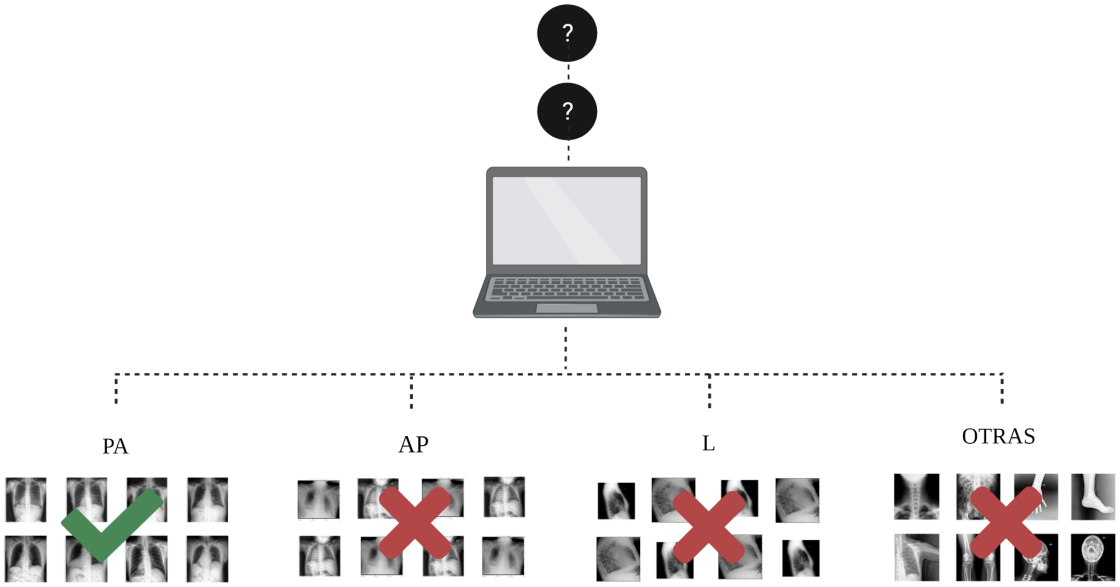


Figura 21: Opción 1: Filtro monoetápico

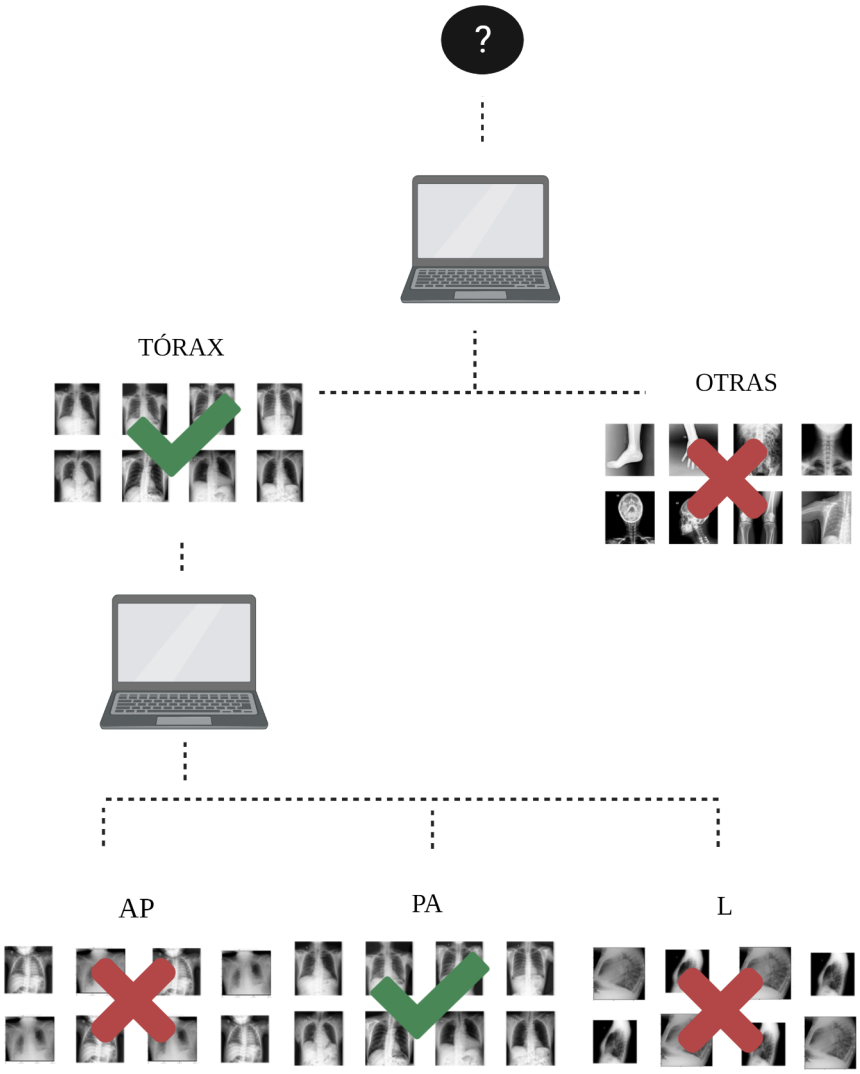


Figura 22: Opción 2: Filtro bietápico

#### 4.1.1.1. Datasets

La interpretación automática de radiografías de tórax ha demostrado ser una herramienta con potencial de mejorar flujos clínicos, priorizando la atención de pacientes según hallazgos radiográficos. A su vez, ha demostrado ser capaz de funcionar como sistemas de soporte a la toma de decisiones ayudando a médicos no especialistas en imágenes. Como consecuencia y para permitir el desarrollo de estos sistemas, se

han liberado públicamente una gran cantidad de *dataset* (DS) con etiquetas que permiten resolver diferentes tareas en este tipo de imágenes médicas. [7] Para la realización de este experimento, se utilizaron múltiples DS combinados de diferentes maneras según el objetivo que se buscaba cumplir.

Las imágenes de tórax se obtuvieron del DS público Padchest [8], que contiene 160861 radiografías de tórax con sus correspondientes metadatos, incluida la proyección de cada radiografía.

Para las imágenes de clase "otros", se utilizó el DS MURA [28], que originalmente se encuentra compuesto por radiografías de miembros superiores divididas en las siguientes clases: "mano", "dedo", "muñeca", "húmero", "codo", "antebrazo" y "hombro".

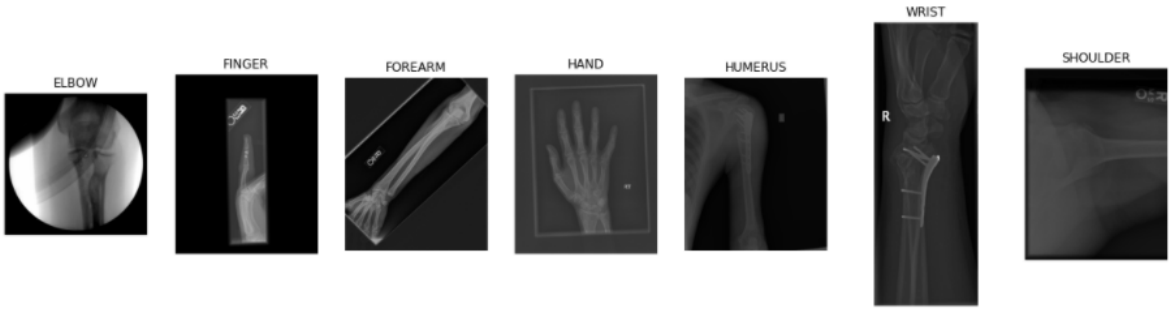


Figura 23: Muestra ds mura

Además, se incluyeron imágenes de un DS público de radiografías de cadera[29]. No fueron incluidas imágenes de otras estructuras corporales debido a imposibilidad de acceso a bases de datos que cumplieran criterios básicos de inclusión, como modalidad de adquisición de imágenes, cantidad mínima de imágenes necesarias y acceso público.


Figura 24: Muestra ds de cadera

Debido a la variedad de orígenes, se desarrollaron dos ETLs distintos con el objetivo de integrar las imágenes en un único DS.

En el caso de Padchest, fue necesario solicitar el acceso a las imágenes. Una vez concedido, se pudo acceder a un repositorio con 52 carpetas comprimidas de 20 GB cada una. Las imágenes en su interior eran de alta resolución (hasta 5000x5000 píxeles), lo que dificultaba su manipulación. Por lo tanto, luego de analizar varias alternativas, se determinó el siguiente flujo: descargar de a una carpeta, descomprimirla localmente, comprimir o submuestrear cada imagen y luego cargar una carpeta con las versiones comprimidas automáticamente a Google Drive (Ver Fig. 25), desde donde se puede acceder utilizando Google Colaboratory.


Figura 25: ETL Padchest

En cambio, en los DS de MURA y de cadera, las imágenes se encontraban en Kaggle. Para estos casos se determinó utilizar la API de Kaggle, extraer las imágenes utilizando Python en Google Colaboratory y desde allí guardarlas en Google Drive.

Una vez disponibles las imágenes en el almacenamiento en la nube, se realizó un preprocesamiento. Debido a la heterogeneidad de cada DS con respecto a la forma de presentación de metadatos, se realizaron preprocesamientos exclusivos para cada uno, que se describen a continuación.

En el caso de Padchest, debido a la gran cantidad de imágenes que componen al DS original (160.000) y a los tiempos consumidos para disponibilizar las imágenes ($\approx$ 4 h por carpeta), se descargaron únicamente las primeras 8 carpetas del DS y se verificó que las proporciones de imágenes por clase fuera similar en cada una de las carpetas descomprimidas, en comparación con el total del DS. Por lo tanto, un primer criterio de inclusión aplicado, es la pertenencia de la imagen a una de las primeras 8 carpetas. Un segundo criterio de inclusión aplicado es la metodología de designación de la proyección de la radiografía. Se incluyeron solo aquellas imágenes cuya proyección había sido designada manualmente por un radiólogo (90 % del DS). En el resto de las imágenes su proyección fue determinada por una red neuronal entrenada con las anotaciones manuales. Esta red distinguía únicamente entre proyecciones PA y L. Como consecuencia, la inclusión de dichas imágenes no representaba una etiqueta de referencia (*Ground Truth*) confiable. Cabe aclarar que no hay que confundir este etiquetado manual de proyección con la asignación de etiquetas diagnósticas, que se realizó a partir de los informes radiológicos (de forma manual en el 27 % de los casos y con minado de texto automático en el 73 % restante).

Un último criterio fue la exclusión de aquellas imágenes pertenecientes a las clases AP con el paciente de pie, costal y *exclude*, ya que la cantidad de imágenes pertenecientes a cada una de estas clases era demasiado baja. Es decir, se incluyeron solo aquellas pertenecientes a las clases AP horizontal, PA y L. Debido a que la clase AP resultó excluida, cualquier referencia que se presente a continuación en este texto a la clase AP, se corresponde con las imágenes pertenecientes a la clase AP horizontal en Padchest. El DS resultante no presenta clases balanceadas (Ver Fig. 26).

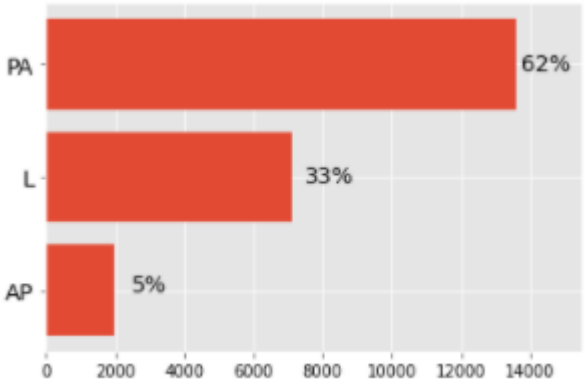


Figura 26: Balance Clases Padchest

Para el armado de la clase “otros”, se decidió que dicha clase representara el 13 % del total del *dataset*. Esta decisión se encuentra fundamentada en datos retrospectivos

del hospital, que se utilizaron para estimar la prevalencia de imágenes de otras partes anatómicas que son procesadas por el sistema TRx a pesar de ser inválidas, ya que no están tabuladas correctamente. Debido a que en todas las clases de MURA y de cadera se contaba con más imágenes que las necesarias, se escogieron aleatoriamente de la totalidad de imágenes disponibles en proporciones iguales.

Finalmente, los tres DS previamente mencionados se integraron y reorganizaron en tres nuevos. El primero, con la totalidad de las imágenes PA, AP, L y otras, para probar un único filtro. El segundo, agrupando todas las imágenes AP, PA y L en una clase llamada "tórax" y por otro lado las imágenes clasificadas como "otras". El tercero, sólo con las imágenes obtenidas de Padchest. Una vez obtenidos los DS a utilizar, se separaron los mismos en *subsets* de *train - validation* (80 %) y de *test* (20 %). Debido a los altos costos computacionales asociados a técnicas de *cross-validation* en modelos de DL, no se implementaron este tipo de técnicas..

#### 4.1.1.2. Entrenamiento de modelos

Buscando que la comparación provenga únicamente de la diferencia entre la utilización del filtro bietápico o monoetápico, los tres modelos se contruyeron a partir de la misma arquitectura, ResNet 50. Además, se mantuvieron constantes los hiperparámetros. Estos, se enuncian en la tabla 1.

Tabla 1: Hiperparámetros Experimento 1

| | |
|---|---|
| **Batch size** | 32 |
| **Epochs** | 30 |
| **Early Stopping** | Si |
| **Patience** | 5 epochs |
| **Learning rate** | $1x10^{-4}$ |
| **Loss Function** | Categorical Cross Entropy |
| **Inicialización de pesos** | Imagenet |

En el caso de la primera etapa del filtro bietápico se cambio la función de activación *softmax* de la última capa utilizada en los otros dos modelos por una función de activación sigmoidea, debido a la naturaleza binaria de la clasificación de esta red.

En los siguientes párrafos se presentan los resultados del experimento. En primer lugar, se presenta para cada uno de los modelos (el filtro monoetápico, y cada etapa del bietápico), la evolución a lo largo de las *epochs* tanto de su *accuracy* como de su *loss*. Estos gráficos permiten por un lado evaluar la elección del hiperparámetro que define la cantidad de *epochs* total, así como también la utilidad del *early stopping* y

la elección de *patience*. A partir de estos, se puede analizar la elección del *learning rate*. Un valor demasiado pequeño mostraría un progreso ínfimo *epoch* a *epoch* y uno demasiado grande generaría resultados oscilantes que incluso podrían llevar a la divergencia. Lo que se busca en este caso es el punto medio, un progreso relativamente rápido para reducir tiempos de procesamiento y recursos computacionales, que converja a valores lo más cercanos a uno en el caso del *accuracy* y más cercanos a cero en el caso de la *loss*. Por último, al superponer el gráfico de *train* con el de *validation*, se busca determinar si existe *overfitting*. Es decir, si el modelo esta siendo capaz de generalizar o no los resultados que aprende con las imágenes de *train* a las imágenes de *validation*. Si las curvas de *train* así como las de *validation* se encuentran prácticamente superpuestas, es posible concluir que no se observa *overfitting* y que el modelo es capaz de generalizar. En cambio, si la curva de *validation* se encuentra por debajo de la de *train* para el caso del *accuracy* y viceversa en el caso del *loss*, implica que el modelo está aprendiendo parámetros tan particulares de las imágenes de entrenamiento que no es capaz de generalizar y funcionar a nivel óptimo cuando se le presentan casos nuevos.

En la Fig. 27 se presenta la evolución de *accuracy* y *loss* para el filtro único. Del mismo se puede concluir que la cantidad de *epochs* escogidas resulta suficiente dada la elección del *learning rate* para garantizar la convergencia del modelo. Por otro lado, el desempeño del modelo varía significativamente en las dos primeras *epochs* y luego si bien sigue progresando lo hace a un ritmo menor. La elección de *patience* parece ser correcta debido a que permite detener el entrenamiento una vez que los resultados no muestran mejoras, sin llegar a agotar la cantidad de *epochs* inicialmente propuestas. Con respecto a la capacidad de generalización del modelo, los gráficos no muestran signos de overfitting, excepto en la *epoch* 11 en la cual las curvas de *train* y *validation* se separan mucho una de la otra. Finalmente, es posible observar que la convergencia se da para valores cercanos a los valores óptimos, tanto de *accuracy* como de *loss*.

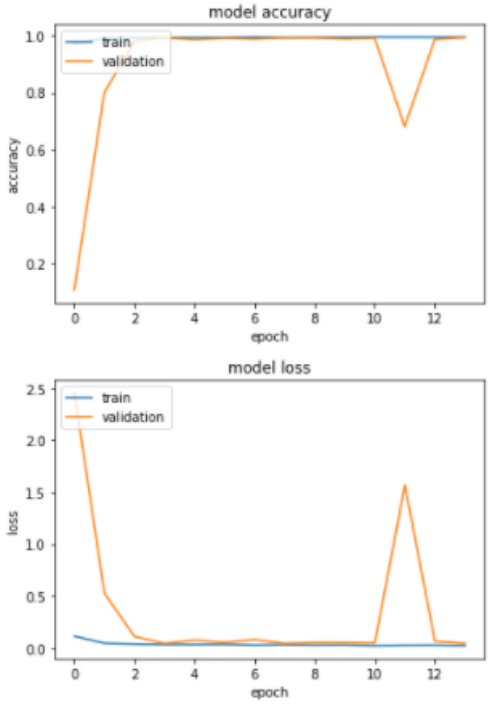


Figura 27: Evolución de Loss y Accuracy para el filtro único

En la Fig. 28 se observan los gráficos correspondientes al entrenamiento del filtro bietápico. En la primera columna se incluyen aquellos correspondientes a la primera etapa, y los de la segunda etapa en la segunda columna. Las conclusiones a las que se llegan en este caso son muy similares a las del modelo de filtro único. Excesivas *epochs*, buena elección de *early stopping*, con su correspondiente *patience*, correcta elección de *learning rate* y una buena capacidad de generalización del modelo a los datos de validación.

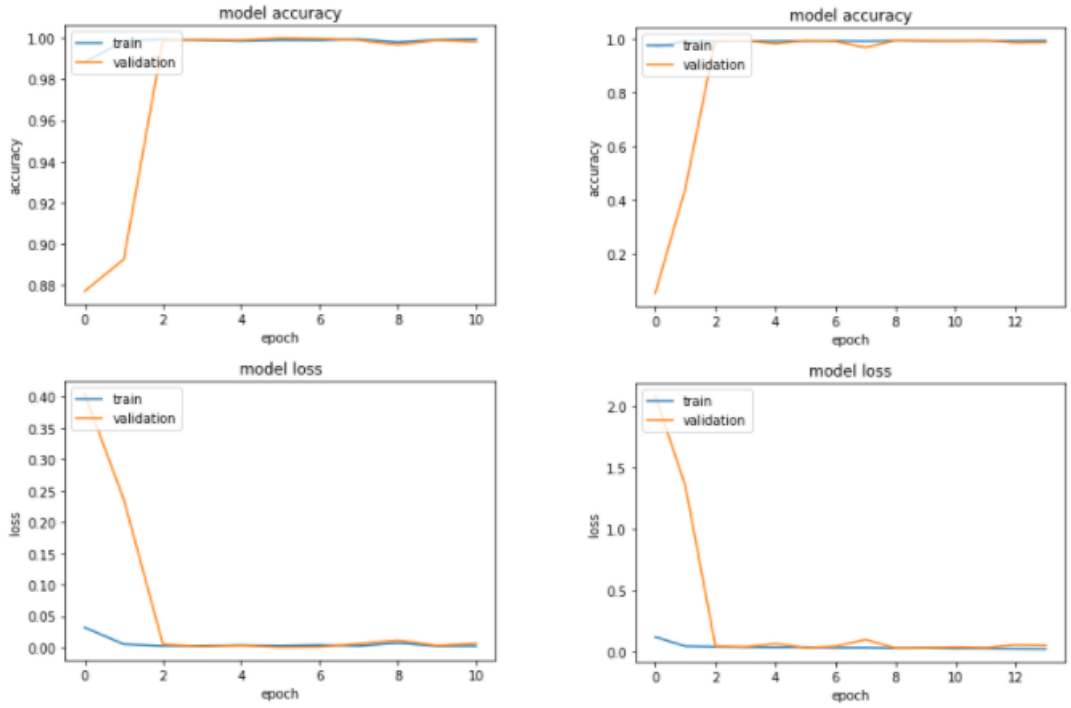


Figura 28: Evolución de *Loss* y *Accuracy* para el filtro Bietápico (primera columna etapa I, segunda columna etapa II)

En la Fig. 29, se puede ver la evolución *epoch* a *epoch* de los valores de *loss* y *accuracy* para cada una de las etapas del modelo. Lo primero que resulta evidente es que la primera etapa del bietápico obtiene una convergencia a resultados mucho menores de *loss*, mayores de *accuracy* y en menor cantidad de *epochs* que los otros dos modelos. En cuanto a los otros dos modelos, evolucionaron en *train* de maneras muy similares, aunque con resultados levemente mejores para el filtro monoetápico. Por lo tanto, pareciera que la discriminación entre diferentes vistas de radiografías de tórax es lo que añade la mayor cuota de complejidad a la tarea propuesta. Sin embargo, los valores obtenidos para los tres modelos se encuentran por debajo de 0.04 para la *loss* y por encima de 0.99 para la *accuracy* para las imágenes de *train*.

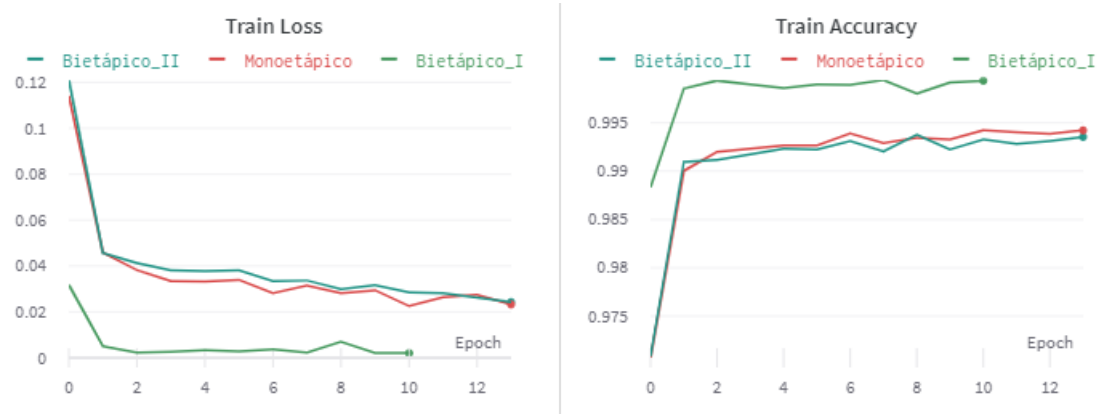


Figura 29: Evolución de Loss y Accuracy en Train para todos los modelos del Experimento 1

En la Fig. 30 se observa el progreso *epoch* a *epoch* de la *validation loss*, es decir, los resultados obtenidos al evaluar la función de costo del modelo en imágenes del grupo *validation* (con las que no fue entrenado). En este caso, al igual que para los gráficos de *train* de la Fig. 29, se observa que la *loss* de la primera etapa del filtro bietápico es menor que la de los otros dos modelos. Sin embargo, en este caso la diferencia resulta mucho menor. Otra diferencia que puede observarse es que la loss que se obtiene para la segunda etapa del filtro bietápico es levemente menor que para el filtro monoetápico. Esto pareciera sugerir que el modelo en dos etapas es mejor que el monoetápico. No obstante, para poder concluir lo anterior, es necesario hacerlo sobre un grupo de datos que no hayan sido utilizados para guiar el entrenamiento. Es decir, con los datos de *test*.

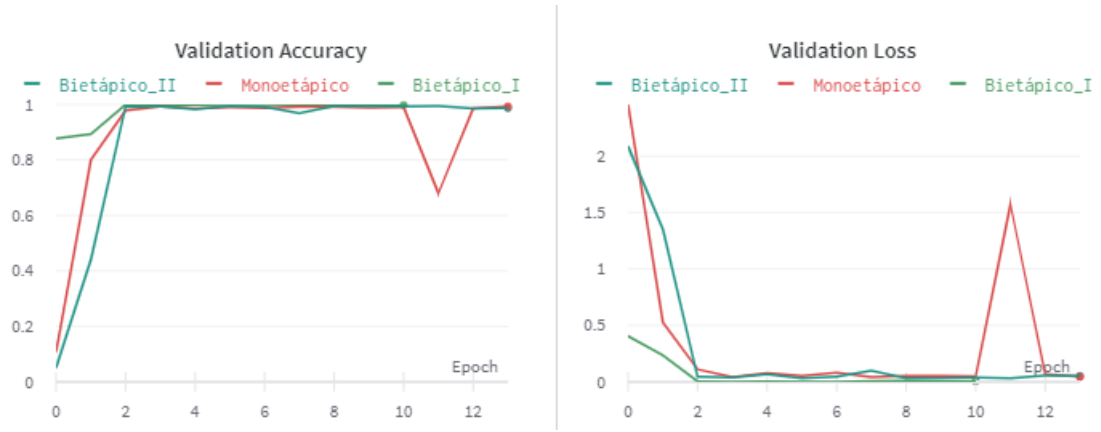


Figura 30: Evolución de Loss y Accuracy en Validation para todos los modelos del Experimento 1

#### 4.1.1.3. Resultados en testeo

**Filtro Bietápico**

Para ilustrar los resultados obtenidos con el *subset* de *test*, se utilizan matrices de confusión con sus correspondientes reportes de clasificación, en los que se presenta la métrica definida para la comparación de modelos (Macro *F1-Score*). Dada la naturaleza del experimento que se describe en este apartado resulta importante aclarar que en este caso se compara la métrica obtenida para el filtro único con la multiplicación de las métricas obtenidas para cada etapa del filtro bietápico. Esto se debe a que se desea evaluar el resultado final de la clasificación y en particular, para el filtro bietápico, se deben atravesar dos etapas de filtrado para encontrarse bien clasificadas mientras que en el filtro único el resultado se obtiene directamente. La matriz de confusión obtenida a la salida de la primera etapa del filtro bietápico se presenta en la Fig. 31, a continuación:

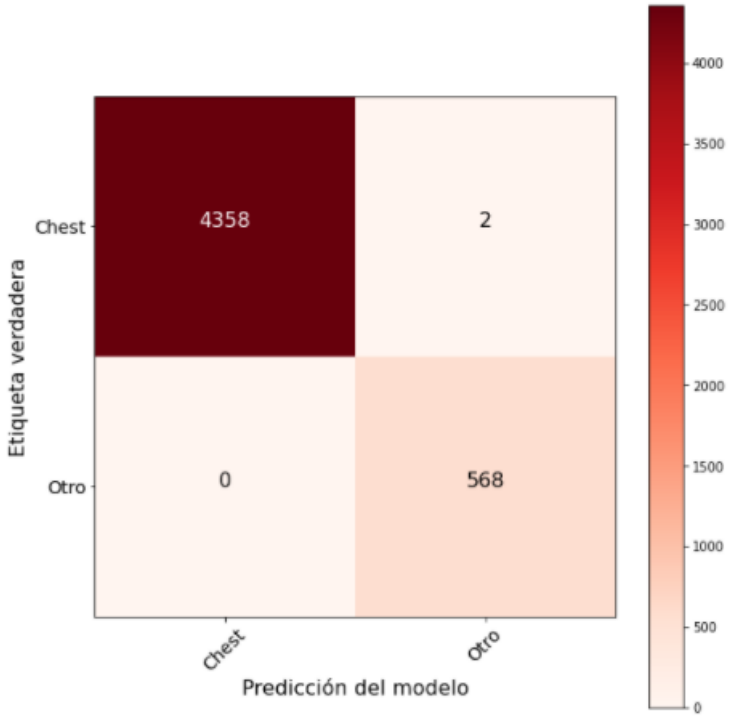


Figura 31: Matriz de Confusión Primera Etapa

La clase "tórax" (*chest*) debe ser interpretada como la que se desea detectar. Por lo tanto, se observan 4358 verdaderos positivos, 568 verdaderos negativos, 2 falsos negativos y ningún falso positivo. Utilizando estos valores se calcularon las métricas*precision*, *recall*, y la métrica elegida para la comparación de modelos, *F1 - Score*.

Estos resultados pueden visualizarse en el reporte de clasificación, que se encuentra a continuación:

Tabla 2: Reporte de Clasificación Primera Etapa

| **Precision** | **Recall** | **F1- Score** |
|---|---|---|
| 1.00 | 1.00 | 1.00 |

En paralelo, se analizaron los resultados incorrectos, y se halló que las dos imágenes falsos negativos se encuentran defectuosas. Ambas radiografías se encuentran poco penetradas. Además, la segunda de ellas se encuentra incompleta (Ver Fig. 32).

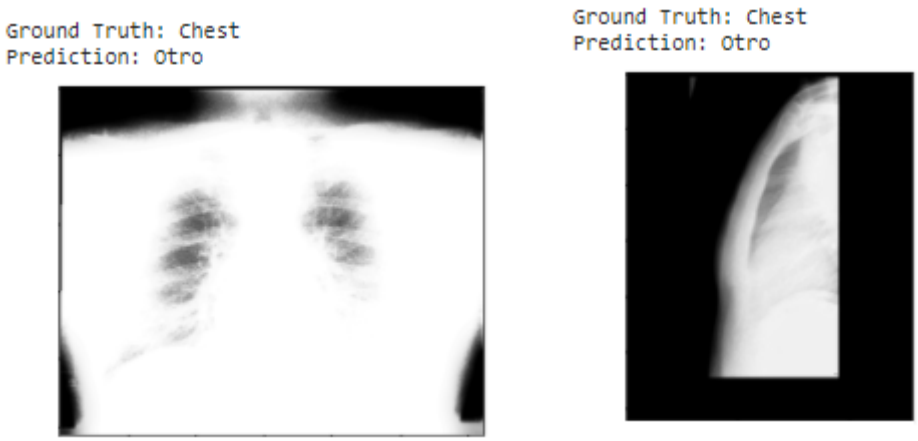


Figura 32: Falsos Negativos Primera Etapa Filtro

Este último análisis lleva casi automáticamente a cuestionar la calidad de datos que componen el DS y resaltan la importancia de utilizar DS de calidad. Así como el sistema fue capaz de clasificar y filtrar estas imágenes mal adquiridas, quizás existen otras correctamente clasificadas, pero que tampoco cumplen con los requisitos de calidad, lo que podría afectar negativamente en el modelo resultante al momento de aplicarlo en la práctica clínica.

Con los resultados obtenidos se debe concluir que la primera parte del sistema es capaz de resolver la tarea para la que fue entrenado prácticamente sin errores. Sin embargo, se plantea el interrogante de la veraciadad de los resultados hallados, como consecuencia de los problemas de calidad detectados en el DS utilizado (Padchest).

Seguidamente, se presentan los resultados correspondientes a la segunda etapa de filtrado. Nuevamente, en primer lugar, se presenta la matriz de confusión, en la que

es posible observar las predicciones del modelo y las etiquetas de referencia para el total de las imágenes.

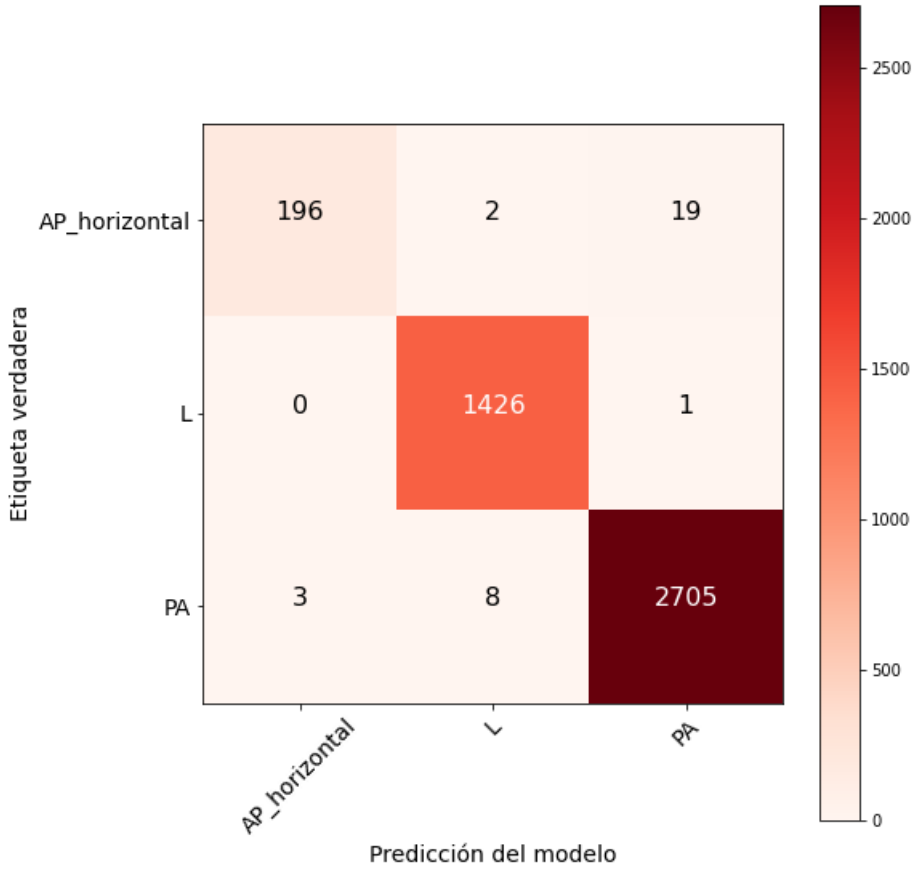


Figura 33: Matriz de Confusión Segunda Etapa

Lo primero que resulta evidente es que en la diagonal principal de la matriz se encuentran la mayor cantidad de imágenes. Esto implica que la mayor cantidad de imágenes fueron clasificadas como correspondientes a su clase original, lo cual habla de una buena capacidad de resolución de la tarea propuesta. Sin embargo, en esta segunda etapa se presentan más errores que en la anterior. De un total de 4360 imágenes utilizadas como *test*, 19 fueron clasificadas como PA, cuando su etiqueta verdadera es AP, y 3 fueron etiquetadas como AP cuando su etiqueta verdadera es PA. Si se observa nuevamente la Fig. 3, cobra sentido el por qué el algoritmo falla al distinguir estas dos clases más que con las laterales. A simple vista se puede ver que las mismas son muy similares, ya que ambas se tratan de radiografías frontales. Incluso para los profesionales de la salud puede no resultar del todo clara esta distinción. Por el contrario, resulta llamativo que muchas imágenes hayan sido

clasificadas como laterales cuando en realidad son PA, debido a que a simple vista estos tipos de imágenes difieren considerablemente una de la otra.

A partir de la matriz de confusión, se confeccionó el reporte de clasificación correspondiente, presentado en la tabla 3.

Tabla 3: Reporte de Clasificación Segunda Etapa

| | **Precision** | **Recall** | **F1 - Score** |
|---|---|---|---|
| **PA** | 0.99 | 1.00 | 0.99 |
| **AP** | 0.98 | 0.90 | 0.94 |
| **L** | 0.99 | 1.00 | 1.00 |
| **Macro Avg** | 0.99 | 0.97 | 0.98 |

Observando los resultados de *precision* y *recall* para todas las clases, resulta evidente que el peor resultado se obtiene para la *recall* de la clase AP. Esto nuevamente deja en evidencia la confusión del modelo para distinguir entre AP y PA, y plantea un nuevo interrogante para definir las etapas del filtrado. Como trabajo a futuro se podría evaluar agregar una etapa más de filtrado (primero separar frontales de laterales, y luego separar las frontales en PA y AP). Otra opción, sería reorganizar las clases del filtro bietápico: en una primera etapa separar tórax frontales, tórax laterales, y otras partes anatómicas, y luego separar las frontales en PA y AP.

A continuación, se exponen ejemplos de las imágenes mal clasificadas por el algoritmo, que se seleccionaron para representar todos los tipos de errores en los que el modelo incurre al momento de realizar la predicción. El primer tipo de error, que es el de mayor prevalencia, es la incorrecta clasificación de imágenes AP como PA. Dos de estos ejemplos son casos pediátricos, que podrían estar fuera de la distribución de imágenes aprendida por el modelo. A futuro, se podrían realizar análisis más profundos sobre la clasificación en este tipo de imágenes, y en caso de hallar que se cometen errores sistemáticos en su clasificación, tener en cuenta esta característica a la hora de entrenar el modelo.

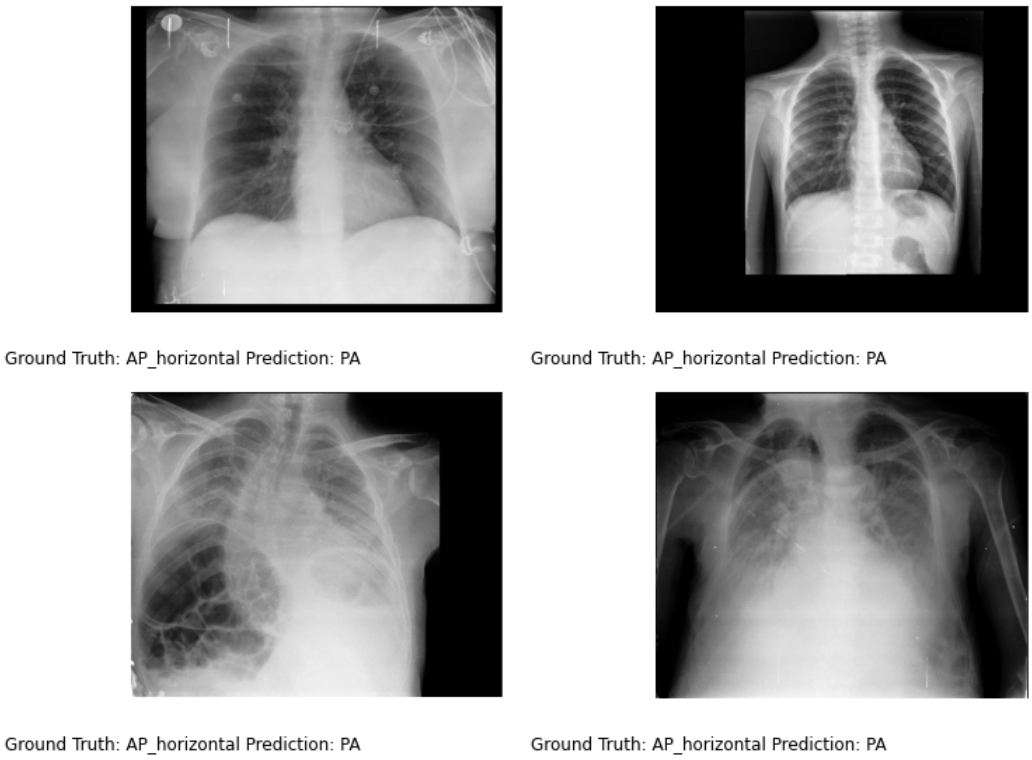


Figura 34: *Ground Truth* AP, Predicción del modelo PA

El error con segunda mayor prevalencia es para aquellas imágenes con GT PA pero que fueron clasificadas como L. En este caso, 7 de las 8 imágenes son efectivamente L, tal como muestra la Fig. 35 (imagen a la izquierda), a pesar de tener PA como etiqueta verdadera. Es decir, el modelo permitió identificar errores de etiquetado en el DS de Padchest. La imagen restante se presenta en la misma figura, a la derecha. Esta se encuentra mal clasificada, y resulta evidente que no es L, pero tampoco es una radiografía correctamente obtenida.

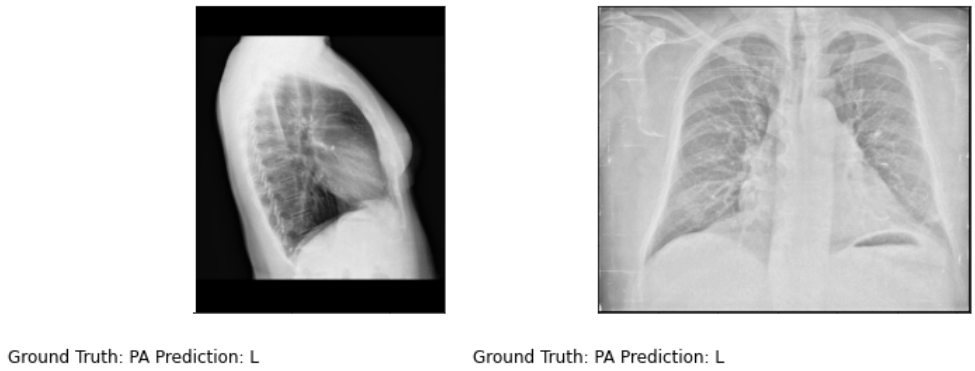


Figura 35: *Ground Truth* PA, Predicción del modelo L

En el siguiente caso, *ground truth* AP, predicción del modelo L, se presentan la totalidad de las imágenes mal clasificadas. Son casos de radiografías poco penetradas, motivo por el cual lucen mucho más blancas que una radiografía normal. Se puede observar que la radiografías prácticamente no permiten distinguir tejido duro de tejido blando y es muy posible que debido a esta característica no se estén clasificando de manera correcta. La mala clasificación de este tipo de imágenes, si bien no implica que el algoritmo falle en la tarea para la que fue diseñado, dejan en evidencia una posibilidad de mejora para el sistema: detectar automáticamente radiografías de mala calidad y apartarlas en una clase propia.

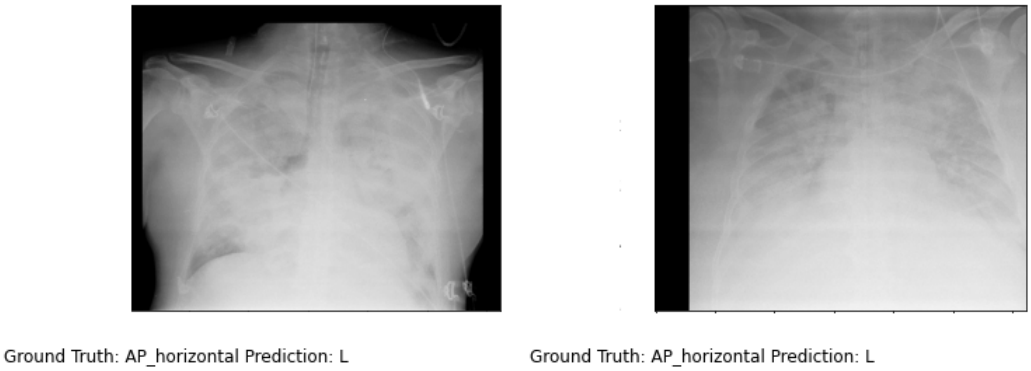


Figura 36: *Ground Truth* AP, Predicción del modelo L

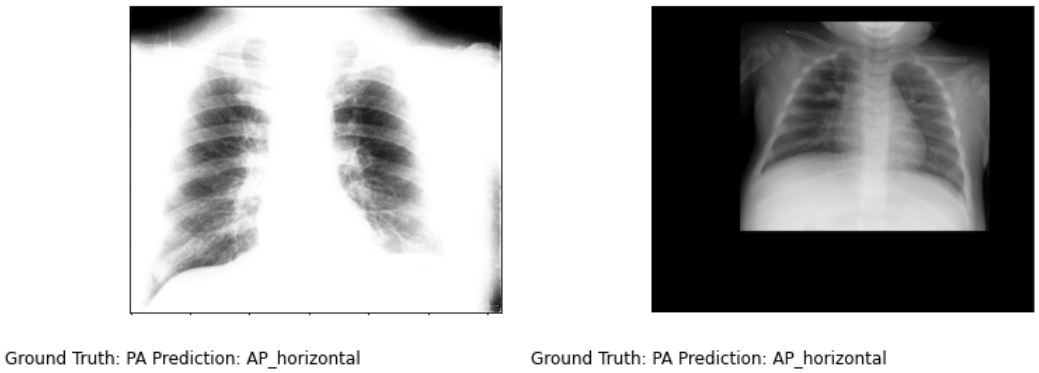


Figura 37: *Ground Truth* PA, Predicción del modelo AP

El último caso se corresponde a la imagen con GT L y predicción PA. En teoría, la imagen que se presenta en la Fig. 38 fue clasificada por profesionales de salud expertos en imágenes como L, pero resulta evidente que es una radiografía frontal. Nuevamente, se suman ejemplos que ponen en duda la calidad del DS de Padchest y por lo tanto la veracidad de los resultados obtenidos.

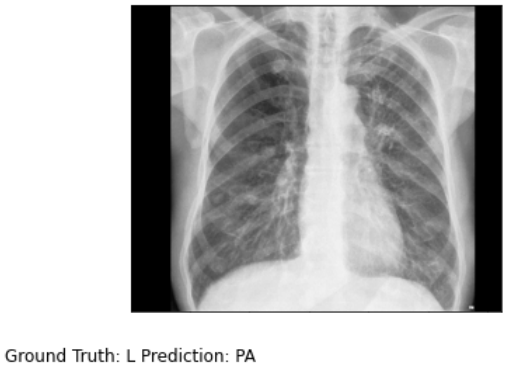


Figura 38: *Ground Truth* L, Predicción del modelo PA

Para finalizar el análisis del filtro bietápico, se calcula el resultado de la métrica para ambas etapas. Esto se muestra en la Ec. 18

$$MFS_{final} = MFS_1 \cdot MFS_2 = 0{,}98 \cdot 1{,}00 = 0{,}98 \tag{18}$$

**Filtro monoetápico**

A continuación se presentan los resultados del *subset* de *test* obtenidos para el filtro monoetápico. En primer lugar, en la Fig. 39 , se presenta la matriz de confusión. Si

se compara el total de imágenes mal clasificadas por cada sistema, el filtro monoetápico comete 77 errores, mientras que el bietápico tan solo 35. Al igual que en la segunda etapa del filtro bietápico, la matriz de confusión deja en evidencia que la mayor debilidad del sistema consiste en la clasificación de imágenes AP y PA. Sin embargo, existe una diferencia entre ambos modelos en la cantidad de imágenes mal clasificadas para este error en particular. Por otro lado, y al igual que en la primera etapa del filtro bietápico, solo dos imágenes pertenecientes a la clase "otros" son incorrectamente clasificadas como pertenecientes a alguna clase de tórax. Esto confirma que esta última tarea tiene un grado de complejidad menor que la clasificación entre las vistas de radiografía frontal.

En el caso del filtro monoetápico, existen 4 clases y 3 niveles distintos de dificultad en la clasificación. En cambio, en el bietápico, la primera etapa se encarga del nivel de dificultad más sencillo, mientras que los otros dos niveles son resueltos por el segundo sistema. Esto sugiere que cuando los grados de complejidad para la clasificación varían entre las distintas clases, un modelo multiclase alcanza menor desempeño que si las tareas de distinta complejidad se procesan con modelos diferentes en forma secuencial.

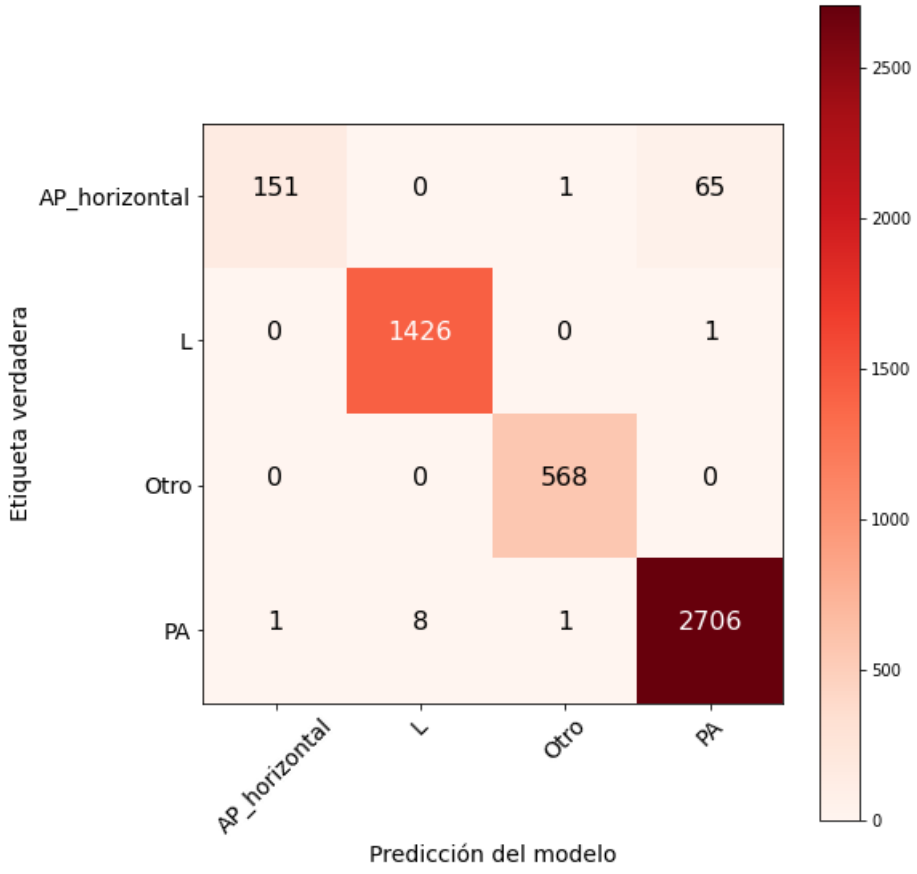


Figura 39: Matriz de Confusión Filtro Único

En el reporte de clasificación (tabla 4) se observa que el *recall* de la clase AP es llamativamente menor que las otras clases. Esto se explica con el hecho de que la clase AP posee muchos falsos negativos, que es análogo a decir que la clase PA posee muchos falsos positivos. Sin embargo, esta cantidad de falsos es una mayor proporción del total de imágenes verdaderamente AP, que del total de imágenes clasificadas como PA. En otras palabras, el desbalance de clases genera que los errores afecten en mayor medida al *recall* de la clase AP que al *precision* de la clase PA.

Tabla 4: Reporte de Clasificación Monoetápico

| | **Precision** | **Recall** | **F1 - Score** |
|---|---|---|---|
| **PA** | 0.98 | 1.00 | 0.99 |
| **AP** | 0.99 | 0.70 | 0.82 |
| **L** | 0.99 | 1.00 | 1.00 |
| **Otro** | 1.00 | 1.00 | 1.00 |
| **Macro Avg** | 0.99 | 0.92 | 0.95 |

Además, en el reporte de clasificación podemos ver directamente varios valores que resumen la *performance* del modelo y entre ellos, el valor de métrica obtenido para este modelo, es decir el macro *F1-score*, que en este caso es 95 %.

Ambos modelos presentan un *F1-Score* de 0.99 para la detección de imágenes PA, que es el objetivo principal de este experimento. El filtro bietápico presenta la desventaja de requerir dos procesos de inferencia para alcanzar la predicción final en una imagen. Sin embargo, teniendo en cuenta las matrices de confusión y el macro *F1-Score*, el filtro bietápico presenta un desempeño superior en la clasificación. Por este motivo, se decidió utilizar el filtro bietápico para el algoritmo integrado.

### 4.1.2. Experimento 2: Mejoras a la Clasificación de Proyecciones

El objetivo que se persigue en este experimento es la optimización del modelo que recibe como entradas imágenes de tórax y como salida las clasifica en AP, PA o L. Para ello, se evaluó la capacidad de tres arquitecturas distintas para realizar esta tarea de clasificación y las variaciones producidas como consecuencia de utilizar o no TL.

En este experimento se utilizó el DS propio creado a partir del de Padchest, descripto en la sección anterior.

Para evaluar el efecto de la arquitectura, se realizaron pruebas con VGG16, Resnet50 e Inception V3. Por otra parte, se comparó el efecto del *transfer learning* mediante la inicialización de los pesos con los valores entrenados para Imagenet (TL) y la inicialización de pesos en valores aleatorios (No TL). El resto de los hiperparámetros del modelo se mantuvieron constantes y se presentan en la tabla 5.

Tabla 5: Hiperparámetros Constantes Experimento 2

| | |
|---|---|
| **Input size** | 128x128 |
| **Batch size** | 32 |
| **Epochs** | 100 |
| **Early Stopping** | Si |
| **Patience** | 10 epochs |
| **Learning rate** | $1x10^{-4}$ |
| **Loss Function** | Categorical Cross - Entropy |
| **Optimizer** | Adam |

Los resultados obtenidos para el *test subset* se resumen la tabla 6, mediante la utilización del *macro F1-score*. Además, con el objetivo de evaluar más en profundidad el efecto de TL, se incluye la *epoch* de la cual fue obtenido dicha métrica.

Tabla 6: F1 Macro Scores obtenidos para cada Arq. con el 100 % del DS

| | **VGG-16** | | **ResNet 50** | | **Inception V3** | |
|---|---|---|---|---|---|---|
| | **F1-Score** | **Epoch** | **F1-Score** | **Epoch** | **F1-Score** | **Epoch** |
| **TL** | 0.99 | 3 | 0.96 | 40 | 0.96 | 3 |
| **No TL** | 0.93 | 99 | 0.98 | 61 | 0.82 | 99 |

A partir de los resultados anteriores, se determinó la utilización del modelo de VGG-16 con el uso de TL. Esto se debe a la métrica superior al resto de los modelos y al hecho de que fueron necesarias solamente 4 *epochs* para alcanzar los valores de pesos finales.

Para analizar más profundamente el modelo, se presenta en la Fig. 40 la matriz de confusión obtenida.

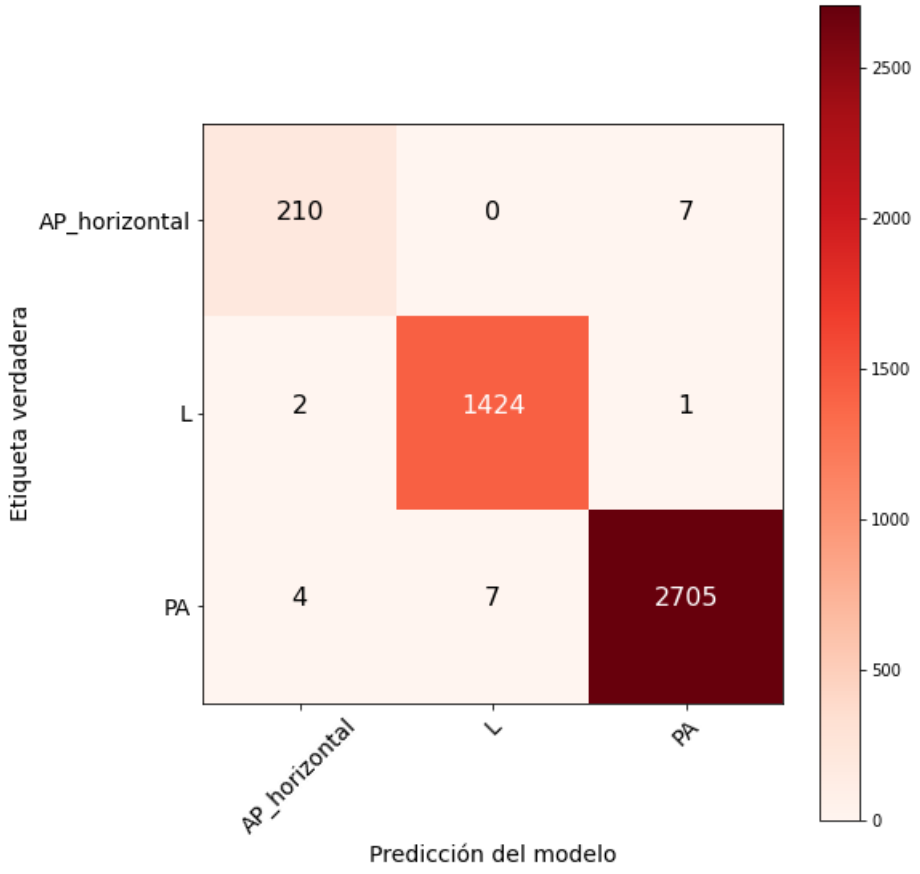


Figura 40: Matriz de Confusión VGG TL

Si se compara esta matriz de confusión con la de la segunda etapa del primer experimento, se puede observar como la cantidad de imágenes AP confundidas con PA disminuyen en más de la mitad.

Volviendo a los resultados de la tabla 6, es posible observar que existe una notable diferencia en términos de métricas entre los resultados obtenidos sin y con TL, tanto para Inception V3 como para VGG-16. En los casos en los que se usa TL, los resultados superan a los que no aplican dicha técnica. Además, en el caso de No TL se reportan los resultados obtenidos en la última *epoch*. Esto es consecuencia de una combinación de *learning rate* demasiado pequeño y cantidad de *epochs* deficientes para permitir la optimización del modelo. Esto puede observarse en la Fig. 41, que presenta la evolución *epoch* a *epoch* en los valores de *accuracy* y *loss* para ambos modelos.

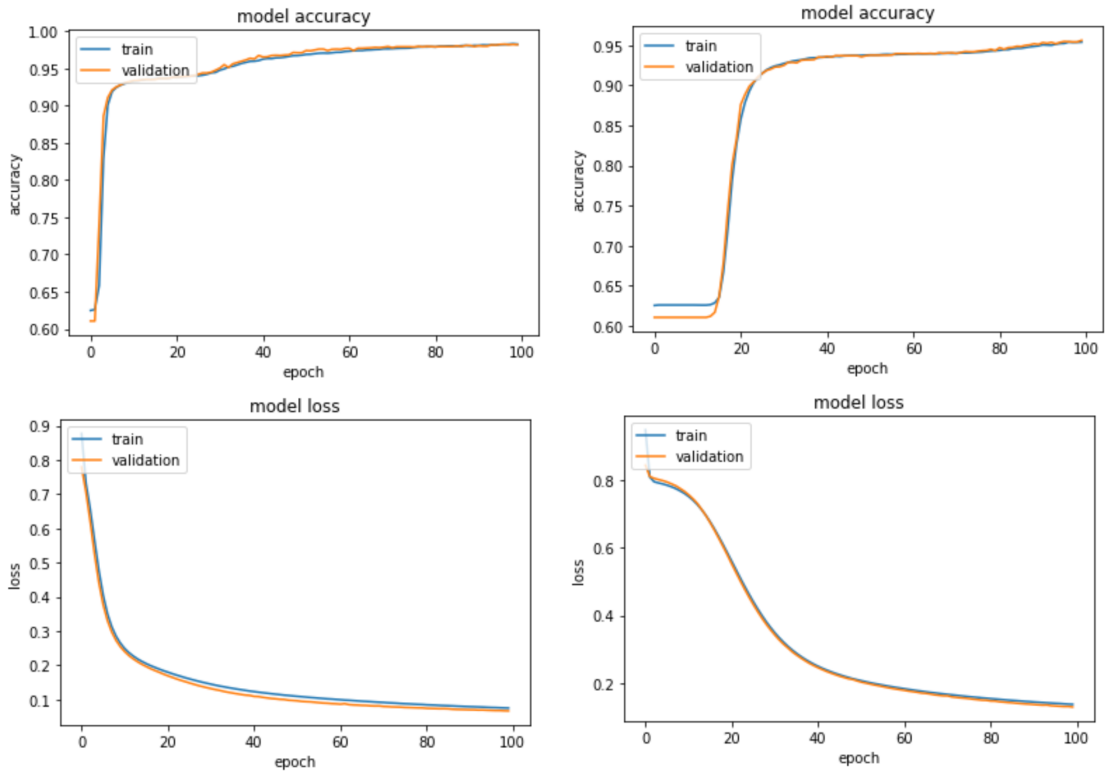


Figura 41: Izquierda: VGG NoTL. Derecha: Inception No TL

Vemos que el salto en desempeño ocurre aproximadamente en las primeas 10 epochs para VGG-16 y en las primeras 20 para Inception. Sin embargo, aún cuando las métricas muestran una tendencia plateau durante el resto del entrenamiento, se observa que continúa habiendo una leve pendiente de mejora. Esto sugiere, por un lado, que continuar los ciclos de entrenamiento podría mejorar aún más el desempeño; y por otro, que una modificación del *learning rate* podría conducir a un mínimo local mejor en la función de costo.

En el caso de ResNet-50 para la inicialización de pesos aleatorios y con los mismos hiperparámetros de entrenamiento, se obtiene un macro *F1-score* propio de una arquitectura que ha llegado a la convergencia en esta tarea. Además, el entrenamiento no utiliza la totalidad de *epochs*, como si es el caso de las otras dos arquitecturas. Esto evidencia la superioridad de las *residual networks* en comparación con las *auxiliary networks* como Inception y con la estructura mucho menos profunda que presenta la arquitectura VGG-16.

Dados los resultados en testeo, se puede concluir que el modelo VGG-16 con TL cumple con el objetivo principal de este experimento.

### 4.1.3. Exprerimento 3: Validación Clasificación HIBA

El tercer experimento realizado en la sección de filtrado es la validación del sistema con imágenes del HIBA. Tal como fue mencionado previamente, uno de los mayores desafíos a la hora de implementar este tipo de algoritmos es la disminución de eficacia en resultados al momento de utilizarlos con datos de un centro de salud distinto al que fue entrenado (*dataset shift*).

Este experimento busca evaluar el funcionamiento de ambas etapas de filtrado con radiografías del hospital pertenecientes a las clases: "AP", "PA", "L" y "otros" (no tórax).

Un total de 106 imágenes fueron disponibilizadas por el HIBA, con sus correspondientes etiquetas asignadas por médicos especialistas en imágenes. Se aplicó el mismo preprocesamiento utilizado para las imágenes de *Test* del DS público: redimensionar a 128x128 y rescalar al rango 0 - 1.

En la Fig. 42, se presentan los resultados obtenidos para el filtro bietápico, comenzando por la matriz de confusión para la primera etapa.

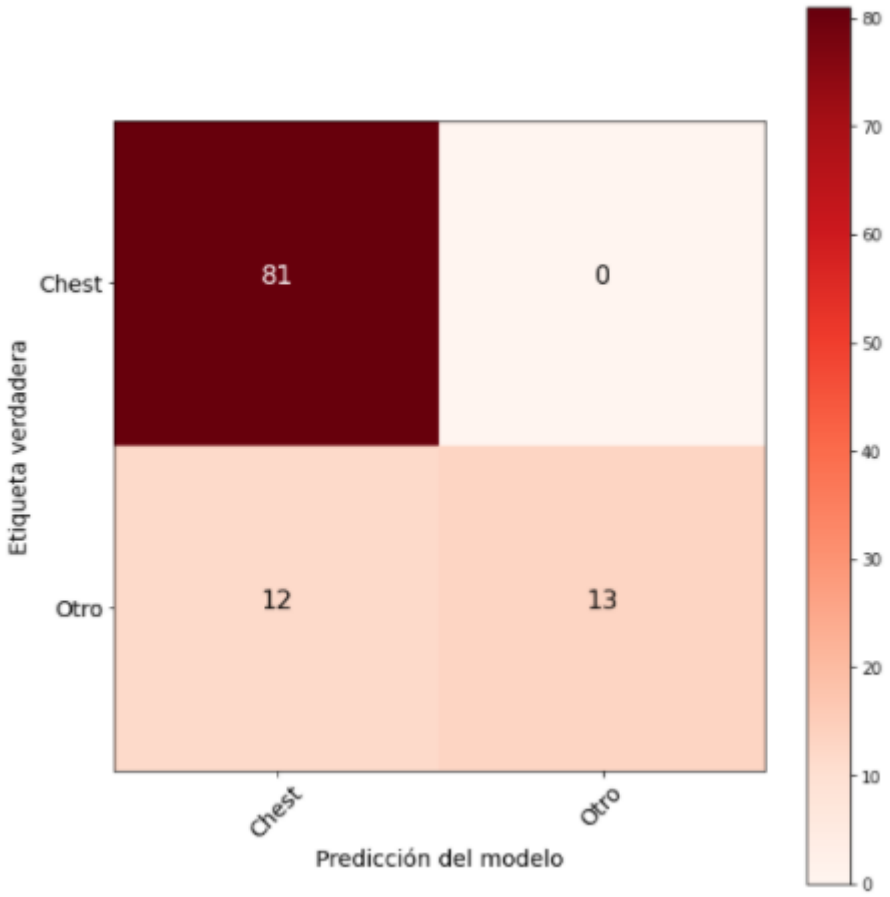


Figura 42: Matriz de Confusión Imágenes HIBA Primera Etapa

A simple vista pareciera que el filtro no se comportó de manera esperada. Existe una gran cantidad de imágenes clasificadas como *chest* con GT otro (falsos positivos). Sin embargo, el modelo no clasificó imágenes de tórax como pertenecientes a otra región anatómica.

Los resultados del reporte de clasificación resumen la matriz de confusión mediante sus correspondientes métricas (Tabla 7). En el mismo es posible observar que la cantidad significativa de falsos positivos afecta la *precision* de la clase "tórax" y como consecuencia el *F1- score* es menor que el del *test* de los experimentos con los DS públicos.

Tabla 7: Reporte de Clasificación Primera Etapa

| **Precision** | **Recall** | **F1 - Score** |
|---|---|---|
| 0.87 | 1.00 | 0.93 |

Buscando explicar la gran cantidad de falsos positivos, se analizaron cualitativamente las imágenes que resultaron mal clasificadas. Se observó que dichas imágenes en realidad eran radiografías de tórax, pero adquiridas con vistas diferentes de las que se usaron para entrenar el sistema, y por lo tanto contaban con la etiqueta de la clase "otros" (Fig. 43 izquierda). Esto evidencia una falencia al momento del armado del DS de entrenamiento, ya que no se consideró esta ambigüedad al momento de definir las clases, y como consecuencia, las imágenes podrían caer en uno u otro grupo indistintamente. Entonces, el error en la clasificación de estas imágenes no parece atribuible a una pobre generalización del modelo, sino justamente lo contrario: la capacidad de generalización parece extenderse a identificar incluso radiografías de otras vistas del tórax.

El único caso que realmente representa un error de clasificación se muestra en la Fig. 43 (derecha). Esta imagen, si bien se encuentra poco penetrada, pertenece a la clase "otros" debido a que se trata de una radiografía de cadera. En conclusión, el sistema demostró un muy buen desempeño, acorde a lo esperado, a excepción del grupo de imágenes que fue mal clasificado como consecuencia de una deficiente planificación del experimento inicial. A futuro, sería interesante reconsiderar en qué categoría incluir a este grupo de imágenes.

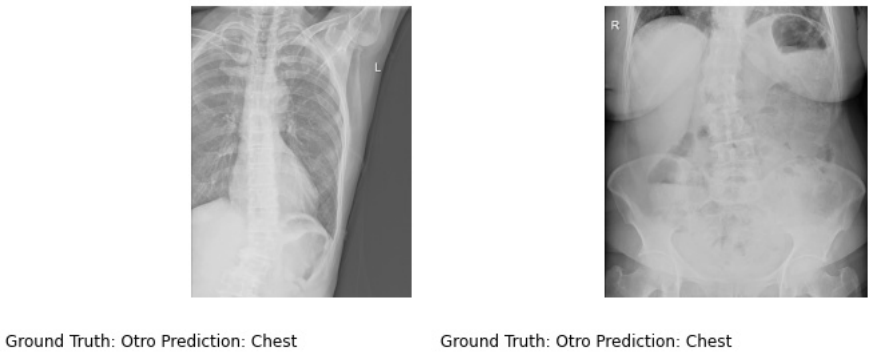


Figura 43: Imágenes Mal Clasificadas HIBA

Para evaluar la segunda etapa de filtrado se quitaron las imágenes pertenecientes a la clase "otros", con el objetivo de evaluar con imágenes del hospital el funcionamiento de este filtro en particular. En la tabla 8 se resume el reporte de clasificación del modelo, en el que se observa un *Macro F1 - score* de 0.83. Nuevamente, este valor

es considerablemente menor al esperado. En particular, se observa un *recall* de la clase AP muy por debajo del valor obtenido en el *test* de los DS públicos.

Tabla 8: Reporte de Clasificación AP, PA y L

| | **Precision** | **Recall** | **F1 - Score** |
|---|---|---|---|
| **AP** | 0.90 | 0.45 | 0.60 |
| **PA** | 0.81 | 0.98 | 0.89 |
| **L** | 1.00 | 1.00 | 1.00 |
| **Macro Avg** | 0.90 | 0.81 | 0.83 |

En la Fig. 44, se observan los resultados de la matriz de confusión del modelo. La clase AP posee una gran cantidad de falsos negativos, mientras que el resto de las imágenes fueron clasificadas casi en su totalidad en la clase a la que corresponden.

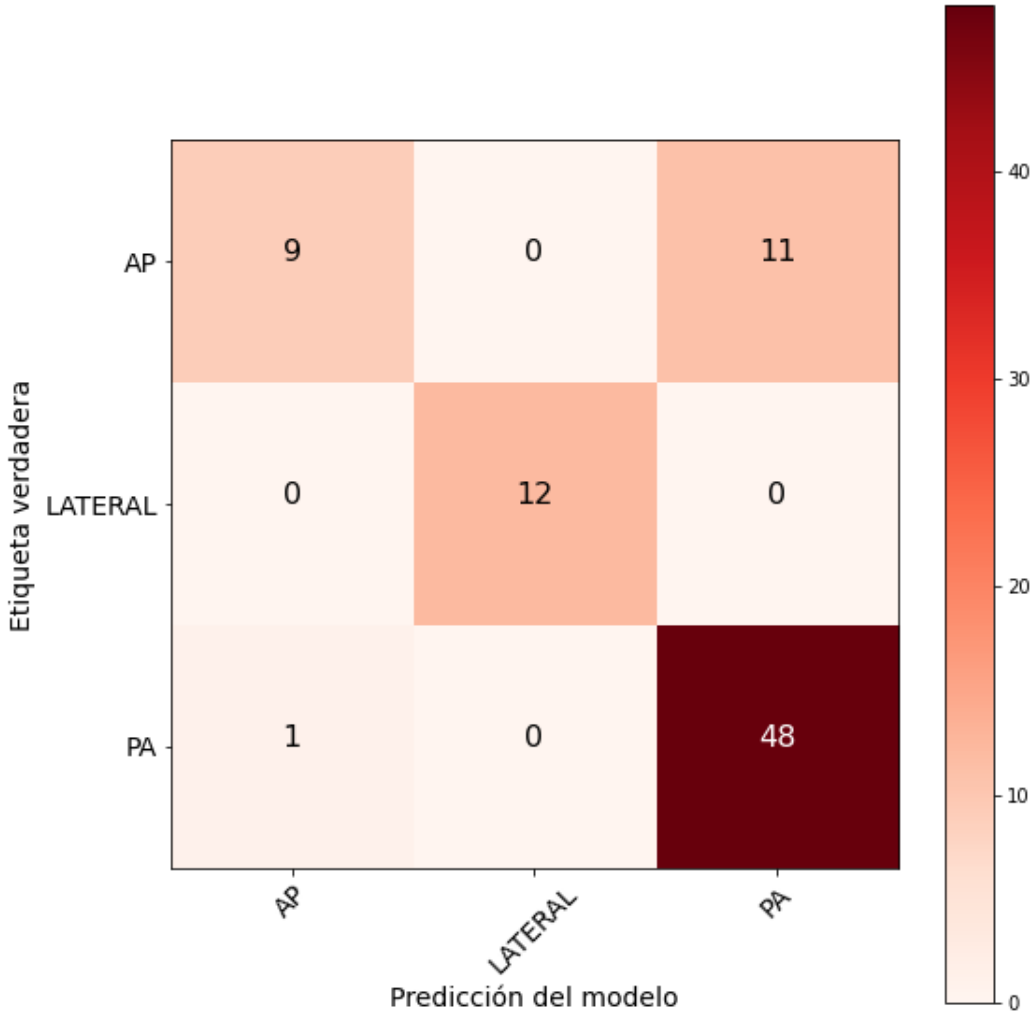


Figura 44: Matriz de Confusión Imágenes HIBA Imágenes AP, PA y L

La idea del filtro bietápico es procesar las imágenes en forma secuencial, de modo que sólo aquellas radiografías clasificadas como "tórax" por el primer modelo sean luego clasificadas por el segundo modelo. Con el objetivo de evaluar el comportamiento de este segundo modelo en el caso en que el filtro inicial falle, se clasificaron las imágenes pertenecientes a la clase "otros" mal clasificadas por la primera etapa. Los resultados obtenidos se presentan en la Fig. 45 como proporción de imágenes asignadas a cada una de las clases. Teniendo en cuenta el objetivo inicial del filtro, que es detectar imágenes PA y separarlas del resto, el sistema no comete una gran cantidad de errores. La mayor parte de las imágenes que no son de tórax son clasificadas como L.

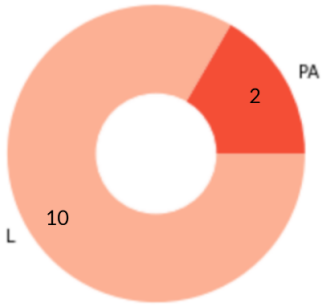


Figura 45: Proporción de imágenes otras clasificadas por la segunda etapa del filtro

Otra característica que decidió evaluar es la salida de la función *softmax* para cada una de estas imágenes. La mediana de dicho valor es 1. Esto implica que el modelo asigna las clases con una certeza absoluta en al menos la mitad de los casos, aunque las imágenes no pertenecen a ninguna de las clases con las que fue entrenado. Esta característica indeseable del modelo se conoce como mala calibración. [30]

En conclusión, se obtuvo un macro *F1-score* global de 0.77. Sin embargo, el sistema se comportó acorde a lo esperado con las imágenes para las que fue entrenado, lo que demostró una buena capacidad de generalización del modelo. Como consecuencia de esta buena capacidad de generalización y una planificación que no tuvo en cuenta imágenes de tórax no pertenecientes a las vistas principales, las métricas del modelo se vieron considerablemente reducidas. A futuro, sería interesante entrenar el modelo nuevamente, incluyendo este tipo de imágenes en la clase "otros".

## 4.2. Opacidades pulmonares

La segunda parte de este proyecto consiste en la creación de un sistema capaz de detectar opacidades pulmonares en radiografías frontales PA de tórax y agrupar las imágenes en dos clases: "opacidades pulmonares" y "sin hallazgos", con el objetivo de formar parte de un sistema de soporte a la toma de decisiones.

Esta sección del proyecto describe el uso de YOLO v5 para la tarea de detección de objetos, combinado con un modelo de regresión logística para obtener una clasificación binaria capaz de funcionar como sistema de soporte a la toma de decisiones.

Los datos utilizados para el entrenamiento inicial fueron obtenidos del *dataset* público conocido como VinDr-CXR[10]. Una vez obtenido un modelo confiable y evaluado en un *test set* previamente definido de este DS, se validaron los resultados con imágenes provenientes del HIBA previamente anotado por personal calificado. Luego,

se entrenó el modelo de regresión logística. En este caso, se utilizó como DS de *train* a los resultados obtenidos como *test* en la primera etapa y como *test* a los datos de validación ofrecidos por *HIBA*.

### 4.2.1. Experimento 4: Detección de opacidades pulmonares

El objetivo de este experimento es el armado del sistema de detección de opacidades pulmonares y su optimización. Es decir, entrenar un modelo de redes neuronales convolucionales capaz de clasificar imágenes en cuanto a presencia o ausencia de opacidades pulmonares y, en caso de determinar presencia, señalar su ubicación en las imágenes mediante el posicionamiento de una BB.

El DS utilizado en esta sección fue obtenido de VinDr-CXR[10]. Más precisamente, se utilizó el *train set* como *dataset* total, debido a que el *test set* aún no se encontraba disponible, por liberarse como parte de una competencia de Kaggle. Las imágenes se disponibilizaron en Google Drive gracias al uso de la API de Kaggle.

El DS de *train* consiste en 15.000 imágenes, cada una de ellas anotada por 3 de un total de 17 radiólogos, todos ellos experimentados en el diagnóstico mediante el uso de radiografías. Los hallazgos se encuentran clasificados en 14 clases, y los mismos se distribuyen tal como se observa en la Fig. 46.

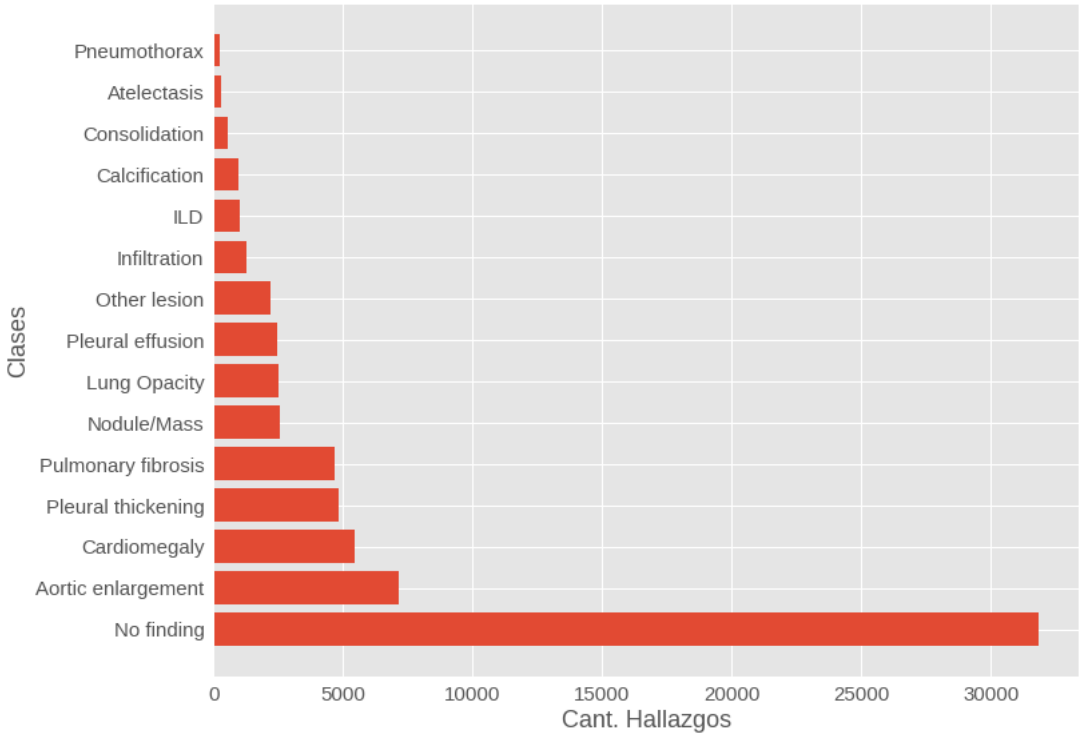


Figura 46: Cantidad de Hallazgos por clase que componen el DS de *train* de VinDr-CXR, utilizado para este proyecto como DS total

Sin embargo, no todas las imágenes que componen al DS de *train* fueron utilizadas. Debido a que el objetivo perseguido consiste en detectar opacidades pulmonares en su concepto amplio, tomando como base las definiciones presentadas en la sección 3.1.3, se seleccionaron las imágenes con etiquetas positivas para las clases: atelectasia, consolidación, opacidad pulmonar y nódulo-masa. Luego, se agruparon las mismas bajo una única clase denominada "opacidad pulmonar" (Ver Fig. 47).

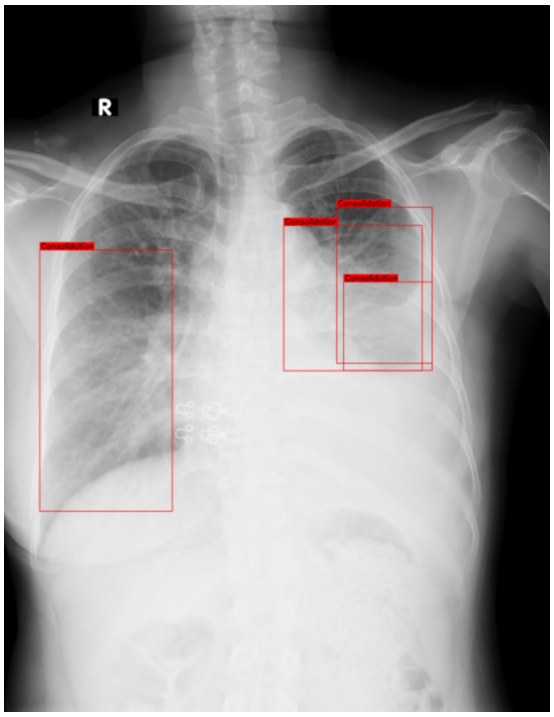


Figura 47: Detección Opacidades Pulmonares

Una vez seleccionadas todas las imágenes disponibles en el *dataset* pertenecientes a dichas clases, se agregó una proporción del 5 % de imágenes perteneciente a la clase "sin hallazgos", tomando parámetros obtenidos con óptimos resultados en la literatura [31]. Esto resultó en un DS total de 1851 imágenes perteneciendo a la clase amplia denominada "opacidades pulmonares", con un total de 5898 hallazgos (BB), y 93 imágenes sin hallazgos. La cantidad de imágenes utilizadas se encuentra dentro de los valores recomendados. En cambio, la cantidad de instancias (objetos detectados) recomendadas para entrenar una red YOLO es de 10.000 [31]. Para intentar subsanar la falta de objetos anotados necesarios, se implementaron técnicas de aumentación de datos y TL.

A partir de las imágenes seleccionadas para este experimento y sus correspondientes anotaciones se realizó un análisis exploratorio. El mismo arrojó los gráficos que se presentan en la Fig. 48, en los que se puede observar la distribución de las posiciones normalizadas de los centros de las BBs (izquierda) y la distribución de sus dimensiones normalizadas (derecha). A partir de dichos gráficos, es posible inferir que los centros de las detecciones se sitúan sobre los pulmones, debido a que se trata de imágenes tipo PA y que las hay de varios tamaños, con mayor amplitud en altura que en ancho, lo cual también es coincidente con la forma de los pulmones en imágenes PA y por lo tanto con las regiones en las que se deberían detectar opacidades.

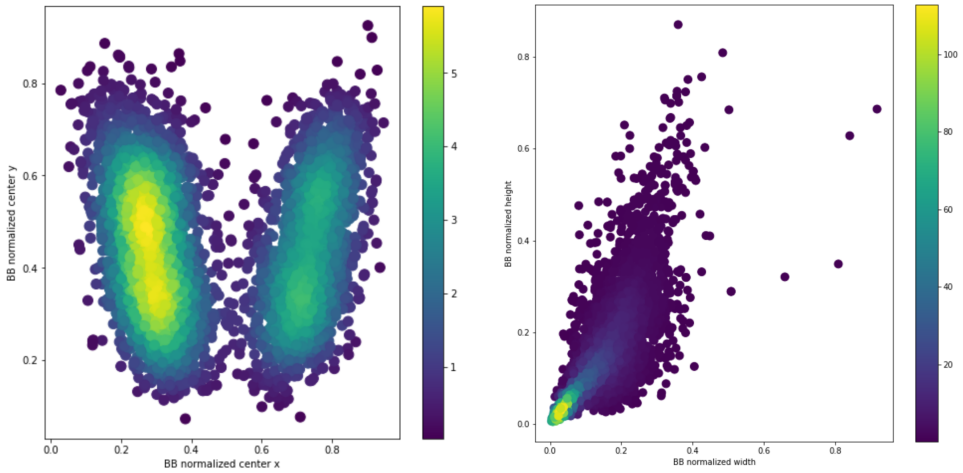


Figura 48: Distribución de la posición normalizada de los centros de las BBs (izquierda). Distribución de las dimensiones normalizadas de las BBs (derecha)

Otra característica que fue tenida en cuenta para el análisis exploratorio del DS, fue la cantidad de instancias de BBs que poseían las diferentes imágenes. En la Fig. 49, se observa dicha distribución. Tanto en el *box-plot* como en el histograma puede observarse una asimetría derecha, es decir, que existen una gran cantidad de imágenes que poseen muy pocas instancias y a medida que aumenta el número de instancias, la cantidad de imágenes que las poseen son cada vez menos. En particular, el *box-plot* evidencia que el valor de la mediana es 2, mientras que el rango se extiende de 1 hasta 55. Esto implica, tal como se puede observar en el gráfico, la existencia de una gran cantidad de *outliers*, es decir, imágenes cuya cantidad de instancias es 1.5 veces mayor que la mediana.

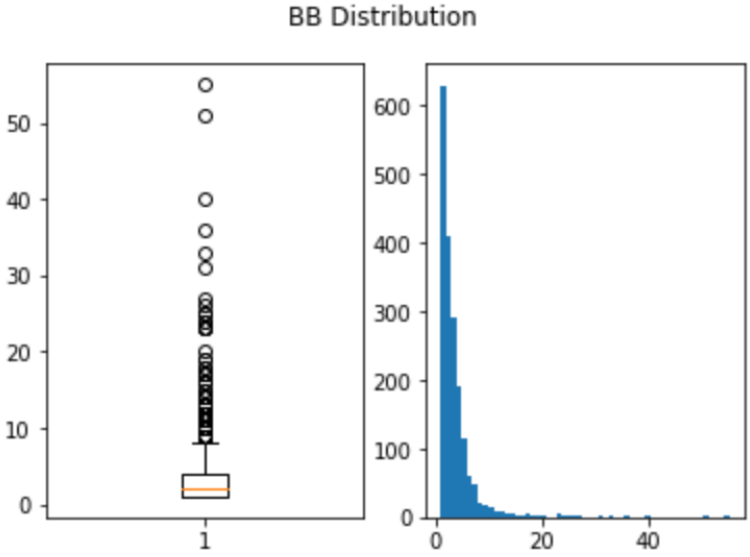


Figura 49: Distribución de instancias de BBs detectadas. *Box-Plot* (izquierda). Histograma (derecha).

Debido a que las imágenes diferían considerablemente en la cantidad de instancias de objetos a detectar, no se dividieron las imágenes en *train* y *test* de manera aleatoria, sino que se buscó asignar imágenes tanto con muchas como con pocas instancias para ambos *subsets*. Para cumplir con este objetivo y teniendo en cuenta la no normalidad de la distribución, se utilizó la mediana de instancias de objetos por imágenes para separar las mismas en dos grupos. Luego, se formaron los *subsets* de *train* y *test* con proporciones similares de ambos grupos.

En cuanto a la arquitectura utilizada, YOLO V5, se tomó la implementación de Ultralytics [31]. Este repositorio brinda una variedad de modelos con distinta cantidad de parámetros entrenables, lo que termina resultando en diferente velocidad de procesamiento y niveles de *performance*. En este caso, se escogió el modelo YOLO V5m y se usaron los pesos preentrenados con el DS de COCO.[32]

En cuanto a los hiperparámetros seleccionados, se utilizaron los sugeridos por la documentación de Ultralytics [31]. Se comenzó entrenando con 300 *epochs*. Debido a que con dicha cantidad de *epochs* se observó *overfitting*, en las siguientes corridas se disminuyó dicha cantidad. Las técnicas de aumentación de datos escogidas inicialmente fueron: traslación, cambios de escala, y reflexión con respecto al eje y. Además, se configuró un umbral de IoU de 0.6 para la determinación de la matriz de confusión.

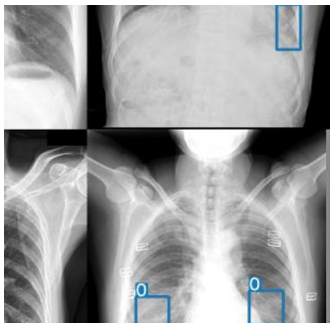

Figura 50: MOSAIQUISMO

Por último, y con el objetivo de comparar resultados, se agregó para la tercera corrida una técnica adicional de aumentación de datos. Esta técnica es conocida como mosaiquismo y se basa en partir a las imágenes en mosaicos y armar nuevas a partir de la unión aleatoria de estos mosaicos (Ver Fig. 50). Esta técnica en principio no había sido implementada porque se tuvo en cuenta que las imágenes radiográficas en general se obtienen de maneras preestablecidas, fijando la posición del paciente con respecto al tubo de rayos y al detector. Sin embargo, en los resultados se observó una mejora en la detección como consecuencia de esta implementación, por lo que se decidió escoger el modelo correspondiente a esta corrida como modelo definitivo (Ver Fig. 51).

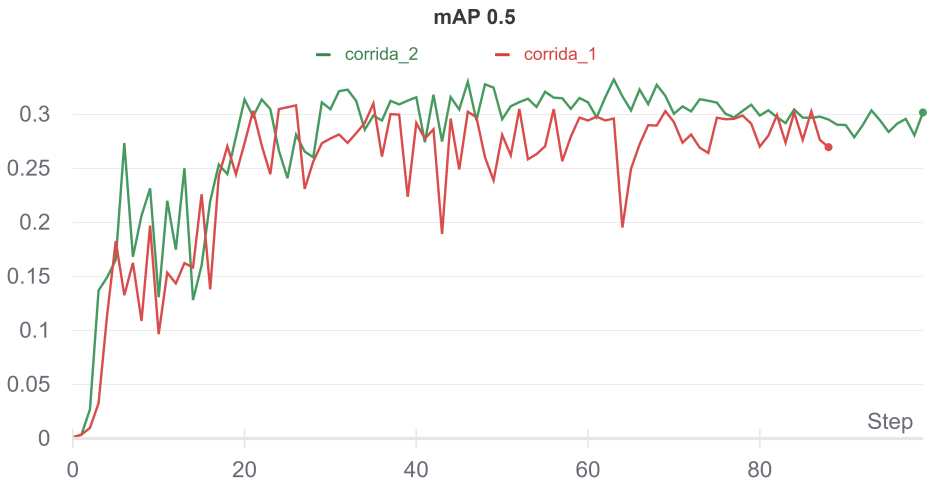


Figura 51: MAP 0.5 EPOCH A EPOCH. LA CORRIDA 1 (ROJO) FUE REALIZADA SIN MOSAIQUISMO, MIENTRAS QUE LA CORRIDA 2 (VERDE) CON MOSAIQUISMO.

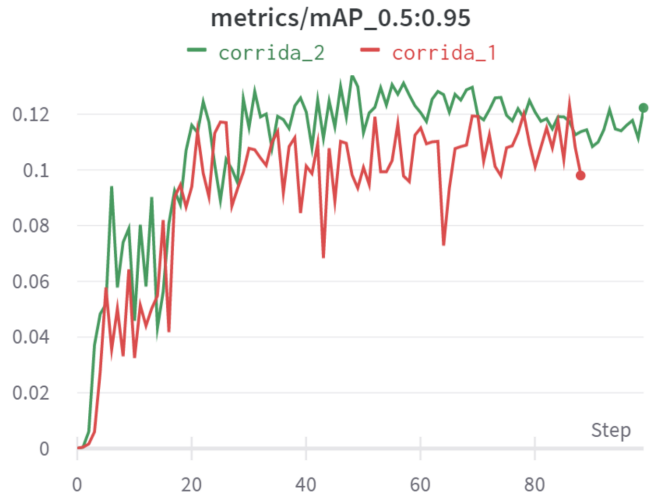


Figura 52: mAP para umbrales de IoU entre 0.5 y 0.95 en el conjunto de validación para las corridas 1 y 2

En la Fig. 52 se muestra el mAP para los diferentes umbrales desde 0.5 a 0.95 en el conjunto de validación. Este gráfico sugiere que la aumentación de datos (curva verde) mejora el desempeño independientemente del umbral de IoU utilizado con respecto a un entrenamiento sin aumentación de datos (curva roja).

Previo a escoger una de las corridas como modelo definitivo, se observaron a su vez los resultados en el *subset* de *test*. En la tabla 9, se presentan los mAP con IoU umbral de 0.5 para la detección de opacidades pulmonares, los cuales también muestran superioridad de la corrida con uso de mosaiquismo respecto a la corrida sin el uso de esta técnica de aumentación de datos.

Tabla 9: Resultados en testeo para la detección de opacidades pulmonares

| | **mAP 0.5** |
|---|---|
| **Sin Mosaiquismo** | 0.27 |
| **Con Mosaiquismo** | 0.34 |

Finalmente, con el objetivo de evaluar más en profundidad los resultados obtenidos, se generó para los datos en *test* la distribución de predicciones de centros de BBs y de dimensiones. Los resultados se presentan en la Fig. 53, en donde es posible observar una distribución muy similar al a obtenida para los BBs originales del DS completo (Fig. 48).

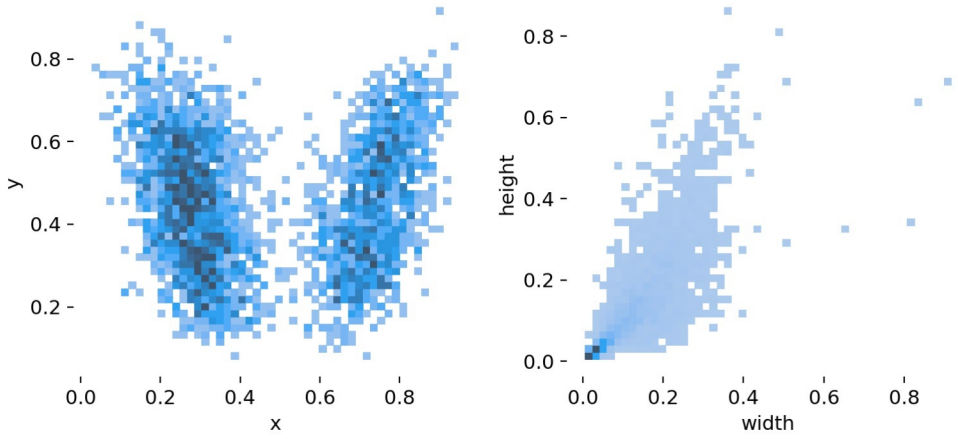


Figura 53: Gráfico de dispersión de la distribución normalizada de centros de BBs detectados por el modelo (izquierda). Gráfico de dispersión de las dimensiones normalizadas de los BBs detectados (derecha)

Los resultados obtenidos en este experimento si bien son buenos para una tarea de detección de imágenes, se encuentran alejados del estado del arte para detección de objetos en el DS COCO (mAP=0.64) [31]. Sin embargo, debe tenerse en cuenta que las detecciones de objetos cotidianos presentan menos variabilidad que las detecciones de opacidades pulmonares en radiografías de tórax. Esto se debe, entre otras cosas, a la variedad de criterios al momento de la detección por profesionales capacitados. Como consecuencia, es esperable que este tipo de tareas sean consideradas bien resueltas con un indicador menor al estado del arte para objetos cotidianos. Además, debe tenerse en cuenta que debido a una deficiente cantidad de imágenes e instancias de detección de objetos disponibles en DS públicos, resultó necesario tomar todas las detecciones disponibles y agruparlas bajo una misma clase, incorporando en un mismo grupo hallazgos con cierta heterogeneidad. Aún así, la cantidad de instancias de objetos necesarias, es aproximadamente mitad de la propuesta por la literatura. En conclusión, dadas las características de la tarea a resolver y de los datos disponibles, un mAP de 0.34 en testeo indica un buen entrenamiento del modelo.

### 4.2.2. Experimento 5: Validación Detección de Opacidades Pulmonares HIBA

El objetivo de este experimento es validar los resultados obtenidos en el experimento anterior con imágenes del hospital, a fin de analizar la factibilidad de implementación del sistema, teniendo cuenta que los sistemas de inteligencia artificial en salud suelen fallar al ser implementados en centros distintos a los que se tomaron las imágenes.

El HIBA disponibilizó para esta etapa un total de 1330 imágenes. Asociada a cada imagen, se disponibilizó una máscara, que indicaba píxel a píxel la presencia de algún tipo de consolidación dividida en clases. Es decir, se presentaron metadatos como para realizar un entrenamiento supervisado de una tarea de segmentación. Por lo tanto, se procedió a traducir cada una de las máscaras en BBs, tomando como límites de las *bounding boxes* los píxeles más alejados en cada eje correspondientes a la máscara (Ver Fig. 54). A su vez, todas las clases de hallazgos: consolidaciones, nódulos masa, atelectasia, patrón intersticial y lesiones de la pared, se agruparon bajo la etiqueta general con la cual fueron entrenados de "opacidades pulmonares". Esta clase general quedo compuesta por un total de 25 hallazgos del tipo atelectasia, 64 consolidaciones, 34 lesiones de la pared, 98 nódulos-masa y 154 patrones intersticiales. Por último, se escribieron las etiquetas de cada imagen, acorde a la entrada necesaria para el sistema.

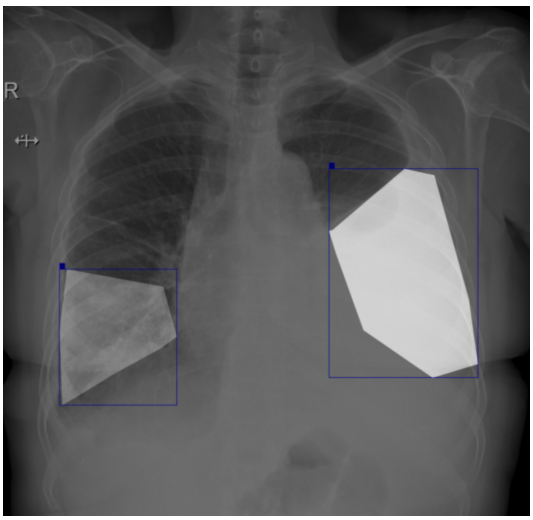


Figura 54: Traducción de etiquetas de tarea de segmentación a detección de objetos

Una vez realizados estos pasos de preprocesamiento, se corrió el modelo sobre este grupo de imágenes y se obtuvo un mAP 0.5 de 0.21. Comparando el mAP del DS de validación del hospital con los obtenidos en la etapa de *test* del DS público, se observa una desmejora en el rendimiento del modelo.

Sin embargo, es importante notar que las imágenes agrupadas como opacidad pulmonar por parte del hospital, incluyen subclases como patrón intersticial y lesiones de la pared, las cuales no fueron utilizadas para el entrenamiento inicial del modelo.

Esto podría estar ocasionando la disminución en el rendimiento. Para probar la anterior hipótesis, se confeccionó un DS de validación reducido (*reduced*), en el cual se quitaron los hallazgos pertenecientes a dichas clases y se corrió nuevamente el modelo. En este caso, el mAP obtenido fue de 0.31, un valor mucho más cercano al del DS de *test* original. A partir de este resultado, se determinó continuar únicamente con las imágenes para las cuales la red neuronal fue entrenada.

A continuación, se presentan los resultados en forma de imágenes, con sus correspondientes detecciones en forma de BBs. En color azul se presentan los resultados del modelo propuesto con su correspondiente nivel de confianza, y en color rojo las detecciones dadas por el hospital como GT. Por motivos visuales, en las imágenes se muestran aquellas BB con un nivel de confianza mayor a 0.4.

En la Fig. 55, se presentan imágenes pertenecientes a la clase "sin hallazgos" y clasificadas por el modelo como tal, motivo por el cual existe ausencia de BBs de ambos colores. Estas imágenes representan los verdaderos negativos.

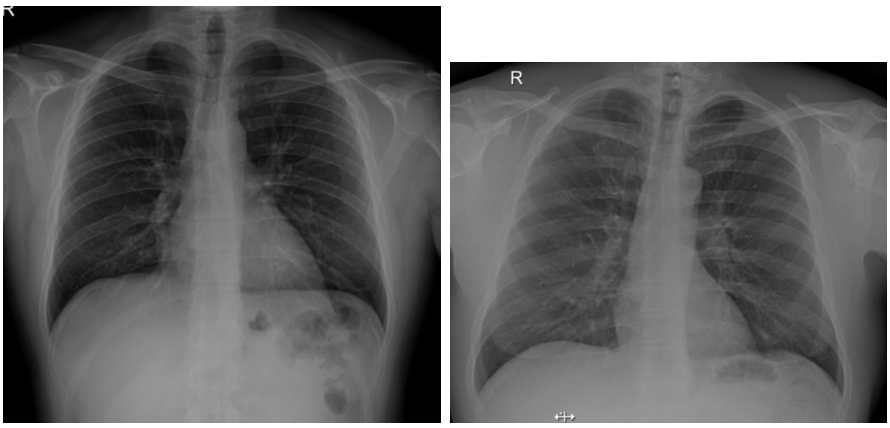


Figura 55: Detección de Opacidades Pulmonares:Verdaderos Negativos

En la Figs. 56, 57 y 58 , se presentan imágenes pertenecientes a la clase "opacidades pulmonares". En la Fig. 56, es posible observar casos que a simple vista se puede inferir que resultaron verdaderos positivos, ya que se presenta una detección de opacidad pulmonar con límites muy similares a su GT.

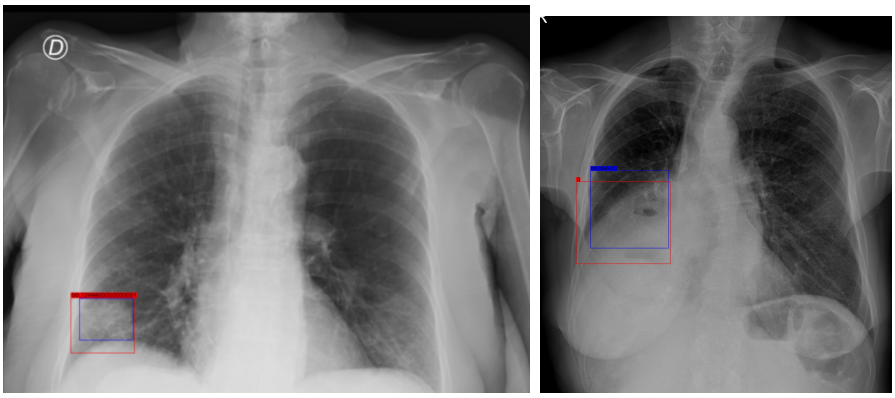

Figura 56: Detección de opacidades pulmonares: Verdaderos Positivos..Azul: detecciones del modelo. Rojo: GT

En la Fig. 57 en la imagen de la izquierda, el modelo realiza una detección de un derrame pleural, en lugar de detectar consolidaciones. Por lo tanto, se trata de un falso positivo. Podría considerarse un falso negativo porque la opacidad no se detecta, sin embargo detecta el hallazgo principal que es el derrame pleural. En la imagen de la derecha, se observa otro caso falso negativo, debido a que se observa una detección del tipo GT y ninguna propuesta por el modelo en dicha región.

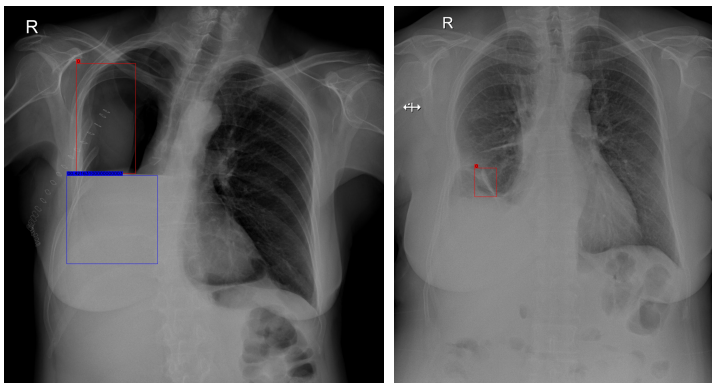

Figura 57: Detección de opacidades pulmonares: Falsos Negativos.Azul: detecciones del modelo. Rojo: GT

Otro fenómeno observado en las detecciones es la presencia de falsos positivos, tal como se presenta en la Fig. 58. En el caso de la imagen de la izquierda, es posible observar que el sistema detecta la presencia de una "opacidad pulmonar" en el tórax superior derecho, pero que se trata de un electrodo. Sin embargo, es importante mencionar, que en la misma imagen se observa la presencia de otros electrodos, que no son detectados por el sistema como opacidades pulmonares. Esto sugiere

que si bien el sistema puede presentar fallas de este tipo, a priori no parecería tan propenso a hacerlo. En la imagen de la derecha, se detectan una gran cantidad de falsos positivos. Lo que sucede en realidad es que las anotaciones por parte del HIBA marcan la presencia de masas en prácticamente toda la extensión de los pulmones y señalan de dónde se encuentran las principales. En cambio, el modelo resulta más preciso debido a que detecta las masas individualmente y de forma más exhaustiva.

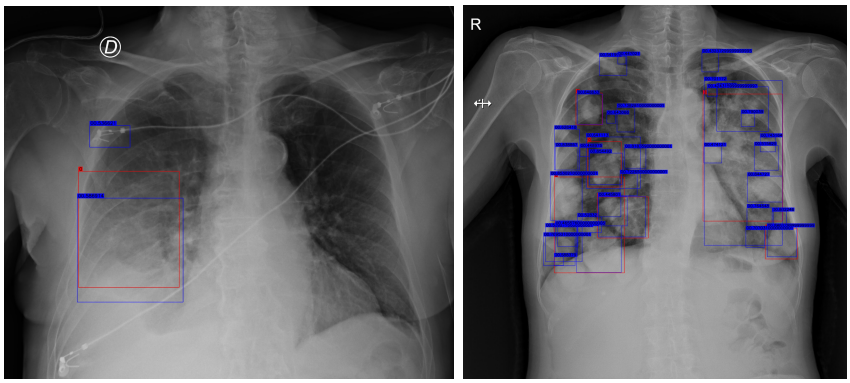


Figura 58: Detección de opacidades pulmonares: Falsos Positivos..Azul: detecciones del modelo. Rojo: GT

En conclusión, el sistema se mostró capaz de reproducir los resultados del *subset* de test para las imágenes propuestas por el HIBA, para las subclases con las que había sido entrenado: atelectasia, nódulo-masa, consolidaciones y opacidades pulmonares. Esto permite inferir que el cambio de dominio no afecta el desempeño del modelo y promete una posible incorporación del sistema al flujo de trabajo en HIBA. Sin embargo, debe tenerse en cuenta que el sistema no es capaz de generalizar resultados para aquellas clases de opacidades pulmonares con las que no fue entrenado: lesiones de la pared y patrón intersticial. A futuro, sería interesante contar con un DS con dichos tipos de hallazgos radiográficos y poder extender el modelo sumando esas subclases.

### 4.2.3. Experimento 6: Clasificación Binaria

Para que un modelo de IA implementado en el flujo clínico pueda asistir en la toma de decisiones es necesario que otorgue una respuesta binaria, que permita integrarlo a un sistema de *triage* o de alertas, y que sea interpretable en forma rápida y sencilla por los profesionales de salud. El objetivo de este experimento consiste en desarrollar y optimizar un modelo capaz de utilizar los resultados obtenidos por el algoritmo de detección de objetos YOLO V5m (es decir, cantidad de BBs y nivel de confianza para cada una de dichas detecciones) para determinar en forma binaria presencia o

ausencia de opacidades pulmonares.

En el contexto de este modelo, al mencionar al *train set* se hace referencia al *extended test set* ya procesado por el algoritmo de detección de objetos YOLO V5m. El *extended test set* está compuesto por el DS de testeo original más un agregado de imágenes pertenecientes a la clase "Sin hallazgos". La proporción final de esta última clase es del 80 % del total de imágenes. El *extended test set* fue confeccionado buscando simular la prevalencia real en la población. Por otra parte, al mencionar al *test set* en el contexto de este experimento, se hace referencia al conjunto de imágenes de validación del HIBA.

El enfoque definido para determinar la presencia o ausencia de opacidades pulmonares en imágenes a partir de los resultados del algoritmo de detección de objetos es el siguiente: se seleccionan como variables de entrada la cantidad de BB detectadas para cada una de las imágenes y un valor de confianza "resumen" obtenido a partir de los valores de los BB de cada imagen. Con dichas variables, se ajustan los coeficientes de la regresión logística (entrenamiento) para distintas configuraciones, y finalmente se comparan los diferentes modelos con el objetivo de escoger uno de ellos para realizar la prueba final en las imágenes del hospital.

El tipo de modelo utilizado es de regresión logística debido a que se desea obtener como salida una variable dicotómica, que es la presencia o ausencia de opacidades pulmonares en una determinada radiografía.

El primer modelo de regresión propuesto es el siguiente:

$$logit(P) = \ln(\frac{P}{1-P}) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmax} \tag{19}$$

donde, $logit(P)$ es el logaritmo natural del ODDS del evento (presencia de opacidades), $\alpha_i$ son los coeficientes que van a ser calculados mediante el método de máxima verosimilitud de manera de optimizar la predicción correcta, y $BB_{counts}$ y $BB_{cmax}$ representan las variables de entrada al modelo: la cantidad total de BB detectadas y el nivel de confianza máximo de entre el grupo de BB respectivamente. Esto busca representar que es suficiente poseer una BB con un valor de confianza elevado para clasificar la imagen como positiva.

$\alpha_0$ representa la ordenada al origen, $\alpha_1$ es el coeficiente que acompaña a la variable $BB_{counts}$ y por lo tanto, un aumento unitario en la cantidad de cajas que detecta el algoritmo implica un aumento equivalente al valor de este coeficiente en $logit(P)$. Dicho comportamiento es análogo para $\alpha_2$ con $BB_{cmax}$.

El segundo modelo planteado incluye como variables predictivas la cantidad de *bounding boxes* detectadas ($BB_{counts}$) y la media de la confianza de estas ($BB_{mean}$). Esta definición busca incorporar una variable que esté afectada por el nivel de confianza de todas las detecciones. La desventaja mayor de esta medida de tendencia central podría llegar a ser la presencia de *outliers* en la detección.

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmean} \tag{20}$$

El tercer modelo evaluado fue construido con el objetivo de seguir representando los valores de confianza de todas las BB, pero mitigando el posible efecto de los valores extremos o *outliers*. Por lo tanto, el valor "resumen" del nivel de confianza se obtuvo como la mediana de la confianza de los BB.

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmedian} \tag{21}$$

En el cuarto modelo se analiza la interacción de las variables del primer modelo. Es decir, si la combinación de ellas posee capacidad de predicción de la variable de salida.

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} + \alpha_2 * BB_{cmax} + \alpha_3 * BB_{cmax} * BB_{counts} \tag{22}$$

Debido a que modelos más simples pueden mejorar la capacidad de generalización, el quinto y sexto modelo consideran sólo una de las variables de salida de YOLO cada uno, con el fin de evaluar la contribución individual de cada una de ellas.

$$logit(P) = \alpha_0 + \alpha_1 * BB_{counts} \tag{23}$$

$$logit(P) = \alpha_0 + \alpha_1 * BB_{cmax} \tag{24}$$

Los modelos se entrenaron con el DS de *train* y los resultados de testeo se presentan en la tabla 10, expresados en términos del AIC y ROC-AUC.

Tabla 10: Comparación de Modelos para Binarización de Detección de Opacidades Pulmonares

| | **AUC** | **AIC** |
|---|---|---|
| **Modelo 1** | 0.9322 | 626.86 |
| **Modelo 2** | 0.9299 | 666.90 |
| **Modelo 3** | 0.9092 | 779.03 |
| **Modelo 4** | 0.9323 | 620.45 |
| **Modelo 5** | 0.9027 | 798.30 |
| **Modelo 6** | 0.9120 | 784.23 |

Recordando lo mencionado anteriormente acerca de estos parámetros y la ecuación 17, se puede concluir que un menor AIC se corresponde con un mejor modelo, y con una ROC-AUC mayor. Debido a los resultados resumidos en la tabla 10, como la diferencia entre los dos mejores modelos es ínfima, se escogió el Modelo 1 por su simplicidad.

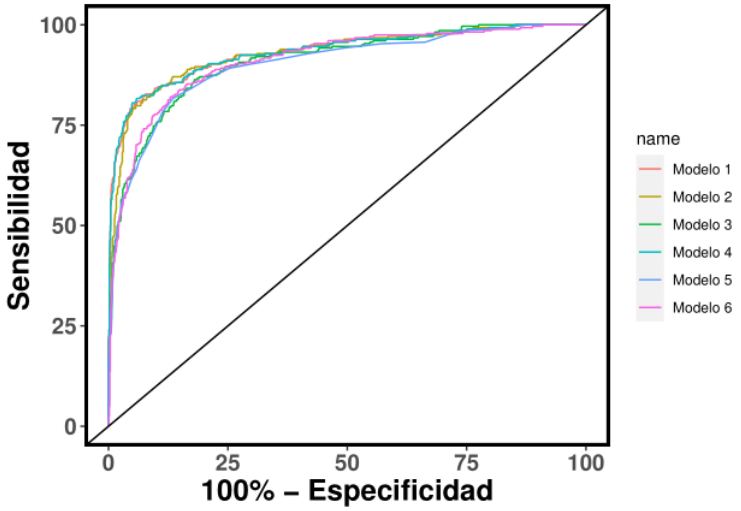


Figura 59: Curva ROC de Comparación Modelos de Clasificación Binaria

Tabla 11: Resultados Regresión logística Modelo 1

| | **Valor** | **P-valor** |
|---|---|---|
| $\alpha_0$ | -5.15 | <2e-16 |
| $\alpha_1$ | 0.13 | <2e-16 |
| $\alpha_2$ | 4.81 | <2e-16 |

En la tabla 11, se describen los resultados del Modelo 1. En primer lugar, se muestran los valores de $\alpha_i$ obtenidos en entrenamiento. Como los parámetros $alpha_i$ representan la variación en la logit(P) para una variación unitaria de la variable del modelo que acompañan, es posible concluir que la cantidad de BB posee menos influencia que el valor máximo de confianza de las cajas detectadas en una dada imagen. Más precisamente, por cada BB detectado el ODDS *ratio* del evento aumenta en un 0.1345, y por cada unidad de aumento del nivel de confianza el mismo aumenta 4.81. Esto implica que en ambos casos, tal como era de esperarse, las variables funcionan como factores de riesgo, es decir, un aumento en las mismas implica un aumento

en la probabilidad de que la imagen de entrada pertenezca a la clase opacidades pulmonares. En la última columna de la tabla se muestran los p-valores correspondientes a cada uno de dichos coeficientes. Dichos valores deben interpretarse como la probabilidad de que dada la muestra obtenida no exista relación entre la variable que representan y la variable de salida. Como esta probabilidad es sumamente baja, se debe concluir que existe una relación estadísticamente significativa entre cada una de las variables predictoras y la variable respuesta.

### 4.2.4. Experimento 7: Clasificación binaria en imágenes del HIBA

El último experimento realizado en el marco de este trabajo consiste en evaluar los resultados del clasificador binario escogido para las imágenes del *dataset* reducido del HIBA.

Se tomaron los resultados obtenidos en el experimento que se define en la sección 4.2.2 y tomando como entradas la cantidad de BB y el valor máximo de su nivel de confianza, se computó para cada imagen el resultado de la regresión logística entrenada en el apartado anterior (Modelo 1).

El primer interrogante que se buscó resolver, es si el DS del HIBA presentaba separatidad lineal, y de ser así, con respecto a cual de las variables escogidas. En la Fig. 60 si bien se observa una mayor densidad de imágenes con detecciones próximas a nulas, y las imágenes que poseen una gran cantidad de detecciones son clasificadas como pertenecientes a la clase “opacidades pulmonares”, existe una gran cantidad de imágenes que acorde a esta variable son clasificadas incorrectamente, independientemente del punto de corte. Por lo tanto, se debe concluir que el DS del HIBA no presenta separatidad lineal con respecto a la cantidad de BBs detectados, pero que resulta de utilidad para clasificar las imágenes en “opacidades pulmonares” y “sin hallazgos”.

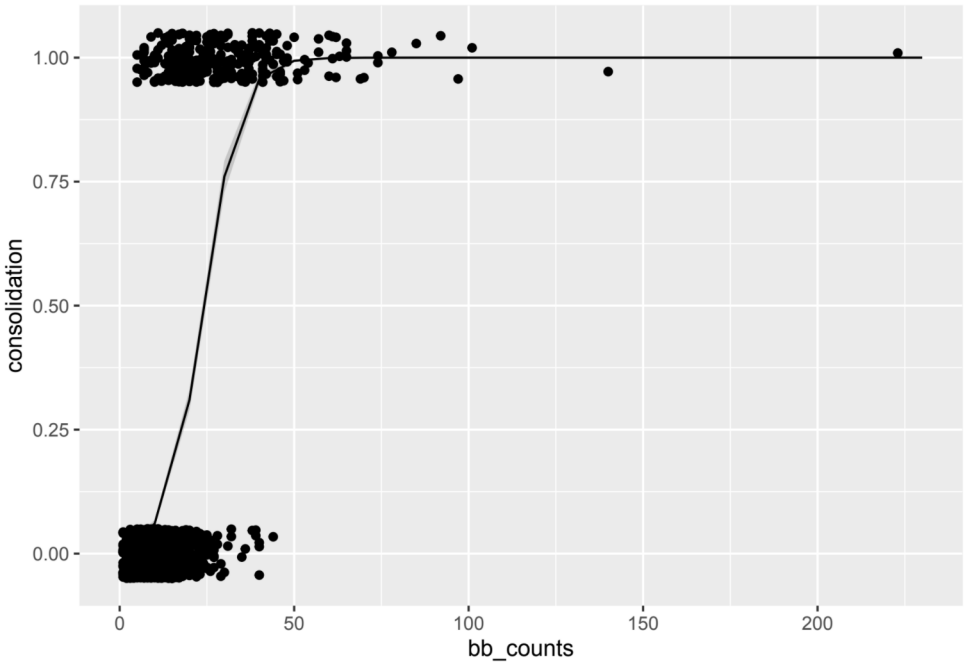


Figura 60: Curva sigmoidea que muestra la contribución individual de la variable $bb_{counts}$ a la clasificación en opacidades pulmonares o sin hallazgos

En la Fig. 61, se presenta el mismo gráfico que en la figura anterior, pero en este caso muestra la capacidad de separación en dos clases de la confianza máxima asignada a una caja en el proceso de detección de opacidades. Es posible observar una mayor densidad con nivel de confianza mínimo para las imágenes pertenecientes a la clase "sin hallazgos", y con nivel de confianza maximizado para las imágenes pertenecientes a la clase "opacidades pulmonares". Esta variable tampoco presenta separatidad lineal, pero evidentemente contribuye a la correcta binarización del modelo.

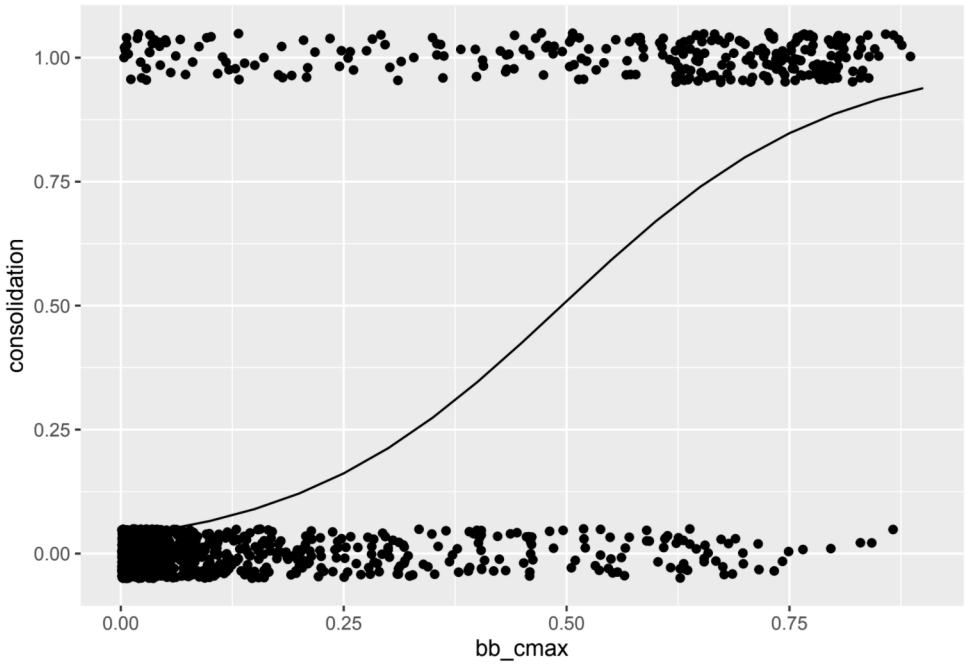


Figura 61: Curva sigmoidea que muestra la contribución individual de la variable $bb_cmax$ a la clasificación en opacidades pulmonares o sin hallazgos

Luego, se evaluó el modelo mediante el uso de VP, FP, VN y FN para todos los posibles puntos de operación del modelo, y a partir de dichos datos se construyó la correspondiente curva ROC. Los resultados obtenidos se resumen en la Fig. 62 que se presenta a continuación:

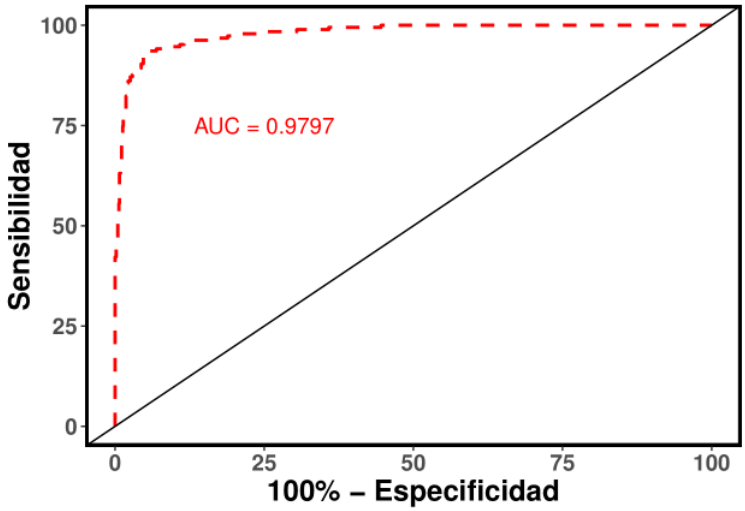


Figura 62: Curva ROC obtenida con Imágenes del HIBA

La curva se corresponde con un resultado por encima del azar (recta a 45°), incluso próximo a lo que se clasificaría como un excelente desempeño. Analíticamente, esto puede observarse en el AUC resultante que es de 0,9797. Esto implica un muy buen desempeño de clasificación, con puntos de operación de alta sensibilidad y especificidad.

Sin embargo, es importante tener en cuenta que las curvas ROC tienden a sobrestimar los resultados en los casos de clases no balanceadas, como en este *test subset*. El desbalance es una problemática muy frecuente en datos de origen biomédico, por lo que es importante complementar la evaluación de resultados con métricas que resalten problemas de clasificación en la clase minoritaria, como la curva de *precision-recall* (PR)[33]. En escenarios desbalanceados, una parte importante de la curva ROC podría corresponder a puntos de operación que presentan un gran número de falsos positivos (es decir, puntos con baja precisión). En consecuencia, el AUC-ROC está muy dominado por el rendimiento en puntos de operación que no serían aceptables en una aplicación real, convirtiéndola en una métrica inadecuada para tales escenarios. Por otro lado, en la curva de PR, aquellos puntos con un gran número de falsos positivos se corresponden con valores bajos de precisión y una menor altura de la curva, reduciendo el valor del AUC-PR. Por esta razón, se estudió también la curva PR, que se muestra en la Figura 63.

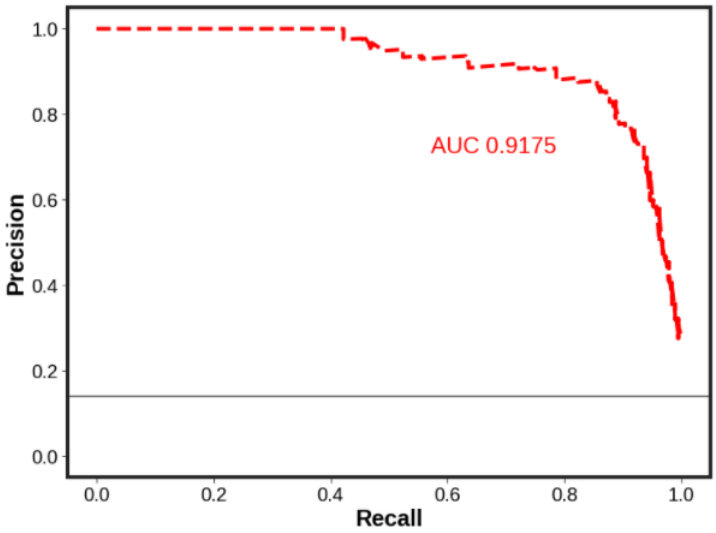


Figura 63: Curva Precision - Recall obtenida con Imágenes del HIBA

El área bajo la curva en este caso es de 0,9175. Este valor si bien es inferior al arrojado por la curva ROC, es ampliamente superior al azar, que en esta curva se representa como una recta con precisión igual a la proporción de muestras de la clase minoritaria (15 %).

En el caso de los clasificadores binarios, las curvas de *precision-recall* se pueden utilizar para determinar el punto de corte óptimo. Es una herramienta que permite escoger dónde ubicar el umbral definitivo cuando la salida del modelo es continua,y así poder brindar una respuesta binaria. En este caso, por tratarse de una aplicación médica, podría parecer prioritario disminuir los falsos negativos, ya que es más costoso no identificar una patología que realizar pruebas diagnósticas extras en un paciente que en realidad no tenía opacidades (falso positivo). Sin embargo, se debe tener en cuenta el rol del sistema en el camino diagnóstico [34].La opinión que el médico tiene de la herramienta es crucial para que sea utilizada en la práctica cotidiana, y esta opinión es afectada fuertemente por los falsos positivos. Por lo tanto, siguiendo la sugerencia de profesionales del HIBA, se escogió aquel umbral que garantizara un 80 % de *recall* con la mejor *precision* posible. El umbral que cumple dicha condición es 0,75 y el valor de *precision* asociado es de 88 %.

Con dicho valor de corte, se clasificaron las imágenes y se obtuvo la matriz de confusión que se presenta a continuación.

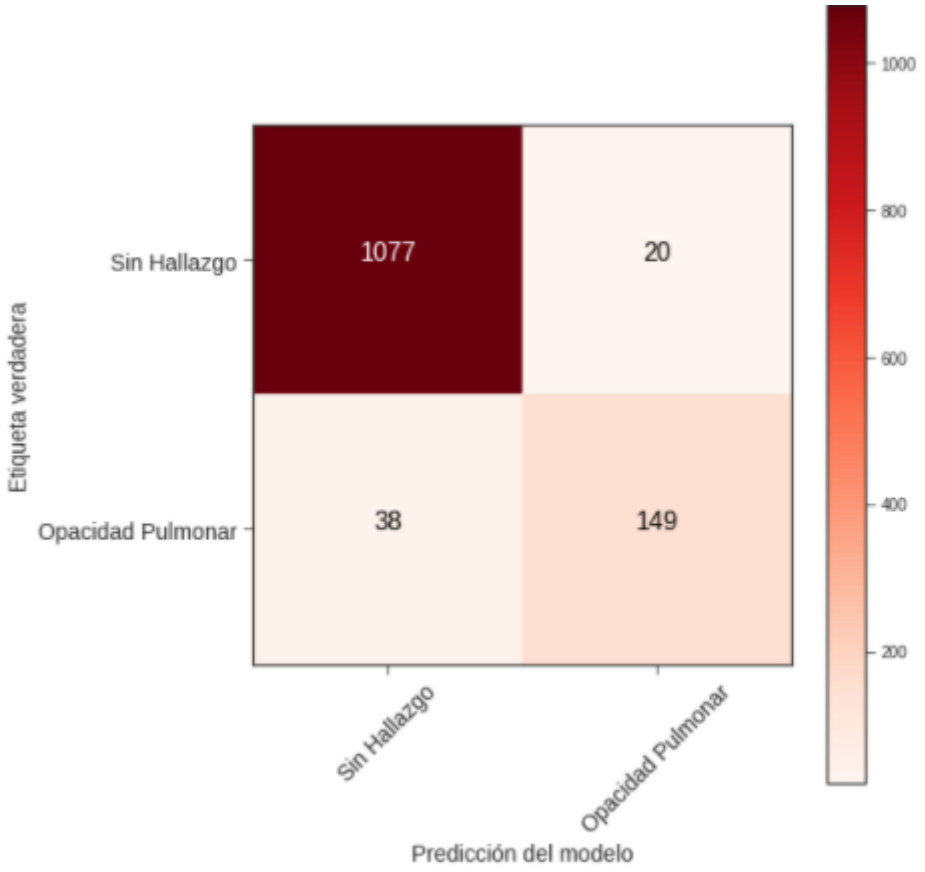


Figura 64: Matriz de confusión obtenida con Imágenes del HIBA y punto de corte en 0.75

Si bien en la mayor parte de los casos el modelo es capaz de clasificar correctamente la imagen, existen casos en los que falla en su predicción. Los errores con mayor prevalencia son los falsos negativos. A partir de estos valores de clasificación final, se calculó la métrica utilizada a lo largo de todo el trabajo para tareas de clasificación, el *F1- score*, que resultó de 0,84.

Por último se analizaron en forma cualitativa ejemplos de detecciones del modelo. En las Figs. 65 y 66 se presentan falsos positivos y falsos negativos, respectivamente. En color azul se observan las detecciones realizadas por el modelo. En color rojo sus GTs.

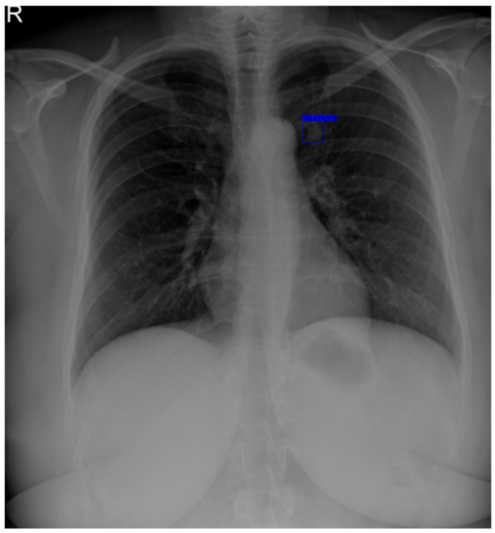


Figura 65: Falso positivo

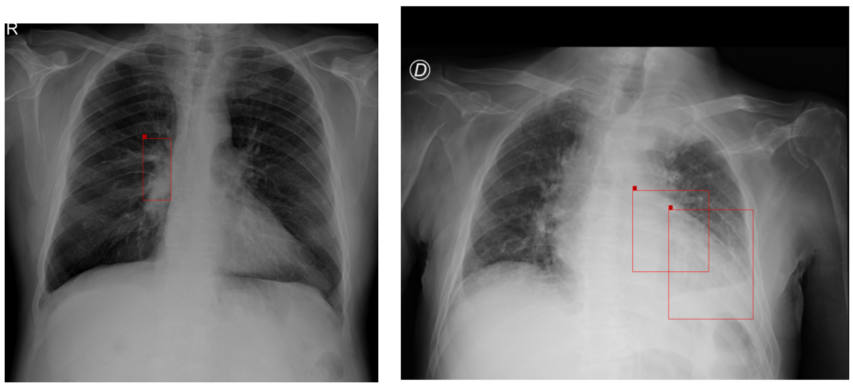


Figura 66: Falso negativo

En conclusión, el Modelo 1 de clasificación binaria a partir de detecciones de opacidades pulmonares demostró clasificar correctamente este tipo de hallazgos con un *F1-score* de 0.84. Para llegar a esta métrica, dado que se utilizó un modelo de regresión logística, fue posible escoger el punto de corte. Dicho valor se escogió tomando en cuenta referencias bibliográficas relativas a SSD, y la experiencia previa en el hospital.

# 5. Conclusiones

En este trabajo se presentó un sistema capaz de:

- Clasificar imágenes radiográficas según su región anatómica, las de tórax según su vista de adquisición y filtrar las radiografías de tórax posteroanteriores
- Detectar y localizar opacidades pulmonares en estas imágenes
- Funcionar como sistema de soporte a la toma de decisiones informando presencia o ausencia de opacidades pulmonares

El sistema fue evaluado tanto en imágenes de DS públicos que no se utilizaron para entrenar, como en imágenes del centro de salud en el que será implementado, el HIBA. Un resumen de los resultados se presenta en la tabla 12.

Tabla 12: Resultados Finales

| | **Métrica** | **DS Público** | **DS HIBA** |
|---|---|---|---|
| **Filtrado** | Macro F1 - Score | 0.99 | 0.77 |
| **Detección** | mAP | 0.34 | 0.21 |
| **Binarización** | AUC PR | - | 0.92 |

En primer lugar, se presenta el *macro F1-score* obtenido en el *subset* de testeo para el DS público armado a partir del de Padchest, MURA y las imágenes de cadera. El valor de la métrica evidencia que el sistema se comporta de manera prácticamente perfecta. Además, el análisis cualitativo de las imágenes mal clasificadas evidencia una deficiencia en la calidad de las imágenes. Esto resulta deseable, debido a que el modelo de opacidades pulmonares no funcionaría con imágenes poco penetradas. Sin embargo, al observar dicho resultado, se plantea el interrogante de si otras imágenes de mala calidad fueron clasificadas acorde a su etiqueta y por lo tanto no pudieron ser filtradas. Como conclusión de este experimento cabe resaltar la importancia de utilizar DS de calidad, debido a que la cota superior de funcionamiento de un sistema entrenado de forma supervisada se encuentra dada por la calidad de las imágenes y etiquetas que componen al *dataset*. Una segunda conclusión a la que se llega gracias a la experimentación con diferentes arquitecturas es que a pesar de su menor profundidad en comparación con ResNet-50, VGG-16 es suficientemente robusta para poder resolver la tarea de clasificación de vistas de radiografías.

Siguiendo con los resultados que se presentan en la Tabla 12, es posible observar una desmejora considerable en la métrica del modelo al utilizar como DS de testeo al propuesto por el hospital. La primera etapa del filtro bietápico arrojó un *F1-score* de 0.93 como consecuencia de una buena capacidad de generalización del modelo,

y una deficiente planificación de imágenes que componen la clase "otros". A futuro, sería interesante contar con un DS de entrenamiento capaz de distinguir otras vistas de las radiografías de tórax. Sin embargo, solo el 16 % de las imágenes mal clasificadas en primera etapa, fueron clasificadas por la segunda como pertenecientes a la clase PA. Lo ideal sería que al detector de opacidades pulmonares solo llegaran imágenes pertenecientes a la clase PA y que el sistema de clasificación tuviera una etiqueta para asignar de manera correcta estas imágenes. De esta forma, el sistema de clasificación resultaría más robusto y podría ser implementado previo a diferentes sistemas que toman como entradas proyecciones particulares.

Siguiendo con los resultados de detección de opacidades pulmonares, la principal conclusión que se puede obtener es que si bien las vistas de entrada para el sistema son todas PA, la técnica de aumentación de datos mosaiquismo, mostró su utilidad frente a la falta de instancias de objetos necesarias para llevar al sistema a una detección óptima.

Al momento de trasladar el sistema entrenado y buscar validar los resultados con el DS disponibilizado por el hospital, el mAP se encontró por debajo de lo esperado, sugiriendo posibles sesgos producto del *dataset shift*. Sin embargo, poniendo atención a las subclases que componen a la clase "opacidades pulmonares" y corriendo nuevamente el modelo sobre aquellas subclases para las que el mismo fue entrenado, los resultados mejoraron a un mAP 0.5 de 0.31, un valor mucho más cercano al obtenido con el DS público, lo que sugiere que el modelo es capaz de generalizar en datos de un dominio distinto al de entrenamiento, pero no de detectar clases que no estaban presentes en este dominio. A futuro, resultaría interesante entrenar el modelo con todas las clases para las que se pretende detectar en la práctica. Para esto, sería necesario contar con un DS con la suficiente cantidad de imágenes, de instancias y de calidad en las etiquetas.

En cuanto a los resultados obtenidos en la etapa de binarización de los resultados de las detecciones a partir de la cantidad de opacidades detectada por imagen y del máximo nivel de confianza para alguna de ellas, se obtuvo un modelo de regresión logística con el objetivo de maximizar la capacidad de separación entre clases. El clasificador obtuvo una AUC PR de 0.92. Se escogió una recall de 0.8, coincidente con una precision de 0.88 y un *F1-Score* de 0.84. Esta etapa permitió convertir al sistema en una herramienta de soporte a la toma de decisiones.

Finalmente, se puso todo en conjunto formando una API que brinda acceso al sistema de soporte a la toma de decisiones para la detección de opacidades pulmonares en radiografías mediante el uso de redes neuronales convolucionales.

Una de las ventajas más importantes que posee el sistema es su modularidad. Es decir, que cada uno de los bloques de clasificación y de detección pueden ser implementados por separado. Esto otorga versatilidad para poder ser aplicados no sólo en el centro de salud que aquí se presenta, sino que en otros con diferentes necesidades sin obligar a reemplazar sistemas existentes. Además, tomando como base el SSD y realizando algunos aditamentos al modelo, se podría utilizar la base de este sistema como herramienta de *triage*.

De aquí en adelante resta sumergirse en el proceso de implementación del algoritmo, la incorporación a los flujos clínicos actuales, y evaluar la utilidad del mismo en términos de los propios usuarios.

# 6. Referencias bibliográficas


[1] L. R. Folio, *Chest imaging: an algorithmic approach to learning*. Springer Science & Business Media, 2012.

[2] H. Cliffe, D. Liu, V. Wykes, E. Denton y G. Maskell, "Summary of The Royal College of Radiologists'(RCR) reporting backlog surveys and assessment of potential causes and solutions," *Clinical Radiology*, vol. 71, S10, 2016.

[3] "Final report: recommendations of the clinical advisory committee—plain x-ray image reporting backlog," Clinical Excellence Commission, Sydney Australia, Technical report, 2014.

[4] C. P. Friedman, "A "fundamental theorem" of biomedical informatics," *Journal of the American Medical Informatics Association*, vol. 16, n.º 2, págs. 169-170, 2009.

[5] X. Pardell. dirección: `https://www.pardell.es/curso-rayos-x.html`.

[6] R. S. of North America (RSNA) y A. C. of Radiology (ACR), *radiological society of north america (rsna) and american college of radiology (acr)*. dirección: `https://www.radiologyinfo.org/en/info.cfm?pg=chestrad#:~:text=The\%20chest\%20x\%2Dray\%20is,of\%20the\%20spine\%20and\%20chest.`.

[7] E. Çallı, E. Sogancioglu, B. van Ginneken, K. G. van Leeuwen y K. Murphy, "Deep Learning for Chest X-ray Analysis: A Survey," *Medical Image Analysis*, pág. 102 125, 2021.

[8] A. Bustos, A. Pertusa, J.-M. Salinas y M. de la Iglesia-Vayá, "Padchest: A large chest x-ray image dataset with multi-label annotated reports," *Medical Image Analysis*, pág. 101 797, 2020.

[9] B. Sahu y R. Verma, "DICOM search in medical image archive solution e-Sushrut Chhavi," en *2011 3rd International Conference on Electronics Computer Technology*, IEEE, vol. 6, 2011, págs. 256-260.

[10] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. Tong, D. H. Dinh y col., "VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations," *arXiv preprint arXiv:2012.15029*, 2020.

[11] F. Chollet, *Deep Learning with Python*. New York, USA: Manning Publications, 2018.

[12] N. Ketkar y E. Santana, *Deep Learning with Python*. Springer, 2017, vol. 1.

[13] R. L. M. A. Mosquera Candelaria Diaz Facundo Nahuel, *Inteligencia Artificial en Imágenes Médicas de la Teoría a la Aplicación*. Delhospital Ediciones, 2021, https://delhospitalediciones.hospitalitaliano.edu.ar/visor/184/0011769C41(visited 2021-30-05).

[14] E. R. Ranschaert, S. Morozov y P. R. Algra, *Artificial Intelligence in Medical Imaging: Opportunities, Applications and Risks*. Springer, 2019.

[15] C. Baskin, N. Liss, E. Zheltonozhskii, A. M. Bronstein y A. Mendelson, "Streaming architecture for large-scale quantized neural networks on an FPGA-based dataflow platform," en *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2018, págs. 162-169.

[16] P. Mishra, *Convolutional Neural Networks (CNN)*, 2019. dirección: https://iq.opengenus.org/convolutional-neural-networks/.

[17] K. Simonyan y A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] C. Baskin, N. Liss, A. Mendelson y E. Zheltonozhskii, "Streaming Architecture for Large-Scale Quantized Neural Networks on an FPGA-Based Dataflow Platform," jul. de 2017.

[19] Y. Zheng, C. Yang y A. Merkulov, "Breast cancer screening using convolutional neural network and follow-up digital mammography," mayo de 2018, pág. 4. DOI: 10.1117/12.2304564.

[20] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, "You only look once: Unified, real-time object detection," en *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, págs. 779-788.

[21] T. A. Team, *Yolo v5-explained and demystified*, 2020. dirección: https://towardsai.net/p/computer-vision/yolo-v5\%E2\%80\%8A-\%E2\%80\%8Aexplained-and-demystified.

[22] L. Velasco, *Optimizadores en redes neuronales profundas: UN Enfoque Práctico*, 2020. dirección: https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr\%C3\%A1ctico-819b39a3eb5.

[23] V. Bushaev, *Understanding rmsprop - faster neural network learning*, 2018. dirección: https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a.

[24] J. P. Cohen, M. Hashir, R. Brooks y H. Bertrand, "On the limits of cross-domain generalization in automated X-ray prediction," en *Medical Imaging with Deep Learning*, PMLR, 2020, págs. 136-155.

[25] M. A. Musen, B. Middleton y R. A. Greenes, "Clinical decision-support systems," en *Biomedical informatics*, Springer, 2021, págs. 795-840.

[26] C. Mosquera, F. N. Diaz, F. Binder, J. M. Rabellino, S. E. Benitez, A. D. Beresñak, A. Seehaus, G. Ducrey, J. A. Ocantos y D. R. Luna, "Chest x-ray automated triage: A semiologic approach designed for clinical implementation, exploiting different types of labels through a combination of four Deep Learning architectures," *Computer Methods and Programs in Biomedicine*, vol. 206, pág. 106 130, 2021.

[27] C. Mosquera, F. Binder, F. N. Diaz, A. Seehaus, G. Ducrey, J. A. Ocantos, M. Aineseder, L. Rubin, D. A. Rabinovich, A. E. Quiroga y col., "Integration of a deep learning system for automated chest x-ray interpretation in the emergency department: A proof-of-concept," *Intelligence-Based Medicine*, vol. 5, pág. 100 039, 2021.

[28] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball y col., "Mura: Large dataset for abnormality detection in musculoskeletal radiographs," *arXiv preprint arXiv:1712.06957*, 2017.

[29] G. Awasthi, *Part1Xray*, Imágenes Obtenidas de Kaggle, `https://www.kaggle.com/gouravawasthi/part1xray/version/1`.

[30] B. Kompa, J. Snoek y A. L. Beam, "Second opinion needed: communicating uncertainty in medical machine learning," *NPJ Digital Medicine*, vol. 4, n.º 1, págs. 1-6, 2021.

[31] Ultralytics, *YOLO V5*, `https://github.com/ultralytics/yolov5`, 2021.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár y C. L. Zitnick, "Microsoft coco: Common objects in context," en *European conference on computer vision*, Springer, 2014, págs. 740-755.

[33] T. Saito y M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, n.º 3, e0118432, 2015.

[34] J. M. Fardy y B. J. Barrett, "Evaluation of diagnostic tests," en *Clinical Epidemiology*, Springer, 2015, págs. 289-300.