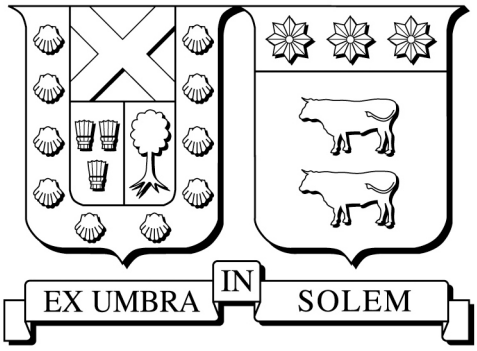
2021-06

Sistema de recuperación de imágenes médicas basado en contenido para el sistema de salud chileno

Molina Barra, Gabriel Andrés

Universidad Técnica Federico Santa María
Departamento de Informática



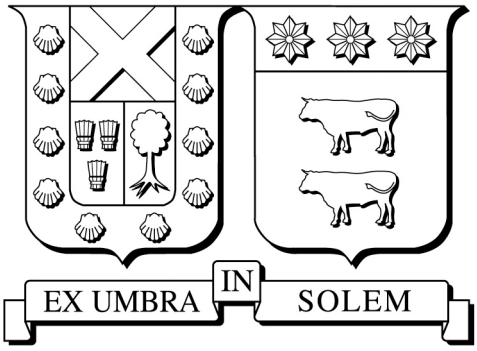
Herramienta de recuperación de imágenes médicas basado en contenido para el sistema de salud chileno.

Gabriel Molina Barra

Magíster en Ciencias de la Ingeniería Informática

Junio 2021.

Universidad Técnica Federico Santa María
Departamento de Informática



Sistema de recuperación de imágenes médicas basado en contenido para el sistema de salud chileno.

DISERTACIÓN

Como requisito parcial para optar al grado de

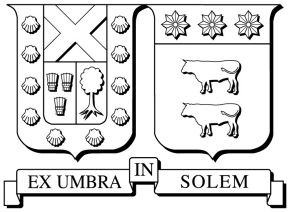
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

presentado por

Gabriel Molina Barra

Profesor Guía
Marcelo Mendoza

Santiago, Chile
Junio 2021.



TÍTULO DE LA TESIS: **SISTEMA DE RECUPERACIÓN DE IMÁGENES MÉDICAS BASADO EN CONTENIDO PARA EL SISTEMA DE SALUD CHILENO.**

AUTOR: **GABRIEL MOLINA BARRA**

Trabajo de Tesis, presentado en cumplimiento parcial de los requisitos para el Grado de **Magíster en Ciencias de la Ingeniería Informática** de la Universidad Técnica Federico Santa María.

COMISIÓN EXAMINADORA:

Marcelo Mendoza
Universidad Técnica Federico Santa María
Chile

Profesor Guía

Mauricio Araya
Universidad Técnica Federico Santa María
Chile

Profesor Co-Guía

Mauricio Solar
Universidad Técnica Federico Santa María
Chile

Profesor Correferente Interno

Victor Castañeda
Universidad de Chile
Chile

Profesor Correferente Externo

Claudio Torres
Universidad Técnica Federico Santa María
Chile

Presidente de la Comisión

A mi familia, compañeros y profesores por apoyarme en todo.

Resumen

En el área de la salud, las imágenes médicas han sido un recurso fundamental para el diagnostico oportuno de patologías, la investigación de enfermedades y herramienta de estudio para los futuros profesionales de la salud. Por esta misma causa, sistemas de búsqueda de imágenes se han desarrollado a lo largo de los años para ayudar a la recuperación de dicha información. Estos sistemas funcionan transformando una imagen médica en un vector multidimensional para así construir índices en bases de datos, generándose de esta forma buscadores rápidos y automáticos. Los sistemas mas tradicionales utilizan descriptores de una imagen para transformarla en un vector multidimensional, pero en la actualidad, esto es poco factible ya que la gran mayoría de las imágenes médicas no están correctamente etiquetadas y/o estudiadas. Esto ocurre debido a que el volumen de datos generados es tan grande que no existen suficientes expertos para estudiar todas las imágenes, resultando en que los sistemas tradicionales de búsquedas sean abrumados por el rápido crecimiento de datos y la falta de evaluación de expertos.

Se han propuesto diferentes métodos para resolver el problema denominado Content Base Image Retrieval System (CBIRS), en el cual, se agrupan todos los sistema de búsqueda de imágenes basado en contenido, incluyendo búsqueda de reconocimiento facial, imágenes naturales y el tema central de esta tesis, las imágenes médicas; siendo estas: radiografías, tomografías computacionales, examen histológico, entre otros. Existen autores que han intentado mejorar estos sistemas por el área de desarrollo de software, incrementando la infraestructura, perfeccionando los métodos de indexación y actualizando sistemas basados en la tecnología más moderna. A pesar de que este esfuerzo es suficiente para hacer los sistemas de búsqueda más robustos al crecimiento actual de datos en la gran mayoría de problemas, no son lo suficientemente eficaces para transformar en sistemas suficientemente escalables para problemas más complejos como son las imágenes médicas. Por estas razón, otros autores han preferido el área del deep learning para mejorar los vectores asociados al sistema, obteniendo mejores resultados

en recuperación de imágenes, obteniendo el estado del arte en muchos tipos de búsquedas de enfermedades, pero a pesar de que estos sistemas basados en redes neuronales son más modernos, sufren de la misma falencia de su contraparte tradicional, lo cual es la falta de datos etiquetados, llevando a estos sistemas a tener una gran dificultad a ser implementados en un ambiente de trabajo médico real.

En este documento se propone un CBMIRS basado en deep learning, el cual estará compuesto por dos secciones de redes neuronales que en conjunto construirán un vector más robusto para el sistema y además, este nuevo CBMIRS podrá ser ocupado a pesar de que las imágenes no lleven etiqueta, permitiendo una mejor escalabilidad en un ambiente clínico real. Esta propuesta se basa en el uso de combinar lo aprendido por un modelo de segmentación y un modelo de clasificación, para transformar una imagen en un vector con mayor información que utilizando una única red neuronal, permitiendo que un sistema de búsqueda de imágenes médica tenga mejores resultados y escalabilidad.

La evidencia experimental, muestra que comparado con otros métodos del estado del arte, en recuperación de imágenes médicas, la propuesta muestra una gran ventaja al buscar imágenes que contengan una cierta enfermedad en escenarios clínicos reales. Además, se demuestra la capacidad efectiva de transferencia de información mediante la utilización de los embeddings resultantes del segmentador en el reentrenamiento del clasificador o mediante la concatenación de embeddings resultantes. Finalmente, se demostró que la utilización de un algoritmo de eliminación de near-duplicate tiene un efecto regularizante en la de búsqueda de imágenes, permitiendo que sistemas que utilizan modelos más simples sean igual de competitivos que sistemas que utilizan modelos más complejos.

Palabras Clave: *Sistema de recuperación médica basado en contenido, aprendizaje profundo, búsqueda de imágenes médicas, sistema asistido por computadora.*

Abstract

In the health area, medical images have been a fundamental resource for the timely diagnosis of pathologies, the investigation of diseases and a study tool for future health professionals. For this same reason, image search systems have been developed over the years to aid in the retrieval of such information. These systems work by transforming a medical image into a multidimensional vector in order to build indexes in databases, thus generating fast and automatic search engines. The most traditional systems use descriptors of an image to transform it into a multidimensional vector, but at present, this is not very feasible since the vast majority of medical images are not correctly labeled and/or studied. This, because the volume of data generated is so great that there are not enough experts to study all the images, resulting in traditional search systems being overwhelmed by the rapid growth of data and the lack of expert evaluation.

Different methods have been proposed to solve the problem called Content Base Image Retrieval System (CBIRS), in which all content-based image search systems are grouped, including facial recognition search, natural images and the central theme of this thesis, medical images; These being: X-rays, computational tomography, histological examination, among others. There are authors who have tried to improve these systems in the area of software development, increasing the infrastructure, perfecting indexing methods and updating systems based on the most modern technology. Although this effort is enough to make search systems more robust to the current growth of data in the vast majority of problems, these efforts are not effective enough to transform into systems scalable enough for more complex problems such as medical images. For these reasons, other authors have preferred the area of deep learning to improve the vectors associated with the system, obtaining better results in image retrieval, obtaining the state of the art in many types of disease searches, but despite the fact that these systems based In neural networks they are more modern, they suffer from the same shortcoming of their traditional counterpart, which is the lack of labeled data, leading these systems to have great difficulty in being implemented in a real medical work environment.

In this document, a CBMIRS based on deep learning is proposed, which will be composed of two sections of neural networks that together will build a more robust vector for the system and also, this new CBMIRS can be occupied even though the images do not have label, allowing better scalability in a real clinical environment. This proposal is based on the use of combining what has been learned by a segmentation model and a classification model, to transform an image into a vector with more information than using a single neural network, allowing a medical image search system to have better results and scalability.

The experimental evidence shows that compared to other state-of-the-art methods in medical image retrieval, the proposal shows a great advantage when looking for images that contain a certain disease in real clinical settings. Furthermore, the effective information transfer capacity is demonstrated by using the resulting embeddings of the segmenter in the retraining of the classifier or by concatenating the resulting embeddings. Finally, it was shown that the use of a near-duplicate elimination algorithm has a regularizing effect on image search, allowing systems that use simpler models to be just as competitive as systems that use more complex models.

Keywords: *Content based medical retrieval system, Deep Learning, Medical image search, Computer Aided system*

Glosario

ANN: Sigla para *Artificial Neural Networks*. Modelo de aprendizaje denominado redes neuronales artificiales.

CNN: Sigla para *Convolutional Neural Network*. Un tipo de red neuronal que realiza operaciones de convolución en sus capas.

CeNet: Acrónimo para *Context Encoder Network*. Un modelo de red neuronal para segmentación de imagenes medicas 2D.

CBMIRS: Siglas para *Content-based medical image retrieval system*. Un sistema de recuperación de información a través de imágenes médicas.

DL: Sigla para *Deep Learning*. Un tipo de modelos de red neuronal que procesa sobre datos brutos.

ELU: Siga para *Exponential Linear Unit*. Función de perdida que a diferencia de ReLU tienen valores negativos que les permiten acercar las activaciones de unidades medias a cero como *batch normalization*, pero con menor complejidad computacional.

FC: Sigla para *Fully Connected*. Una capa de una red neuronal con todas las neuronas o unidades conectadas entre sí, indicando que interactúan entre sí.

FF: Sigla para *Feed Forward*. Una red neuronal clásica, igual que una MLP.

Ground Truth: Etiqueta real para un cierto dato, definida bajo la supervisión humana en cada problema.

Modelo Predictor: modelo de aprendizaje, usualmente una función, que predice un cierto valor, usualmente el Ground Truth.

Near-Duplicate: Terminología utilizada en para describir una imagen o vector que tiene una alta similitud a una imagen original.

PACS: Sigla para *Picture archiving and communication system*. un sistema de almacenamiento y distribución de imagen médicas.

Índice general

Índice de tablas

Índice de figuras

Publicaciones Asociadas

- Camilo Sotomayor, Marcelo Mendoza, Victor Casteñeda, Humberto Farías, Gabriel Molina, Gonzalo Pereira, Steffen Hartel, Mauricio Solar, Mauricio Araya. Content-based medical image retrieval and intelligent interactive visual browser for medical education, research and care. Diagnostics, in press, 2021 [WoS]

- Gabriel Molina, Marcelo Mendoza, Ignacio Loayza, Camilo Nunez, Mauricio Araya, Victor Castaneda, Mauricio Solar. A new content-based image retrieval system for SARS-CoV-2 computer-aided diagnosis. In: Proceedings of the International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2021), 25-26 March, LNEE 784, Springer, 2021 [Scopus].

Introducción

En esta introducción se presenta el alcance de la investigación, el dominio del problema, los objetivos, hipótesis y la organización del documento.

. Alcance de la Investigación

En los últimos años, la medicina y la tecnología se han fusionado de tal forma que ya no es posible pensar en una sin la otra.

La medicina junto a la ingeniería han empujado la solución de problemas complejos dando lugar a herramientas que otorgan una nueva perspectiva del cuerpo humano. El uso de estas nuevas herramientas, genera información (*Data*) la cual es almacenada en PACS. Lamentablemente, el volumen de datos es tan grande que la recuperación de dicha información se vuelve compleja, perdiendo instancias valiosas para investigar o curar una patología anteriormente vista. Por ejemplo, si un médico tuviese acceso a comparar los resultados de una tomografía con casos anteriores, podría realizar un diagnóstico más rápido, permitiendo al paciente que inicie el tratamiento lo antes posible. La cantidad de información almacenada es demasiado vasta para que un médico la pueda analizar en su completitud. Para solucionar este problema en específico se requiere la implementación de un **sistema de recuperación de imágenes médicas basado en contenido (CBMIRS)**, con el cual los doctores o tecnólogos médicos podrán buscar casos similares ya tratados con la finalidad de aplicar ese conocimiento a casos actuales. Desafortunadamente, con el incremento de datos producidos diariamente y la complejidad innata de las imágenes médicas, la comunidad científica ha decidido buscar opciones más modernas, siendo éstas, el área del *Deep Learning (DL)* y en específico, el área de las *Convolutional Neural Network (CNN)*.

Según Litjens et al.[**LKB+17**] el DL ha tenido un fuerte impacto en como se desarrollan

las aplicaciones en el área de la salud, en especial, en el área de la imagenología médica, tomando impulso a partir del año 2015 y concentrándose en tareas de segmentación de órganos y subestructuras, detección de objetos, clasificación de exámenes médicos, recuperación de información de imágenes entre otros.
Uno de los más tempranos CBMIRS básados en DL fue propuesto por Anavi et al. [**AKG$^+$15**] donde se demostró que un sistema de recuperación de información basado en una CNN básica de 5 capas tiene un mejor desempeño que técnicas más tradicionales al recuperar radiografías de tórax.
Posterior a esto, los esfuerzos en esta área se centraron en mejorar el resultado de la búsqueda de contenido del sistema y aumentar el número de clases que el sistema puede analizar, ya sea imágenes del mismo órgano pero con una enfermedad distinta, imágenes de distintos órganos o imágenes en distinto formato (Rayos-X, CT, MRI, Ultrasonido, Etc).
Azevedo-Marques et al.[**DAMMSR17**] introdujo los conceptos básicos para implementar un CBMIRS exitoso en un ambiente clínico real y presenta el sistema *Higiia*: un CBMIRS para mamografías. Este sistema tiene la particularidad de poner en práctica una búsqueda mejorada con eliminación de elementos *near-duplicate* y feedback de usuario, permitiendo generar una búsqueda más rica en contenido que un modelo de indexación clásico.
Owais et al. [**OACP19**] creó un CBMIRS capaz de clasificar y recuperar con una métrica F1 entre 81 % y 82 % 50 distintos tipos de clases, incluyendo imágenes de distintos órganos, enfermedades, formato (Rayos-X, CT, MRI, Ultrasonido, Etc) y tamaño. Esto prueba que los CBMIRS pueden soportar una gran carga de clases, demostrando en su investigación que las redes pre-entrenadas son útiles para problemas en la medicina.
Cai et al. [**CLQ$^+$19**] destacó el problema que en los CBMIRS la sección de codificación vectorial de la imagen y la indexación de dichos vectores se presentan como tareas separadas. Cai introdujo una red siamesa y una nueva función de perdida. Esta red es capaz de comparar dos imágenes y aproximar o alejar los vectores resultantes, dependiendo si son imágenes parecidas o no. Esto permitió generar búsquedas mas exactas y tanto el tiempo de respuesta como el re-entrenamiento del sistema.
Pinho et al. [**PSC19**] estudió la factibilidad de un entrenamiento no supervisado a través de un modelo Encoder-Decoder y un discriminador, para generar un sistema de búsqueda que no utilice etiquetas previas obteniendo resultados poco concluyentes en su investigación. El autor propone distintas ideas de como proseguir en el caso de implementar sistemas no supervisados o semi-supervisados.

La gran limitación de los CBMIRS actuales, es que la gran mayoría necesita un *dataset* etiquetado para poder ser entrenado. Esto, y otros factores humanos, limitan la capacidad de generar CBMIRS útiles en hospitales o clínicas a largo plazo, ya que con la llegada de nuevos datos y su lenta etiquetación es difícil mantener el sistema actualizado. Por otro lado, los CBMIRS basados en técnicas de entrenamiento de redes no supervisadas, no son muy exitosas, ya sea por que fallan fuera del estudio con casos de

borde o por que el problema que resuelven es muy simple y aporta poco a la comunidad médica.

. Hipótesis y Objetivos

Se presenta la idea de implementar un CBMIRS basado en la composición de información de dos redes supervisadas, una de ellas dedicada a la segmentación de órganos o enfermedades y la segunda dedicada a la clasificación de enfermedades. Este estudio se focaliza en dos enfermedades respiratorias para su estudio, las cuales son el actual Covid-19 y nódulos pulmonares. A diferencia de investigaciones previas en el estado del arte, el presente trabajo utilizará los *embeddings* resultantes de la red de segmentación como entrenamiento para la red de clasificación, generando un proceso de vectorización de la imagen que también toma en cuenta la información generada al segmentar una cierta región de interés (ROI), diferenciándose de trabajos previos que solamente ocupan la imagen. La hipótesis planteada es lo siguiente:
La generación de *image embeddings* que contengan la información de un segmentador y un clasificador de imágenes médicas permite producir un CBMIRS más robusto, permitiendo así la implementación en un ambiente clínico real. Las dos formas de generar este *embeddings* son:

- Transferir la información del segmentador al clasificador mediante el entrenamiento del modelo, donde se utilizaran los *embeddings* resultantes del segmentador como datos de entrada al clasificador para finalmente usar los *embeddings* del clasificador en el CBMIRS.
- Transferir la información del segmentador al clasificador mediante el entrenamiento del modelo, donde se utilizará los *embeddings* resultantes del segmentador como datos de entrenamiento. Posterior a esto, se usará en el CBMIRS los *embeddings* que son la concatenación del vector resultante del segmentador y del clasificador.

Esta hipótesis es falsable manteniendo la restricción de que el sistema creado bajo esta propuesta mantenga una eficiencia mínima de *precision* y *recall* con respecto a otros trabajos del estado del arte.

El objetivo principal de esta investigación es el diseño e implementación de un CBMIRS en hospitales Chilenos. Este nuevo sistema, utilizará técnicas de vectorización inspiradas en áreas de procesamiento de lenguaje natural para permitir crear un sistema más robusto, escalable y que busca ser competitivo en resultados de recuperación de información con respecto a modelos del estado del arte en un ambiente clínico real. Los objetivos específicos son:

1. Implementar un modelo de DL de vectorización de imagen combinando información de segmentación y clasificación.
2. Establecer un sistema de indexación adecuado para el modelo propuesto.

3. Comparar resultados sobre *dataset* públicos para validar propuesta.
4. Probar el sistema completo con datos generados en clínicas y hospitales Chilenos.
5. Validar la propuesta con un equipo de expertos en medicina.
6. Presentar la investigación en una conferencia nacional o internacional donde se comuniquen los resultados obtenidos.

. Metodología del Trabajo

La metodología utilizada para el desarrollo de esta investigación consiste en lo siguiente: En primer lugar, una revisión extensa del estado del arte en CBMIRS. Con un conocimiento más claro en estos sistemas y sus distintos enfoques, se analizarán las principales debilidades. Posterior a esto, analizar qué estudios no se han realizado, para así producir una nueva solución que no se haya explorado aún. Se necesita de un ambiente controlado para la experimentación inicial, para esto se trabajará con *dataset* públicos de distintas patologías antes de trabajar con datos Chilenos. Utilizando iteración y una constante revisión del estado del arte es un factor clave para el desarrollo del proyecto, como también conocer las debilidades prácticas y empíricas.

. Organización del Documento

El presente trabajo se divide en secciones. A continuación, se presenta todo el *background* teórico con los antecedentes necesarios para entender el trabajo propuesto, en el Capítulo 1. Luego, en el Capítulo 2 se formaliza el problema y se presenta el estado del arte, además de una breve discusión acerca de los métodos, sus diferencias y similitudes. Con esto, la propuesta es introducida en el Capítulo 3, detallando cuatro métodos distintos, además de una discusión comparativa de éstos en conjunto a la propuesta. Finalmente, se presentan los resultados bajo diferentes configuraciones experimentales y distintos dataset detallados en el Capítulo 4, para dar lugar a la conclusión en el último capítulo. Además de todo este material, se adjuntan ciertos detalles adicionales en el Apéndice A y B al final del documento.

Capítulo 1

Antecedentes - Background

A continuación, se entregarán los antecedentes necesarios para comprender las temáticas usadas en el presente trabajo sobre el contexto en el cual se sitúa. Los antecedentes presentados entregan una breve introducción a los temas tratados tales como aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi-supervisado, redes neuronales artificiales, clasificación y segmentación en el área de la salud y sistema de recuperación de imágenes basado en contenido o *content based image retrieval system*

. Deep Learning

Las redes neuronales artificiales o *artificial neural network* (ANN) son un modelo computacional inspirado en el comportamiento básico de las neuronas biológicas del cerebro. Las ANN están compuestas por capas que contienen un conjunto de nodos (los cuales representan las neuronas) y a su vez, cada capa de la ANN está conectada a una capa siguiente a través de una conexión neuronal artificial de una capa a otra. Esta conexión permite simular el proceso básico de transmisión de información en el cerebro, permitiendo a las ANN aprender y especializarse automáticamente a una tarea determinada.

Rumelhart et al. [**RHW85**] declara que una ANN se puede descomponer en 3 partes importantes. Primero se tiene una capa de entrada o *input layer* la cual transforma la información de entrada para ser procesada por la red. Esta información se puede interpretar como un impulso que se irá transmitiendo a las distintas capas. La segunda etapa ocurre en las capas escondidas o *hidden layers*, las cuales se pueden describir como funciones no lineales que transforman la representación inicial de los datos de entrada en una representación propia de la red. La tercera y última etapa se centra en la capa de salida o *output layer*. En esta capa, se recoge la representación interna de la red y se

transforma en un resultado capaz de ser procesado por una función de pérdida y/o un humano. Un ejemplo visual se puede observar en la figura 1.1.

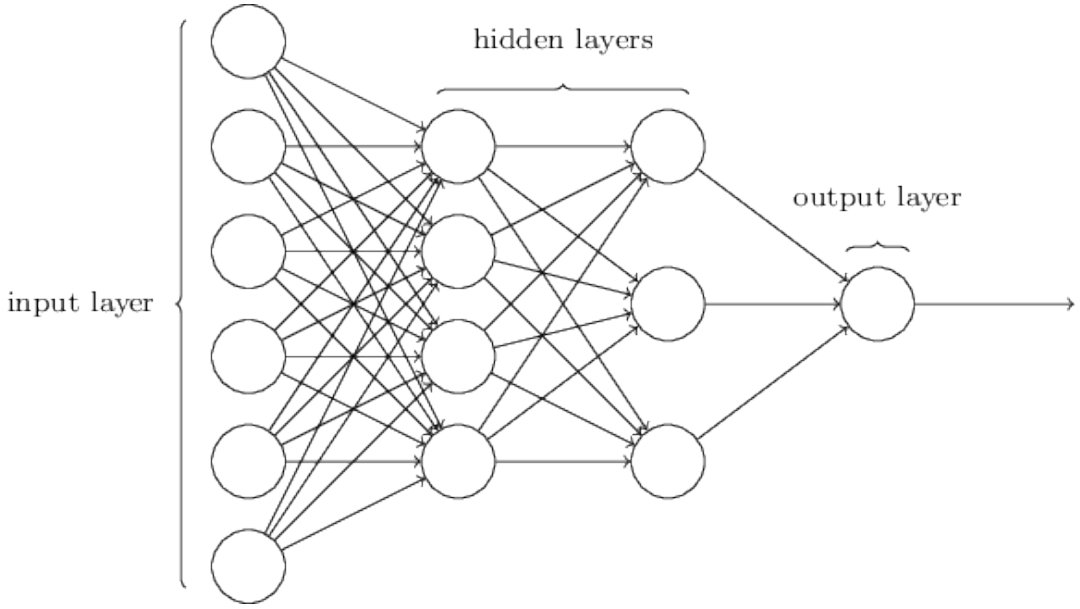


FIGURA 1.1. Modelo básico de una red neuronal artificial.

.1. Componentes de una red neuronal artificial. El componente básico de las redes neuronales es la **neurona** artificial, la cual simula el comportamiento de enviar o no un "impulso" a otras unidades de la red. El primer modelo de neurona artificial fue definido por McCulloch et al. [**MP43**] mediante la siguiente función:

$$N = \sigma\left(\sum_{i=1}^{d} W_i\, X_i - b\right).$$

Los elementos de dicha función son:

- σ = Función de activación.
- X = Vector de datos de entrada representado de forma que la red neuronal pueda aceptar el input.
- W = Vector de pesos. El peso W_i se asocia a una conexión entre dos neuronas y regula la importancia de la transmisión de información entre las capas de la red.
- b = Valor de sesgo que permite oscilar la función de activación en distintos valores.

La **función de activación** σ es un elemento indispensable en redes neuronales ya que introduce propiedades no lineales en el modelo, permitiendo a la red aproximar relaciones complejas entre los *inputs* y los *outputs* que se le entregan. Su objetivo principal es convertir una señal de entrada de una neurona en una señal de salida que se usa

como una señal de entrada en la siguiente capa de la red. Según Hornik et al. [**HSW89**], la función de activación tiene que ser no constante y diferenciable de modo que la red obtenga la propiedad denominada "aproximación universal", es decir, que sea capaz de aproximar cualquier función entre *inputs* y *outputs*. Un ejemplo de este componente puede ser visualizado en la figura 1.2.
Alguna de las funciones de activación más comunes son:

- **Relu:** Usada ampliamente en el área del *Deep Learning*. Esta función asigna a 0 todos los valores de entrada menores a 0 y es lineal cuando los valores de entrada son mayores a 0.

$$\sigma(x) = max(x, 0).$$

- **Sigmoid:** Una función ampliamente usada en los inicios del *Deep Learning*, famosa por su forma de "S" y entregar valores máximos entre -1 y 1.

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

- **Tangente Hiperbólica:** Variación de la función *Sigmoid*. Su rango de normalización es de $[-1, 1]$. La ventaja sobre la función *Sigmoid* es que las entradas negativas se mapearán a valores fuertemente negativos y las entradas cero se mapearán más cerca de cero.

$$\sigma(x) = tanh(x) = \frac{2}{1 + e^{-2\,x}} - 1 = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}.$$

- **Softmax:** Función de activación logística más generalizada para la clasificación multiclase. Esta función es necesaria para que las probabilidades condicionales obtenidas por las neuronas en las capas de salida sumen 1.

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}.$$

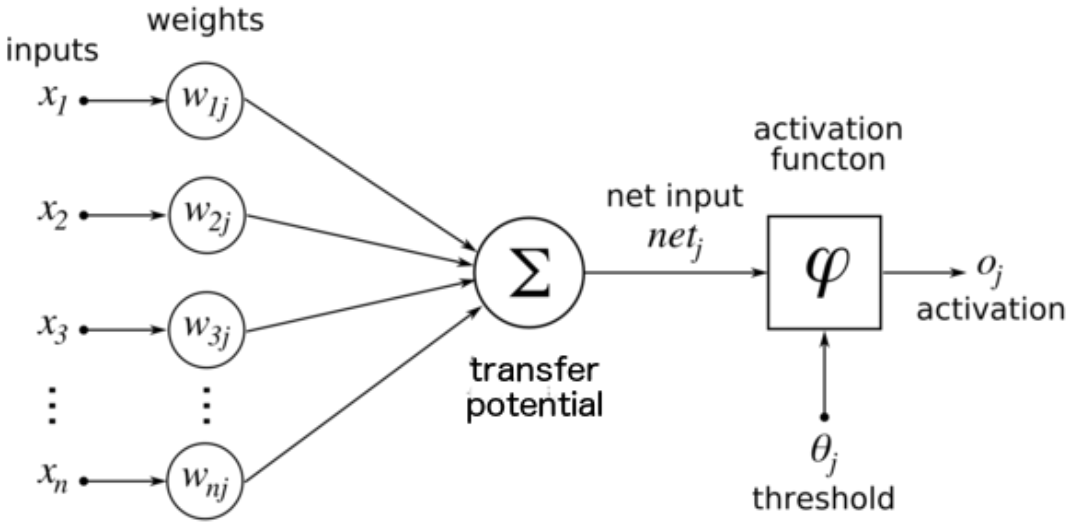


FIGURA 1.2. Modelo básico que ejemplifica el comportamiento de una neurona con función de activación φ.

.2. Redes *feedforward*. La idea detrás de este tipo de red es que el conocimiento o información que entra a la red tiene que ser transmitido ordenadamente de capa en capa hacia la salida, por lo cual, las neuronas no tienen ciclos ni *loops* entre sí y solo ocurre conexión entre capas vecinas. Este tipo de red se define especificando:

- Una capa de entrada donde se representan los datos de entrada como un vector de información.
- Una cantidad N de capas ocultas donde se procesa internamente la información.
- Una capa de salida donde se le da "sentido" al resultado obtenido por la red para que este sea entendido por los humanos.

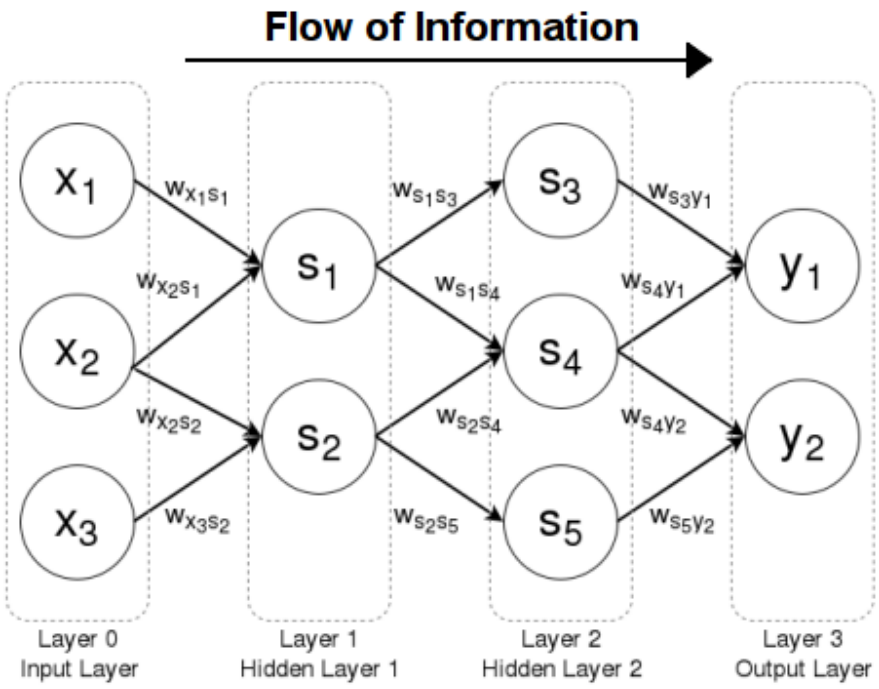


FIGURA 1.3. Modelo que ejemplifica una Red Feedforward.

El entrenamiento de una red neuronal feedforward es importante, ya que en este proceso se ajustan los pesos de las neuronas y dependiendo de este ajuste la red neuronal logrará resolver el problema para cual fue construida o no. El método utilizado actualmente para realizar este ajuste de pesos se conoce como ***Backpropagation*** y su expresión matemática es:

$$\Delta w_{ij} = \eta \, \phi_j \, y'_i.$$

- w_{ij}^k: Peso w de la neurona j a la neurona i.
- η Factor de aprendizaje, con valores $]0, 1[$.
- ϕ Error de la neurona j.
- y'_i Output de la neurona i o estimador de i.

Podemos descomponer la variable ϕ_j en 2 tipos, una para neuronas en la última capa o capa de salida:

$$\phi_j = (y - y') \, f'_j(total_j).$$

y una para las neuronas en una capa escondida:

$$\phi_j = f'_j(total_j) \sum_{l \in L} (\phi_l w_{jl}).$$

- y es el output esperado.
- f'_j derivada de la función de activación.
- L Conjunto de neuronas de la siguiente capa.
- $total_j$ Ponderación de los y' de la capa anterior.

Una red neuronal realiza varias iteraciones de la regla antes mencionada para terminar su entrenamiento, con el objetivo llegar a parámetros ideales que le permitan realizar la tarea para cual fue diseñada.

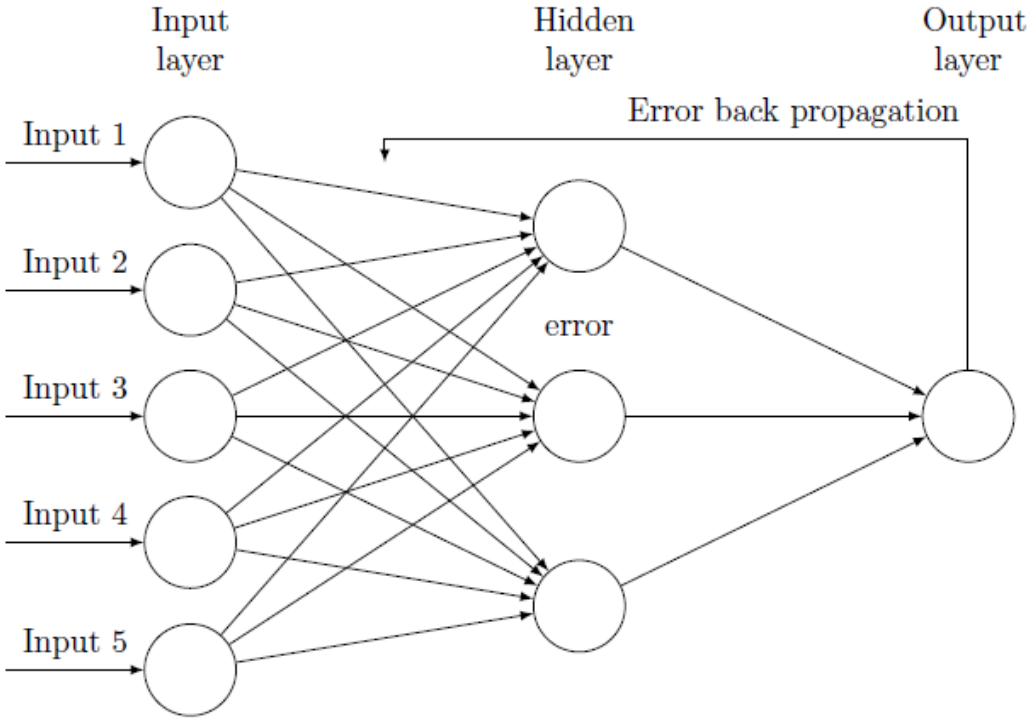


FIGURA 1.4. Modelo que ejemplifica el algoritmo de *backpropagation*.

.2.1. Redes Convolucionales (CNN). Estas redes son especializadas en problemas relacionados con la computación visual, por lo cual tienen un gran desempeño en clasificación de imágenes y reconocimiento de patrones. Además, no se quedan atrás en otras áreas, como por ejemplo, sistemas recomendadores o sistemas generadores de imágenes [**Kno18**].

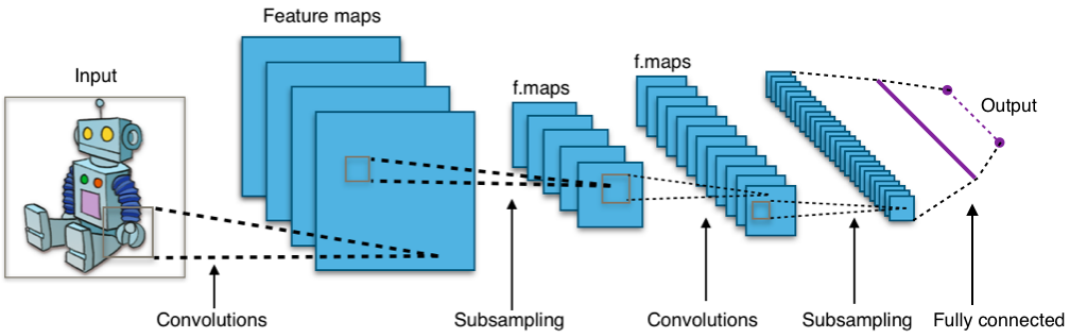


FIGURA 1.5. Esquema completo de una red convolucional.

Las CNN comparten muchas características de las redes feedforward, incluyendo la existencia de neuronas, capas y pesos. La gran diferencia es que los datos de entrada en una CNN pueden ser multidimensionales. En el caso de una imagen natural, esta tendrá 3 dimensiones, las cuales son el alto, el ancho y los canales de colores [1]. La gran diferencia que tienen las CNN es la implementación de una capa de convolución. Esta capa toma una cierta cantidad fija de píxeles de la imagen, predeterminados por una ventana de un cierto tamaño, y realiza una convolución con una cierta matriz de pesos para determinar la activación de una neurona de la capa. Este proceso se repite hasta que la ventana recorre toda la imagen en todos sus canales, generando un conjunto de activaciones conocido como mapa de activación, donde este mapa se transforma en la entrada de la próxima capa de la red. Además de las capas de convolución, una CNN típica también incluye una nueva capa denominada capa de *pooling*, la cual se usa para reducir las dimensiones espaciales del patrón de entrada, pero no la profundidad [2]. Esta capa tiene como finalidad:

- Reducir las dimensiones de la imagen.
- Al tener menos dimensiones se tiende a generar un representación de los datos de entrada más compactos y con la misma información, reduciendo el *overfitting* de la red.
- Una representación reducida permite a la red descubrir características de bajo nivel mientras se reduce la dimensionalidad para encontrar características de alto nivel al final del procesamiento de la imagen.

[1] Nos referimos a los distintos colores de los píxeles de una imagen, normalmente son rojo, verde y azul, pero también existen imágenes con otros canales y colores

[2] Esto se refiere a que no se afecta los canales de la imagen.

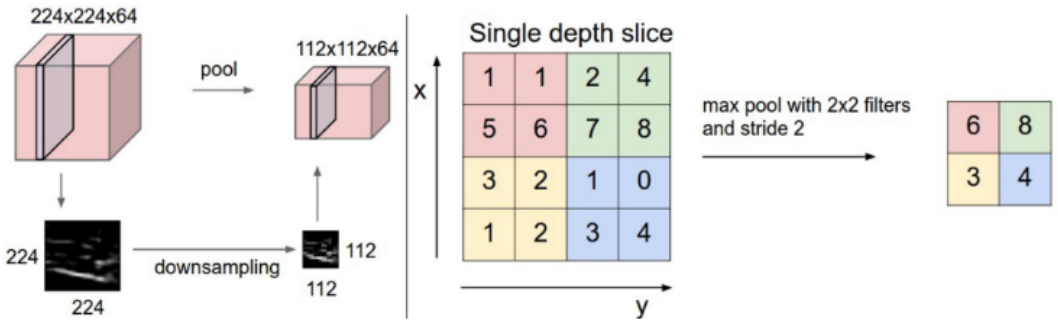


Figura 1.6. Esquema que ejemplifica el comportamiento de una capa de *pooling* sobre una imagen.

. Clasificación

Esta tarea tiene como objetivo aprender un mapeo de un patrón de entrada a una etiqueta o *label* de salida, más conocido como *ground truth*. Este objetivo se logra al entrenar un modelo inteligente a través de un conjunto de datos, los cuales le entregan la experiencia necesaria al modelo para realizar su función. Para desarrollar la tarea de clasificación de manera **supervisada**, se debe tener un conjunto de datos de entrada $x \in \mathbb{X}$ con una cierta dimensionalidad $d \in \mathbb{R}^3$ y una etiqueta *ground truth* $y \in \mathbb{Y}$. Este conjunto de datos tiene que cumplir la distribución de probabilidad $p(x)$ y $p(y|x)$. La clasificación de manera supervisada es calcular la probabilidad condicional de $p(y|x)$ de un subconjunto de datos $S = \{\mathbb{X}, \mathbb{Y}\} = \{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$ siendo $(x_i, y_i) \sim p(x, y) \forall i \in N$, siendo $N = [c_1, c_2, ..., c_n]$ la cantidad de clases.

. Segmentación Semántica

Esta tarea tiene como objetivo aprender un mapeo de una imagen de entrada a una imagen binaria o máscara binaria de salida, la cual al ser sobrepuesta con la imagen original, se extrae o resalta una zona de interés o *region of interest* (ROI). Este objetivo se logra mediante un entrenamiento **supervisado** en el cual se tiene un conjunto de imágenes de entrada $x \in \mathbb{X}$ y un conjunto de máscaras binarias $y \in \mathbb{Y}$ donde cada mascara binaria y_i representa la ROI de la imagen x_i.

. Content based image retrieval system

Estos sistemas utilizan información y características únicas de una cierta imagen con el propósito de buscar y recuperar otras imágenes similares de una base de datos. Para realizar esta tarea, se transforman las imagenes a vectores llamados *embeddings* y usando dicho vector se buscan otros vectores similares en la base de datos para luego coincidir cada vector con su respectiva imagen y retornar los resultados. Las técnicas clásicas para construir dichos *embeddings* se basan en la información de color, textura y morfología, logrando abarcar problemas como el reconocimiento facial, pronostico de climas o incluso la prevención de crímenes [**CRC$^+$14**]. En la actualidad, la popularidad del *deep learning* y en especifico las CNN, han llevado a expertos a utilizar estos modelos para mejorar los CBIRS[**SPK19**]. Para lograr la transformación de imagen a *embedding* se obliga a las redes a aprender una cierta tarea, las cuales pueden ser **tarea supervisada**, **tarea no supervisada** o el punto medio conocido como **tarea semi-supervisada**, y utilizar como *embedding* resultante alguna de las capas de activación de dichas redes.

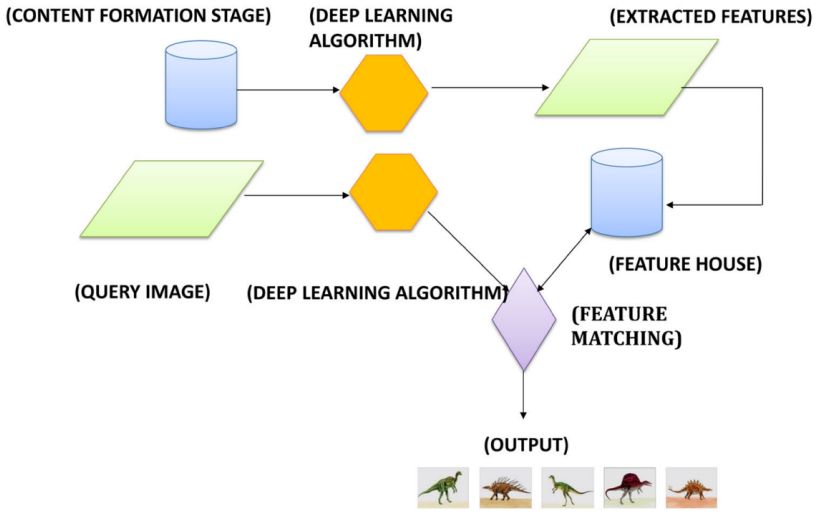


Figura 1.7. Ilustración representativa de un CBIRS basado en *deep learning*.

.1. CBIRS con Tarea supervisada. Esta modalidad de generación de *embeddings* trata de enseñar a la maquina una tarea que requiera de una etiqueta o un valor predeterminado, es decir, para todo $x \in \mathbb{X}$ existe un valor conocido $y \in \mathbb{Y}$ el cual la red tiene que predecir. Posteriormente, una vez que la maquina este entrenada, se puede extraer una de las capas de activación de dicha red para representar la información compactada de la imagen y a así, crear un *embedding* representativo que luego se ocupará para la búsqueda de imágenes.

.2. CBIRS con Tarea no supervisada. Para esta modalidad, la máquina tiene que aprender sin etiqueta a reconstruir la imagen de entrada, es decir, el *input* de la

red es un conjunto de datos $x \in \mathbb{X}$ y la salida de la red es la reconstrucción aproximada de los datos de la entrada $\hat{x} \in \mathbb{X}$. Para generar el *embedding* que se utilizará en el sistema, se emplea una representación compactada de alguna de las capas de la red, donde tradicionalmente se extrae la capa "intermedia" o *bottleneck*, la cual, es una capa pequeña con pocos nodos que compacta la información de entrada.

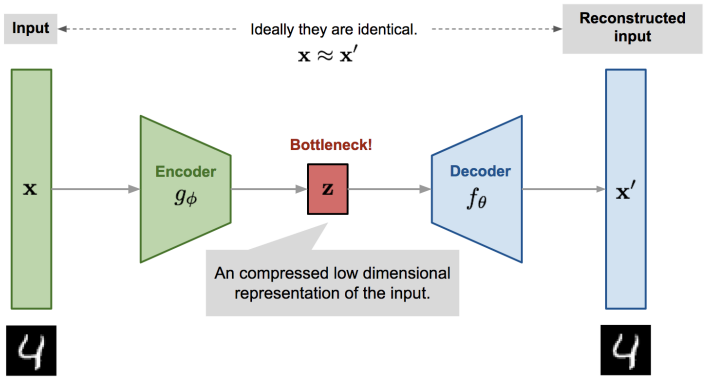


FIGURA 1.8. Ilustración de una estructura *encoder-decoder* donde se utiliza un *bottleneck* para reducir la dimensionalidad de los datos de entrada.

(Fuente: plataforma [**Ire20**])

.3. CBIRS con Tarea semi-supervisada. Esta modalidad tiene dos conjuntos de datos, un conjunto de datos etiquetados D_l y otro conjunto de datos no etiquetados D_u y el objetivo de estos sistemas es utilizar el conjunto D_u para mejorar el desempeño o la calidad de construcción de los *embeddings* de las redes entrenadas utilizando el conjuntoD_l. Para trabajar con esta modalidad, es necesario definir supuestos adicionales que vinculan las propiedades de la distribución de las características de los datos entrada a las propiedades de la función de decisión que las redes tiene que aprender [**CZ05**], estos incluyen el ***supuesto de suavidad*** [3], el ***supuesto de agrupamiento*** [4] y el ***supuesto de baja densidad*** [5] [**CdBP19**].

. Transfer Learning

En el área de *deep learning*, es posible transferir y re-utilizar el conocimiento de una tarea a otra si comparten alguna similitud en el dominio a trabajar, esta técnica es conocida como *transfer learning* [**PY09**].

[3] Las muestras cercanas en el espacio de características probablemente sean de la misma clase

[4] Es probable que las muestras en un agrupamiento sean de la misma clase

[5] Es posible que los límites de clase se encuentren en áreas del espacio de características que tienen menor densidad que los clústeres

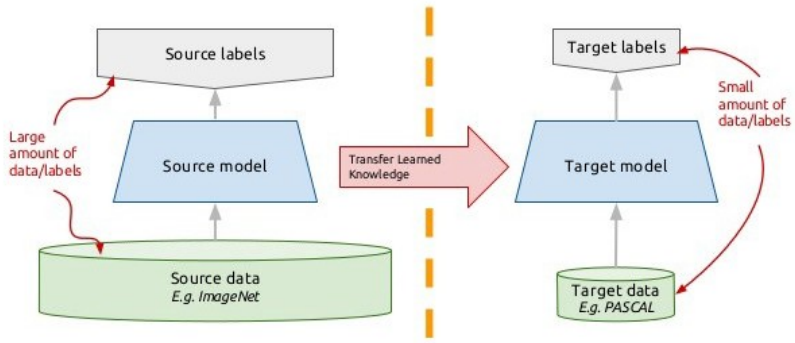


FIGURA 1.9. Ilustración de la técnica de *transfer learning*.
(Fuente: plataforma [**McG20**])

.1. *Transfer Learning* en imágenes. Para transferir el conocimiento de una red entrenada en un problema de *computer vision* se tiende a utilizar la información adquirida por redes altamente profundas, por lo cual las arquitecturas como VGG [**SZ14**], ResNet [**HZRS16**], InceptionV3 [**SVI$^+$16**] y AlexNet [**Kri14**] entrenadas en grandes conjuntos de datos como ImageNet [**DDS$^+$09**] son frecuentemente utilizadas en la comunidad para realizar *transfer learning* por su capacidad de aprender información general de un problema. Para realizar la transferencia de conocimiento de una tarea a otra, se toma la representación de las capas convolucionales de la red con sus pesos ya entrenados y se eliminan o reemplazan las capas finales para ajustar la red a los nuevos problemas a tratar, posteriormente, se vuelve a entrenar el modelo congelando o no las capas ya pre-entrenadas para completar la transferencia de información.

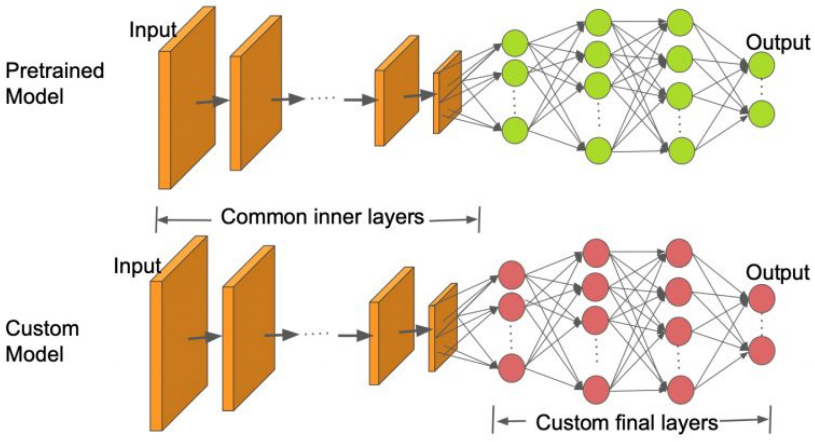


FIGURA 1.10. Ilustración de la técnica de *transfer learning* para imágenes, en especifico a la tarea de clasificación.
(Fuente: plataforma [**Nay20**])

.2. *Transfer Learning* en Texto. En el área de NLP se utiliza una vectorización densa de texto denominada *embedding*. Estos vectores altamente densos en información son el resultado de una compresión de información proporcionada por arquitecturas como los *autoencoder*, a través de una compresión por una capa *bottleneck*. Estos vectores se conocen como *char embedding*, *word embedding* y *sentence embedding*. Existe una gran cantidad de *text embeddings*, donde *Word2Vec* es un ejemplo popular del resultado de esta técnica de compresión [**MSC+13**]. También se pueden construir *embeddings* combinando la salida de la red y generar co-ocurrencia de las palabras, logrando generar *embeddings* como *GloVE* [**PSM14**].

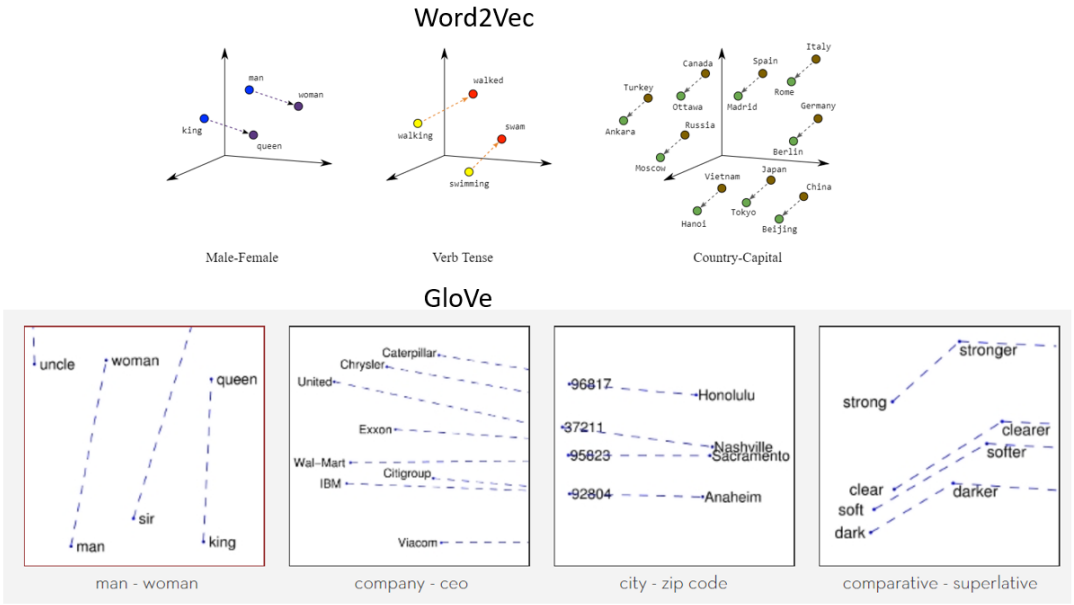


FIGURA 1.11. Ilustración representativa de *word embeddings*. Cada par de vectores de palabras similares está relativamente próximo uno del otro en el espacio proyectado.

(Fuente: plataforma [**Kha19**])

. *Data Augmentation*

En el mundo del aprendizaje automático es común no contar con suficientes imágenes para entrenar un modelo. Para solucionar este problema se recurre a esta técnica la cual nos permite generar de manera infinita imágenes similares pero distintas y gracias a estas pequeñas distinciones, se consideran otras imágenes.

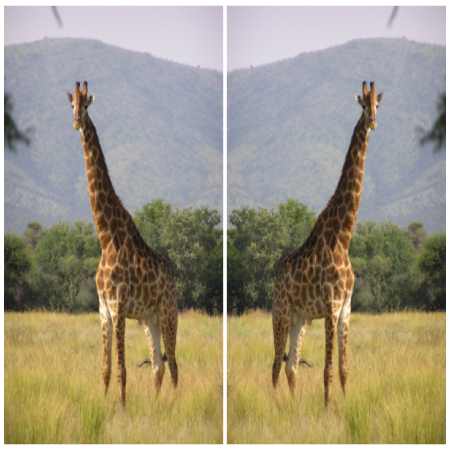


FIGURA 1.12. Ilustración representativa de *data augmentation*. A la imágen original se le aplico una transformación de reflexión.

(Fuente: plataforma [**Ham**])

.1. Rotación y Reflexión. Estas técnicas tienen la capacidad de generar nuevas imágenes sin pérdida de información, ya que simplemente, re-ordenan la información de la imagen en una posición distinta como se demuestra en la figura 1.13.

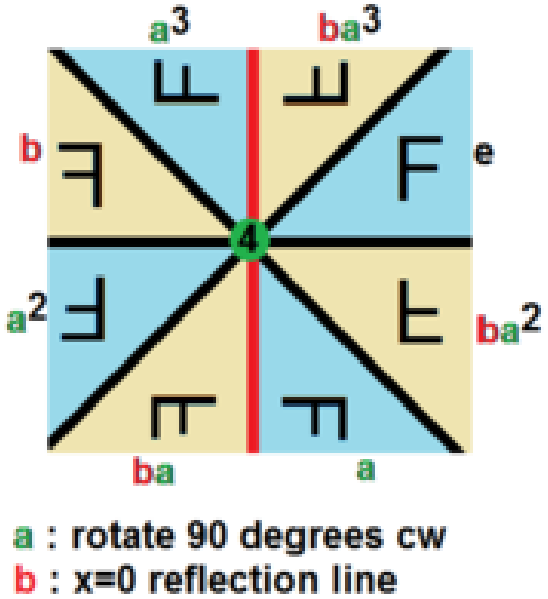


FIGURA 1.13. Ilustración representativa de las posibles reflecciones y rotaciones sin perdida de información en una imagen 2D.

(Fuente: plataforma [**Ham**])

.2. Escala, Cortar, Translación y Ruido. Estas técnicas tiene como cualidad, tomar una cierta sección de la imagen y desechar el resto para así lograr la generación de una nueva imagen. La característica de estas técnicas es que la imagen resultante tiene menor información que la imagen original.

.3. Transferencia de Estilo. Una forma avanzada y más moderna de generar nuevos datos es utilizar las *Generative adversarial network (GAN)*, las cuales pueden transformar una imagen de un dominio en una imagen de otro dominio como se ejemplifica en la figura 1.14.

FIGURA 1.14. Ilustración representativa del resultado de una red GAN al generar nuevas imágenes de ciudades.

(Fuente: plataforma [**Gan21**])

Capítulo 2

Estado del Arte

El siguiente capítulo, tiene como objetivo formalizar matemáticamente el problema previamente presentado, además, exponer todo el trabajo realizado para resolver dicho problema, con diferentes metodologías y técnicas.

. Problema

Introducción: Los CBMIRS tienen como objetivo asistir al personal de la salud, estudiantes de medicina, radiólogos, entre otros, y también, ayudarlos a descubrir casos clínicos similares al realizar una búsqueda mediante una imagen para ayudar a investigar, trabajar y/o acelerar el proceso del diagnóstico de un paciente. Sin embargo, la cantidad de imágenes médicas generadas en hospitales y centros médicos han incrementado exponencialmente en los últimos años, provocando que sistemas de búsqueda tradicionales basados en descriptores o *metadata* dejen de ser la opción más viable para este problema debido a su incapacidad de lidiar con altos volúmenes de datos, la existencia de información nula o mal ingresada y la falta de diagnostico o anotaciones en las imágenes. Este problema, ha llevado a la comunidad científica a utilizar estrategias más modernas como el *Deep Learning* con la finalidad de extraer automáticamente descriptores de las imágenes y generar *embeddings* que posteriormente se utilizan en la indexación y futura búsqueda [**LKB+17**]. Desafortunadamente, debido a problemas como el alto costo de etiquetar imágenes médicas, la complejidad de los múltiples tipos de imágenes y la alta volatilidad de los *dataset* en los hospitales o centros de salud, se ha dificultado la transición desde los tradicionales CBMIRS basados en *metadata* hacia los nuevos sistemas basados en *deep learning* [**CHBJ+18**].

. Generación de *Image Embedding*

Tradicionalmente el proceso de transformar una imagen a un *embedding* se realizaba a través del análisis de color, textura o morfología. Hoy en día, estas técnicas no son lo más recomendado para tratar con imágenes altamente complejas como son las imágenes médicas. Las redes neuronales tienen la habilidad de comprender de una manera más eficiente la información de imágenes altamente complejas y transformarlas en *embeddings* más ricos en información que su contraparte clásica. Para realizar esta tarea, existen tres tipos de aprendizaje de redes neuronales que permiten generar modelos para construir *embeddings* ricos en información a partir de imágenes, los cuales son:

- ***Supervised Learning*** es el método de entrenamiento más conocido en el ámbito de las redes neuronales. Su funcionamiento se basa en que cada imagen tiene asociado una etiqueta y la red aprende una función de mapeo que transforma la imagen en dicha etiqueta. Algunos de los CBMIRS basados en esta metodología son:
  Owais et al.[**OACP19**] el cual introdujo un CBMIRS capaz de clasificar y recuperar con una métrica F1 entre 81 % y 82 %, 50 tipos de clases distintas, estas incluyen imágenes de distintos órganos, enfermedades, formato (Rayos-X, CT, MRI, Ultrasonido, Etc) y tamaño. El resultado obtenido para la media armónica posiciona este trabajo como el estado del arte en el año de publicación para el problema de múltiples clases.
  Cai et al. [**CLQ$^+$19**] utilizó una red siamesa para generar un *embedding* binario a través de hashing y construyó una función de perdida especial que pérmite distanciar o aproximar dicho *embedding* si el par de imágenes de entrada a la red tienen la misma etiqueta o no.
  Por ultimo, uno de los trabajos más recientes es de P. Yang et al. [**YZL$^+$20**] el cual logró el estado del arte en el dataset publico histológico *Kimia Path24* [**BKS$^+$**] utilizando un mecanismo de atención mixto que involucra tanto atención espacial como atención por canal y entrenando el modelo con una pérdida de similitud múltiple bajo la supervisión de información categórica para generar el *embedding*.
- ***Unsupervised learning*** es lo opuesto del *Supervised Learning*, por lo cual, en vez de forzar a la red neuronal a aprender una función de mapeo de imagen a etiqueta, la red aprende a reconstruir la imagen original sin necesidad de etiquetas.
  Pinho et al. [**PSC19**] estudió la factibilidad de un entrenamiento no supervisado a través de un modelo *Encoder-Decoder* y un discriminador para generar un sistema de búsqueda que no utilice etiquetas previas en imágenes de torax 3D, obteniendo resultados poco concluyentes en su investigación, demostrando la dificultad de construir este tipo de sistemas e incentivando a la comunidad a implementar otras técnicas, además del método no supervisado de entrenamiento de redes neuronales.

Ahn et al. [**AKF+19**] desarrolló un framework basado completamente en *Unsupervised learning* tanto en pre-entrenamiento como en entrenamiento utilizando *kernel learning* y *deep learning* correspondientemente. Su método logró obtener el estado del arte en técnicas no supervisadas de recuperación de imágenes y tuvo un desempeño igual a técnicas supervisadas de recuperación y clasificación de imágenes médicas en el dataset *IRMA* [**LSK+03**], *ImageClef* [**DHBSM16**] y *ISIC* [**CGC+18**].

- ***Semi-Supervised Training*** es el punto medio entre *Supervised Learning* y *Unsupervised learning*. Los CBMIRS basados en *Semi-Supervised Training* se basan en el uso de entrenar y desarrollar modelos usando datos etiquetados D_l mientras se utilizan datos no etiquetados D_u para mejorar el desempeño de los modelos ya entrenados. En la práctica, existen dos formas típicas de implementación de este método de aprendizaje, la primera forma es un sistema compuesto por distintos módulos que trabajan con técnicas tanto supervisadas como no supervisadas donde los módulos no supervisados alimentan y/o regularizan los módulos supervisados para generar buenos *embeddings*. Un ejemplo de esto es la propuesta de Şaban Öztürk [**Özt20**] el cual construyó un sistema en módulos que ocupa una sección supervisada y un autoencoder no-supervisado para generar códigos IRMA [**LSK+03**] los cuales son códigos únicos de imágenes médicas para posteriormente utilizarlos en búsquedas de imágenes.
  La segunda forma de construir un sistema semi-supervisado es cuando el sistema no esta fragmentando en pequeños módulos, si no mas bien, trabaja como un módulo completo de tal forma que el aprendizaje del modelo toma en cuenta los datos etiquetados y los no etiquetados aprendiendo de la información de ambos conjuntos de datos. Wei et al. [**WMQQ16**] utilizó este método al enfocarse únicamente en mejorar la métrica de distancia para los sistemas de recuperación de imágenes médicas, empleando técnicas tradicionales como el *kernel trick* logran que su modelo aprenda a calcular la distancia de Mahalanobis, permitiendo manejar tanto información visual como semántica.

. Indexación

Los Esquemas de indexación adecuados son altamente necesarios para tener un sistema de búsqueda de imágenes rápido y eficiente. En CBIRS sobresalen dos técnicas de indexación: indexación de archivos invertidos e indexación basada en hash [**ZLT17**], pero no se descartan otros métodos.

.1. Indexación de archivos invertidos. Inspirado en el resultado de sistemas de búsquedas de texto, la estructura de archivos invertidos es una representación compacta en columnas de una matriz dispersa, donde las filas y la columnas denotan imágenes y "palabras visuales", respectivamente. En una recuperación de imágenes *on-line* solo deben verificarse aquellas que comparten palabras visuales comunes con la imagen de consulta. Por tanto, el número de imágenes candidatas a comparar se reduce considerablemente, consiguiendo una respuesta eficaz.

.2. Indexación basada en hash. Cuando se tiene una representación vectorial densa de una imagen y dicho vector esta compuesto por coeficientes distintos de cero, no es adecuado aplicar directamente la estructura de archivo invertidos para la indexación. Para lograr una recuperación eficiente de resultados relevantes, las técnicas de hash se adopta para estos casos. El esquema de hash más utilizado es el *localty sensitive hash*, el cual divide el espacio de características con múltiples funciones hash de proyecciones aleatorias, con la intuición que para los objetos que están cerca unos de otros, la probabilidad de colisión es mucho mayor que para aquellos que están lejos.

.3. Indexación Aleatoria. Otro método no tan popular en el mundo de la búsqueda de imágenes es la indexación aleatoria. Este tipo de indexación es utilizada en el área del NLP donde es un método ligero de reducción de dimensiones, que se utiliza, por ejemplo, para aproximar relaciones semánticas vectoriales en sistemas de procesamiento de lenguaje en línea, para realizar búsquedas de elementos similares. Esta capacidad de aproximación se puede utilizar al realizar búsquedas por generación de vectores aleatorios, como por ejemplo utilizando **Random Projection Tree**.

. Trabajo Relacionado

El área de la imagenología médica es bastante interesante para la comunidad científica, generando un basto numero de investigaciones y trabajos. Los más interesantes desde el punto de vista de este trabajo son:

.1. Segmentación. Este problema se enfoca en la búsqueda de una una zona de interés (ROI) en distintos tipos de imágenes médica. La dificultad radica en que los ROIs son difíciles de identificar incluso para ojos expertos, por lo cual se han desarrollado herramientas automáticas para ayudar a los médicos y tecnólogos a realizar su diagnóstico.
Gi et al. [**GCF$^+$19**] crearon *Context Encoder Network* (CeNet) la cual logró el estado del arte en octubre del 2019 en segmentación del disco óptico, detección de vasos, segmentación pulmonar, segmentación del contorno celular y segmentación de la capa de tomografía de coherencia óptica retiniana. Esto se logró mediante la combinación de la capacidad de segmentación de la U-net [**RFB15**] y la implementación original de una sección de extracción de contexto que combina dos nuevos bloques convolucionales

llamados *dense atrous convolution* (DAC) y *residual multi-kernel pooling* (RMP).

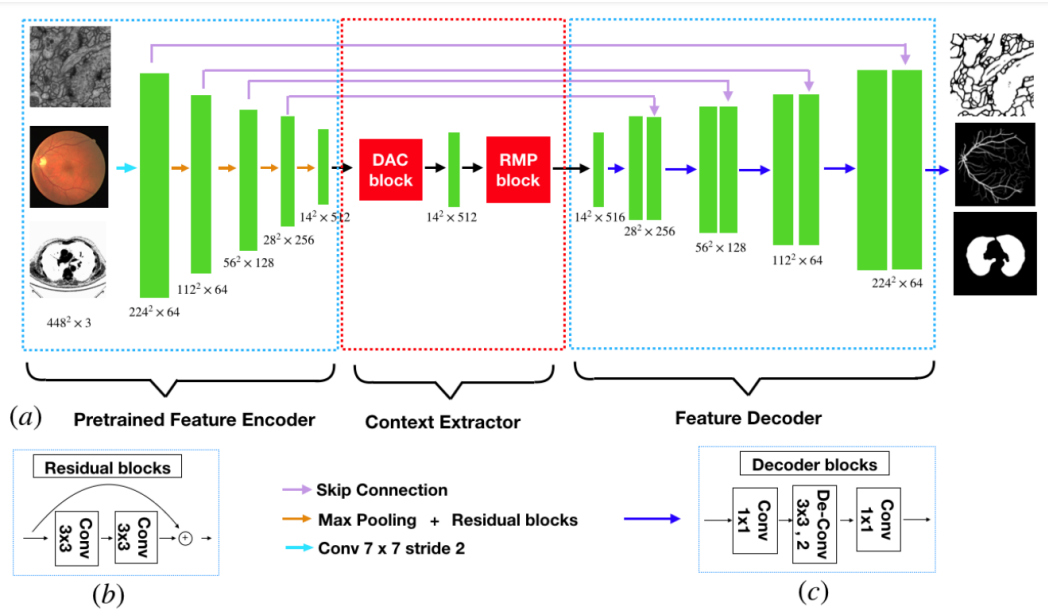


FIGURA 2.1. Ilustración de CeNet. Primero, las imágenes son introducidas en el módulo encoder compuesto por una ResNet-34 pre-entrenada en Imagenet. El extractor de contenido es propuesto para generar mapas de activación de alto nivel. Este módulo esta compuesto por el bloque DAC y RMP. Finalmente, el mapa de activación pasa por el decoder que incluye las *skip connection* originales de UNet.

(Fuente: Paper [**GCF+19**])

El **Módulo de extración de contenido** de CeNet está compuesto por los bloques DAC y RMP. El bloque DAC se origina en la idea de la técnica *atrous convolution*, la cual adopta la filosofía de múltiples *pooling layers* para prevenir la pérdida de información semántica. Esta técnica se originó para el cálculo eficiente de transformaciones de onda, donde la expresión matemática es la siguiente:

$$y[i] = \sum_k x[i + rk]\dot{w}[k].$$

Donde el mapa de activación x y el filtro w producen la salida y donde el *atrous rate* r corresponde al paso con el que muestreamos la señal de entrada. Esta operación es equivalente a realizar una convolución con un filtro muestreado producido insertando $r-1$ ceros entre dos valores de filtros consecutivos a lo largo de cada dimensión espacial[1]

[1]de ahí el nombre *atrous convolution* en el que la palabra francesa atrous significa agujeros en inglés.

y donde la convolución normal es un caso especial de esta técnica donde $r = 1$. La atrous convolution nos permite cambiar el tamaño de visión simplemente variando el valor de r como podemos ver en la figura 2.2.

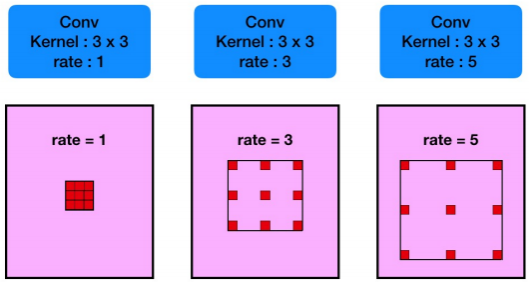


FIGURA 2.2. Ilustración demostrativa de la **atrous convolution** con un valor de $r = 1$, $r = 3$ y $r = 5$.

(Fuente: Paper [**GCF+19**])

Los autores tomaron la idea de *atrous convolution* y la cualidad de las redes **Inception** creadas por Google para finalmente producir el módulo DAC. El bloque ***Dense Atrous Convolution*** es el compuesto de 4 ramas en cascada con un grado incremental de *atrous convolution* como se ejemplifica en la figura 2.3

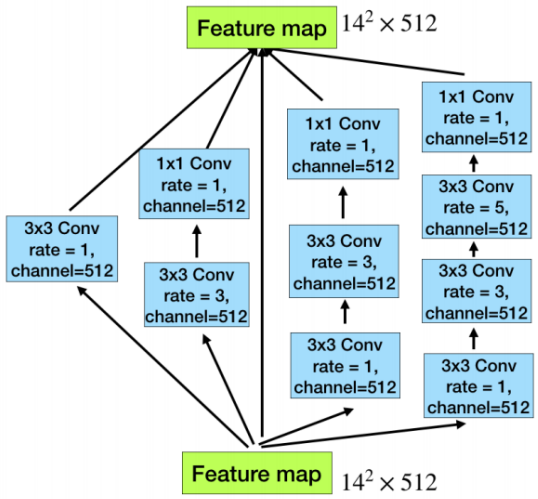


FIGURA 2.3. Ilustración del bloque DAC. Esta ilustración muestra cuatro ramas con un incremento gradual en el número de atrous convolution desde 1 hacia 1, 3 y 5 respectivamente. Permitiendo a la red extraer características de distintas escalas.

(Fuente: Paper [**GCF+19**])

El bloque RMP resuelve el problema de la volatilidad del tamaño de objetos en imágenes médicas, por ejemplo un tumor puede ser mucho más grande o más pequeño desde un examen a otro. El bloque RMP utiliza cuatro **pooling layers** distintas como se muestra en la figura 2.4. Estos bloques son transformados a las mismas dimensiones que el mapa de activación de entrada por una red convolucional 1×1 y finalmente, son concatenados al mapa de activación original.

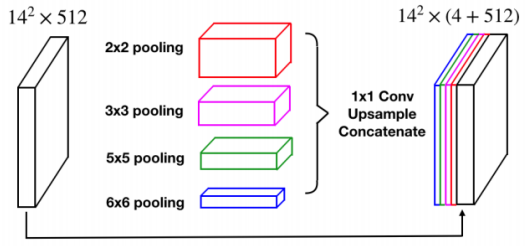


FIGURA 2.4. Ilustración del bloque RMP. Se propone un extractor de contenido compuesto de 4 *pooling layers* de distinto tamaño donde posteriormente, son alimentadas a una capa convolucional 1×1 para reducir la dimensionalidad del mapa de activación y finalmente, se realiza una concatenación junto al mapa de activación original.
(Fuente: Paper [**GCF+19**])

Baur et al.[**BAN17**] logró modificar la red de segmentación UNet para poder ser entrenada mediante el uso de metadata de las imágenes, y gracias a esto, creó el concepto de *Auxiliary Manifold Embedding Loss* para entrenamiento semi-supervisado de redes neuronales junto con la ayuda de su segundo aporte científico llamado *Random Feature Embedding*.

Auxiliary Manifold Embedding Loss $\mathcal{L}_E$ es una función de pérdida que permite minimizar la discrepancia entre datos similares en el espacio latente, esta función esta definida como:

$$\mathcal{L}_{E_l}(X, A) = \sum_{i}^{n_E} \sum_{j}^{n_E} \begin{cases} d(h^l(x_i), h^l(x_j)) & \text{if } a_{ij} = 1, \\ max(0, m - d(h^l(x_i), h^l(x_j))) & \text{if } a_{ij} = 0 \end{cases}$$

permitiendo escribir la función de pérdida total como:

$$\mathcal{L} = \mathcal{L}_P + \sum_{l} \lambda_l \cdot \mathcal{L}_{E_l}$$

donde $\mathcal{L}_P$ es una función de pérdida principal como la *Dice Loss* y λ_l es un parámetro de regularización asociado con el *embedding loss* E_l en la capa oculta l.
Teniendo en cuenta que $h(\cdot)$ es el espacio de características latentes y teniendo el conjunto de entrenamiento $D = \{X, Y\}$ siendo los conjuntos de datos etiquetados X_L y no etiquetados X_U, donde el total de datos $X = \{X_L \cup X_U\} = \{x_1, x_2, ..., x_{N_L}, x_{N_{L+1}}, ..., x_{N_{L+U}}\}$

y nuestras etiquetas correspondientes $Y = \{y_1, y_2, ..., y_L\}$, la función del *embedding loss* apunta a reducir la distancia entre similares representaciones latentes $h^l(x_i)$ y $h^l(x_ij)$ de datos vecinos x_i y x_j o a incrementar la distancia entre dichos espacios si se supera un margen de distancia máximo m.
Lo interesante de esta función de pérdida es el uso de la matriz A, una matriz de adyacencia entre todos los *embeddings* en un lote de entrenamiento, la cual se obtiene calculando una métrica de distancia arbitraria d. Esta matriz nos permite incorporar otro tipo de información que no requiere etiqueta al entrenamiento de la red, como por ejemplo, metadata de la misma imagen.
Random Feature Embedding se creó para solucionar el problema del costo computacional que es comparar cada espacio latente creado en cada capa de la red al calcular la función de perdida. Esta técnica toma aleatoriamente un espacio reducido del mapa de activación de cada capa para realizar los cálculos correspondiente. Según los autores, la función de pérdida sigue siendo válida ya que el *backpropagation* solo se calcula y cambia en las secciones seleccionadas por el *Random Feature Embedding*.

.2. Clasificación. Este problema típicamente radica en detectar si una cierta imagen médica presenta una condición específica o no. Chollet el al. construyó en el 2017 la famosa red *Xception* [**Cho17**]. La genialidad detrás de esta red es utilizar una versión mejorada de la convolución clásica, llamada *depthwise separable convolution operation*, mejorando los resultados de clasificación de *Inception V3* [**SVI$^+$16**] en el dataset *ImageNet* [**DDS$^+$09**], considerando que *Inception V3* fue diseñada para este problema.
Xception crea el punto medio entre la convolución tradicional y su versión *depthwise* mediante la expansión del concepto de inception block. La figura 2.5 representa un *inception block* tradicional el cual analiza las correlaciones entre canales a través de un conjunto de convoluciones 1×1, mapeando los datos de entrada en 3 o 4 espacios separados que son más pequeños que el espacio de entrada original, y luego mapea todas las correlaciones en estos espacios 3D más pequeños a través de convoluciones regulares 3×3 o 5×5. La idea de este proceso es realizar la convolución sin la necesidad de sobrecargar un kernel convolucional con las tareas de mapear correlaciones entre canales y correlaciones espaciales.

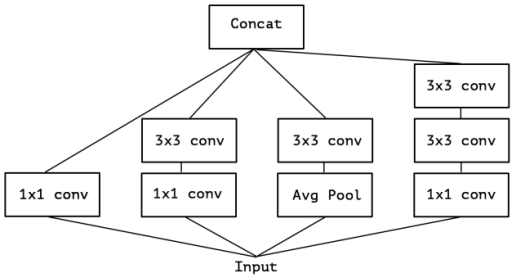


FIGURA 2.5. Ilustración que representa un inception block típico de la red *Inception V3*.

(Fuente: Paper [**Cho17**])

Por otro lado, la operación *depthwise convolution* y su segunda parte conocida como *pointwise convolution* tienen el mismo objetivo, reducir el estrés total del kernel convolución y por defecto, acelerar el proceso de convolución de las redes, tal como se ilustra en la figura 2.6.

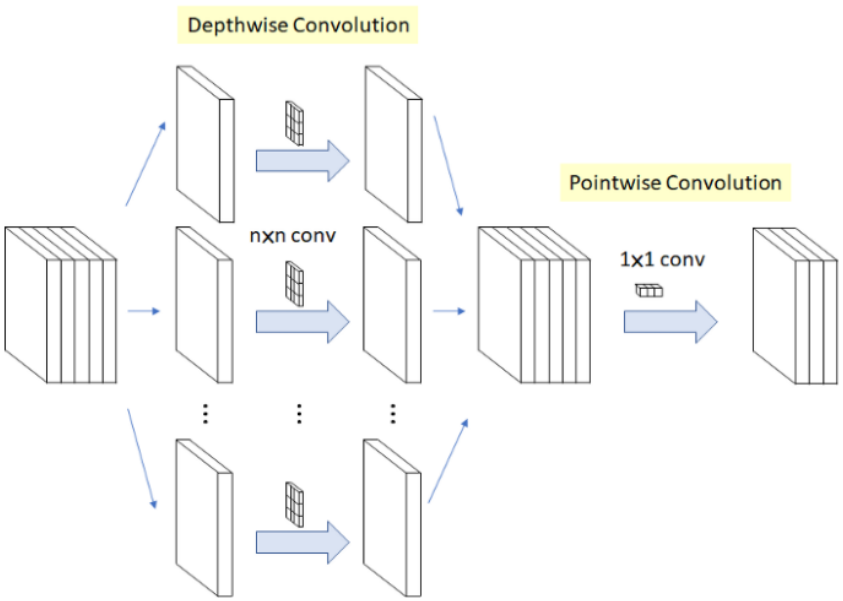


FIGURA 2.6. Ilustración que representa el proceso total de la operación *Depthwise Convolution*.

(Fuente: Plataforma [**Tsa19**])

Xception nos entrega la operación *"Extreme" Inception module* la cual utiliza una convolución 1×1 para mapear correlaciones entre canales, para posteriormente mapear

por separado las correlaciones espaciales de cada canal de salida como representa la figura 2.7. Esta convolución tiene dos características importantes:

El orden de las operaciones: las *depthwise separable convolutions* como se implementan habitualmente (por ejemplo, en TensorFlow) realizan la primera convolución espacial por canal y luego realizan la convolución 1×1, mientras que Inception realiza primero la convolución 1×1.
La presencia o ausencia de una no linealidad después de la primera operación: En Inception, ambas operaciones van seguidas de una no linealidad de ReLU, sin embargo, las *depthwise separable convolutions* se implementan normalmente sin dicha no linealidades.

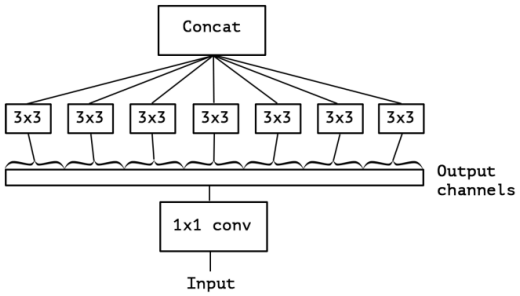


FIGURA 2.7. Ilustración que representa el modulo *Extreme Inception*.
(Fuente: Paper [**Cho17**])

.3. RMPT. El algoritmo *Fast Nearest Neighbor Search With Random Projections* (RMPT) combina el trabajo de Hyvonen et al. [**HPT$^+$16**] y Jaasaari et al [**JHR19**] con el propósito de generar una librería liviana y fácil de usar para la búsqueda aproximada del vecino más cercano. La innovación de este trabajo proviene de solucionar el alto costo computacional y de almacenamiento de generar proyecciones aleatorias en data de alta dimensionalidad, siendo sus soluciones las siguientes:

- En lugar de utilizar vectores densos muestreados a partir de la distribución normal d-dimensional, los autores utilizan vectores dispersos para reducir la complejidad tanto del almacenamiento como del cálculo.
- En lugar de usar diferentes vectores de proyección para cada nodo intermedio de un árbol, los autores usaron un vector de proyección para todos los nodos en el mismo nivel de un árbol, de modo que se pueda reducir aún más el requisito de almacenamiento para maximizar el paralelismo de bajo nivel a través de la vectorización.
- Los autores dividieron los vectores en la mediana para hacer que el rendimiento sea más predecible y permitir guardar los árboles en un formato más compacto.

Para generar la proyección aleatoria, los autores utilizan un sparse random vector $r = (r_1, ..., r_d)$, donde los componentes son muestreados desde una distribución normal estándar con probabilidad α y los ceros con una probabilidad $1 - \alpha$:

$$r_i = \begin{cases} N(0,1), & \text{Con probabilidad } \alpha \\ 0, & \text{Con probabilidad } 1 - \alpha \end{cases} \tag{2.1}$$

Respecto a la búsqueda de vecinos cercanos, los autores dividen esta tarea en dos secciones: (1) La generación de candidatos y la búsqueda exacta. (2) El paso de generación de candidatos consiste en recorrer T árboles construidos en la fase de indexación. La hoja a la que pertenece un punto de consulta q se recupera proyectando primero q en el nodo raíz del árbol en el mismo vector aleatorio que los datos, y luego, asignándolo a la rama izquierda o derecha dependiendo del valor de la proyección. Si es menor o igual que el punto de corte s (mediana de los puntos de datos que pertenecen a ese nodo en el espacio proyectado) guardado en ese nodo, es decir:

$$q^T r_i \leq s, \tag{2.2}$$

el punto de consulta se enruta al nodo secundario izquierdo o, de lo contrario, al nodo secundario derecho. Este proceso se repite luego de forma recursiva hasta que se encuentra una hoja.
Luego de la construcción de las proyecciones, se ejecuta un sistema de votos que consiste en elegir en el conjunto de candidatos sólo los puntos de datos que residen en la misma hoja que el punto de consulta en al menos v árboles:

$$S = x \in X : F(x; q) \geq v. \tag{2.3}$$

El umbral de voto v es un parámetro ajustable. Un valor de umbral más bajo produce una mayor precisión a expensas de un mayor tiempo de consulta.

.4. Algoritmo Bridge. *Better result with influence diversification to Group Elements* (Bridge) es un algoritmo de detección de *Near-duplicate* [**SBPS$^+$14**]. Bridge mejora el algoritmo BRID propuesto por Santos et al [**SOF$^+$13**] permitiendo la generación de una hiper-esfera alrededor de los elementos menos influenciados en la búsqueda de vecinos cercanos. La influencia de un elemento está definida como la inversa de la distancia hacia el resto de los elementos del espacio de búsqueda:

$$I(s_i, s_j) = \frac{1}{D(s_i, s_j)}, \tag{2.4}$$

Donde s_i y s_j son elementos distintos al nodo inicial de búsqueda y D es una métrica de distancia cualquiera.
La hiper-esfera generada por el algoritmo Bridge esta definida por la siguiente ecuación:

$$\xi = \frac{1}{i}\sum_{u=1}^{i} d(s_u, s_q), \tag{2.5}$$

Donde dado un dominio $\mathbb{S}$ y un dataset $S \in \mathbb{S}$, un objeto de búsqueda $s_q \in \mathbb{S}$, con una distancia $\xi_{max} \in \mathbb{R}$ obtenida a partir de una distancia de evaluación d, los elementos más similares a s_q en el sub-conjunto $S' \subset S | \forall s_j \in S', d(s_j, s_q) \leq \xi_{max}$.
Esto permite generar, detectar y agrupar *near-duplicates* como lo ejemplifica la figura 2.8 y la figura 2.9 permitiendo devolver elementos que otorgan más información a la búsqueda.

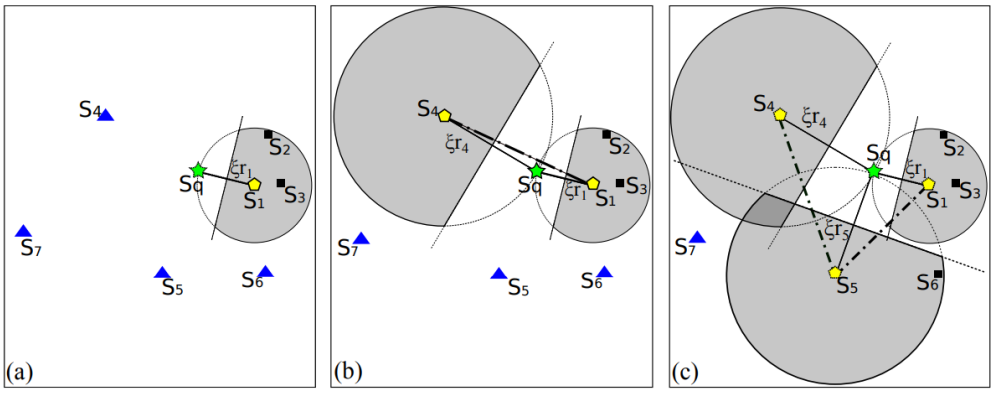


FIGURA 2.8. Tres pasos de la búsqueda de los 3 vecinos diversos más cercanos en un espacio euclidiano bidimensional, utilizando Bridge con k vecinos cercanos. Los pentágonos son elementos de la respuesta establecida en ese paso. Los cuadrados son elementos del conjunto de influencia fuerte. Los triángulos son elementos candidatos que aún no están influenciados por ninguna respuesta.

(Fuente: Paper [**SOF+13**])

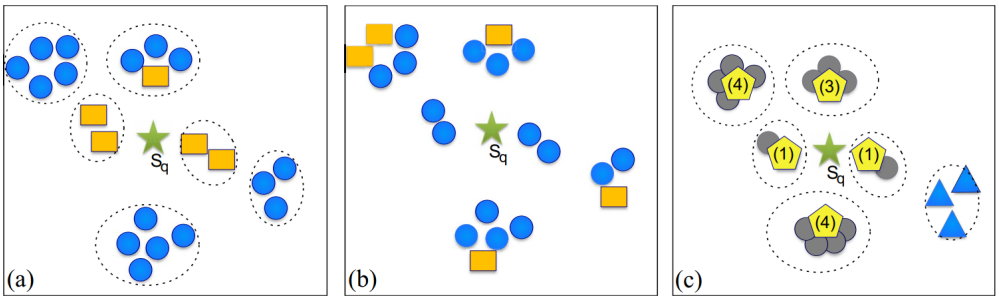


FIGURA 2.9. Selección de elementos para consultas de similitud en un espacio bidimensional euclidiano. Los cuadrados son los elementos seleccionados. (a) El espacio de solución para el k-NNq tradicional centrado en el elemento de consulta (s_q), incluidos los casi duplicados. (b) El conjunto de resultados obtenido mediante un enfoque de diversidad de optimización. (c) La diversificación de resultados recuperado por la agrupación propuesta en CBMIR: Elementos agrupados cerca de duplicados (círculo) y sus elementos representativos (pentágonos). Los triángulo significan elementos no devueltos/utilizados para responder a la consulta.

(Fuente: Paper [**SBPS$^+$14**])

Capítulo 3

Propuesta

En el siguiente capítulo se describe el modelo propuesto para resolver el problema de una manera distinta al estado del arte presentado en la sección anterior.

. Descripción General

Como objetivo principal, se propone desarrollar un CBMIRS basado en el aprendizaje supervisado que pueda trabajar con información de segmentación y clasificación al mismo tiempo. Este nuevo CBMIRS tiene la finalidad de introducir una nueva forma de tratar con el problema de *image retrieval* en el área de la salud. Para lograr este propósito, se utilizará la idea de *transfer learning*, en específico, el método de generación de *embeddings* del área de procesamiento de lenguaje natural. Se utilizarán los *embeddings* resultantes de una primera red como datos de entrada de una segunda red, con la finalidad de transferir la información de la primera red hacia la segunda simulando la técnica de *look-up table*. Los *embeddings* resultantes serán ricos en información y capaz de distanciarse adecuadamente para realizar una búsqueda de imágenes médicas. La primera estructura utilizada se llama *Context Encoder Network* o Ce-Net [**GCF$^+$19**] la cual permite procesar imágenes médicas 2D y obtener una máscara de segmentación de la imagen. Como se mencionó anteriormente, CeNet es innovadora ya que implementa los módulos llamados *Dense Atrous Convolution* (DAC) y *Residual Multi-Kernel Pooling* (RMP) permitiendo capturar información de alto nivel y preservar dicha información a través del entrenamiento de la red. Esta red es entrenada al mostrarle a la red un conjunto de imágenes médicas $x \in \mathbb{X}$ y una máscara binaria generar por un experto $y \in \mathbb{Y}$ con $\mathbb{Y} \in [0, 1]$.

La segunda arquitectura se llama Xception [**Cho17**], la cual es una arquitectura especializada en clasificación de imágenes naturales. Xception tiene la cualidad de capturar dependencias en los datos de entrada, permitiendo identificar relaciones y patrones en las imágenes y transmitir dichos patrones a distintas etapas de la red, permitiendo generar una mejor representación en las capas ocultas.
La presente investigación tiene la finalidad de estudiar la factibilidad de combinar CeNet y Xception ya entrenadas en sus correspondientes tareas para solucionar el problema de búsqueda de imágenes médicas. La representación latente otorgada por CeNet en la capa final del *encoder* donde se sitúa el extractor de contexto será utilizada como vectores pre-entrenados para entrenar y ajustar Xception acorde a un diagnóstico.
La idea de esta propuesta radica en que la representación latente otorgada por algún *embedding* extraído de un modelo segmentador de enfermedades, reduce la variabilidad y dificultad de un clasificador en detectar dicha enfermedad. Por lo cual, Xception trabaja con vectores que conllevan información de una tarea previa.
Para realizar el almacenamiento y búsqueda de las imágenes en el CBMIRS se combina la representación latente de la penúltima capa de Xception con el vector generado por CeNet en el extractor de contexto para así obtener un *embedding* final más rico en información que se usara en el sistema final como lo demuestra la figura 3.1.
Finalmente, las dimensiones de imágenes que se procesan en la propuesta son: Como entrada al extractor de características (CeNet) se ingestan imágenes con dimensiones (512,512,X), siendo X, 1 o 3 dependiendo de si las imágenes viene con o sin canales. Luego, al transitar por Xcecption, se inician como un mapa de activación de dimensiones (384,344,1), por ende, la salida dimensional del RMP que alimenta a Xception siempre es de (384,344,1). Finalmente, el vector de salida de Xception termina con una dimensión de (262144,1) y este vector puede concatenarce con una versión *flatten* del vector de entrada de dimensiones (384,344,1) para su uso en el sistema de recuperación de imágenes.

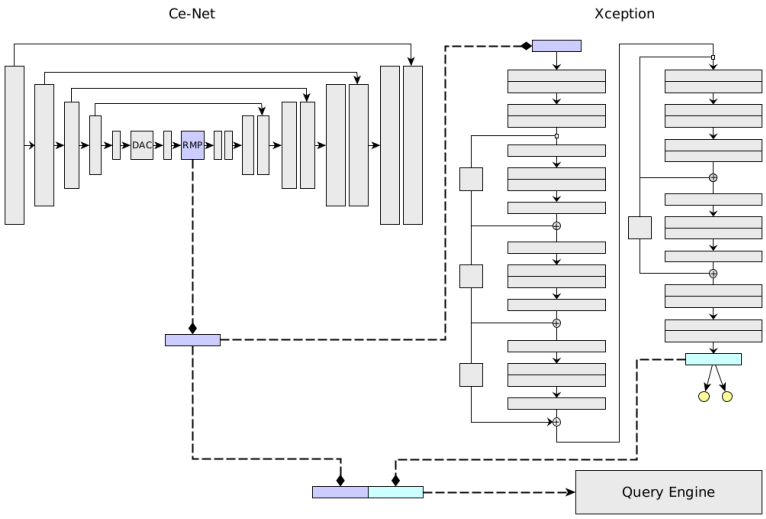


FIGURA 3.1. Ilustración de la combinación de representaciones latentes para construir el *embedding* de la propuesta. Desde CeNet se extrae la representación latente del bloque extractor de contexto, el cual se utiliza para entrenar Xception y posteriormente se concatena con alguna representación latente de Xception para generar el *embedding* final.

. CeNet

Para realizar la segmentación de enfermedad u órgano, entrenaremos nuestro modelo utilizando *Dice Coefficient loss*, el cual nos permitirá detectar las pequeñas regiones de interés que ocupa alguna enfermedad. Esta función de pérdida se expresa como una medida de superposición ampliamente utilizada en el área de la segmentación mientras se tenga disponible un *ground truth*, la expresión matemática se expresará como:

$$L_{dice} = 1 - \sum_{k}^{K} \frac{2w_k \sum_i^N p(k,i) g(k,i)}{\sum_i^N p^2(k,i) + \sum_i^N g^2(k,i)} \tag{3.1}$$

Donde N es el número de pixel, $p(k,i) \in [0,1]$ y $g(k,i) \in 0,1$ denotan la probabilidad predicha y la etiqueta *ground truth* por clase k. K es el número de clases y $\sum_k w_k$ son los pesos de cada clase. En esta experimentación, se definió $w_k = \frac{1}{K}$ empíricamente. Finalmente, se incorpora un factor regularizador a la función *Dice* denominado L_{reg} para evitar el *overfitting*. La función de perdida final se expresa como:

$$L_{loss} = L_{dice} + L_{reg} \tag{3.2}$$

.1. Entrenamiento. El entrenamiento de la red será realizado en base a los pesos pre-entrenados de ResNet con Imagenet. Durante el entrenamiento, se utilizó el algoritmo *Adam* como optimizador, el tamaño de *batch* igual a 1 (por el gran tamaño natural de las imágenes médicas), un *learning rate* inicial de $2e^{-4}$ y se utilizó un total de 20 epochs de entrenamiento.

. Xception

Este modelo se fue entrenado con dos modalidades distintas, la primera modalidad se utilizó como *baseline* y fue un entrenamiento directo sobre las imágenes médicas. La segunda modalidad será parte de la propuesta de este estudio, donde se utilizarán los *embeddings* resultantes de CeNet como datos de entrada para Xception.

.1. Entrenamiento utilizando imágenes. El optimizador seleccionado fue SGD con un *learning rate* inicial 0,001 y momentum de 0,9 sobre un *batch* de tamaño 1 por la naturaleza de las imágenes. Se utilizó *learning decay* con un valor de 0,94 cada 2 *epochs* durante un total de 40 *epochs* de entrenamiento con un sistema de *early stop* si la función de pérdida no presenta cambios significativos en los últimos 5 *epochs*. Experimentalmente, el entrenamiento tomó alrededor de 15 a 20 *epochs* en promedio sin signos significativos de mejora si se entrenaba por más tiempo.
Como medidas de regularización se utilizo *Weight decay* con un valor de $1e^{-5}$ y *dropout* con un ratio de 0,5.

.2. Entrenamiento utilizando embeddings. Para este tipo de entrenamiento se realizaron un par de modificaciones gracias a los *embeddings* resultantes de CeNet. Primero, se logró aumentar el tamaño del *batch* de entrenamiento a 16 gracias a la cualidad de los *embeddings* de contener información con una menor dimensionalidad, además de eso se aumentó el número de *epochs* a 100.

. Técnica de ensamble

Para construir la propuesta presentada en la figura 3.1 se necesita por un lado, entrenar a CeNet en la segmentación de algún tipo de enfermedad u órgano, por ejemplo, para segmentar las zonas de vidrio esmerilado de una imagen de pulmón. Por otro lado, se debe entrenar Xception en clasificación binaria en la presencia o ausencia de una cierta enfermedad, por ejemplo, si una imagen de pulmón presenta o no Covid-19. Para realizar esta tarea, se utilizará el entrenamiento explicado anteriormente en cada una de las redes. Se puede observar un resumen en la tabla 3.1.

Hyperparameter \ Model	CeNet	Xception Image	Xception Embedding
Loss	Dice Loss	Cross Entropy	Cross Entropy
Pretrained	Backbone: ResNet & Imagenet	-	-
Optimizer	Adam	SGD	SGD
Learning Rate	$2e^{-4}$	0.001	0.001
Learning Decay	-	0.94 - Each 2 epochs	0.94 - Each 2 epochs
Batch Size	1	1	16
Epochs	20	40	100
Early Stop	5 Last Epochs	5 Last Epochs	5 Last Epochs

Tabla 3.1. Tabla resumen de los híper parámetros utilizados en el entrenamiento de los distintos modelos usados para construir el CBMIRS propuesto.

Para generar este ensamblado que producirán los *embeddings* que contengan la información de segmentación y clasificación de dos redes, se necesitan los siguientes pasos:

- Entrenar a CeNet como modelo de segmentación con los parámetros especificados en la tabla 3.1.
- Utilizar el modelo CeNet previamente entrenado para generar un conjunto de *embeddings* a través de la extracción de alguna capa de activación de CeNet.
- Entrenar a Xception utilizando los parámetros especificados en la tabla 3.1 con los *embeddings* obtenidos por CeNet, los cuales contienen la información de segmentación.
- Una vez entrenados los modelos, el CBMIRS deberá utilizar CeNet y Xception como generadores de *embeddings*, donde la imagen médica es procesada primero por CeNet, y posteriormente, el *embedding* resultante de dicha imagen pasa a través de Xception donde finalmente, el *embedding* generado por Xception se utiliza en el sistema de búsqueda. Como opción adicional, el *embeddding* generado por Xception puede ser concatenado al *embedding* generado por CeNet con el fin de incluir información adicional.

. Auxiliary Manifold Embedding UNet

En esta investigación se identificó un factor importante: la posibilidad de incorporar diferentes tipos de información al proceso del entrenamiento, como por ejemplo, imágenes tomadas de exámenes de CT y datos tabulares que contienen metadatos de algunos pacientes previamente anonimizados. La razón para determinar la adición de esta información en el proceso de construcción del modelo como un aspecto importante del trabajo surge en la idea de que los *embeddings* pueden beneficiarse de una diversidad

de información sobre el dominio, mientras que el contenido de la imagen (tomado del examen de una persona) y la tarea para la que se entrenó el modelo, es una línea que nos guía en la construcción de los *embedding* a utilizar, también, pueden haber otros aspectos no tan explícitos (o incluso completamente ausentes) en esta imágen, como la edad, el género o antecedentes médicos del paciente. Sin embargo, una dificultad importante en la configuración experimental de este esquema de aprendizaje mixto fue la escasa cantidad de datos etiquetados para diferentes tareas de aprendizaje con la presencia conjunta de metadatos de los pacientes. Mientras que algunos conjuntos de datos médicos existen con imágenes y un *ground truth* correspondiente, no muchos de ellos tienen metadatos asociados a la imagen de los pacientes de los cuales se tomó el examen y en los raros casos en los que se encontró un conjunto de datos con tales características, las imágenes en sí carecían de las especificaciones técnicas necesarias para el trabajo. En respuesta a lo expuesto anteriormente, se consideró un esquema de aprendizaje semi-supervisado como una solución que puede ser lo suficientemente flexible como para permitir el entrenamiento con diversidad de tipos de datos. Inspirada en el trabajo de Baur et al.[**BAN17**], se diseñó una arquitectura estilo UNet para poder entrenar con tres tareas: segmentación supervisada, clasificación supervisada y *Manifold Embedding* no-supervisado.

.1. Arquitectura de red neuronal. Para esta sección de trabajo, se utilizo un modelo *end-to-end* para aprender tres tareas diferentes usando la arquitectura ilustrada en la figura 3.2. Esta arquitectura hereda el *backbone* de la clásica Unet [**RFB15**] con modificaciones en la capa final del decoder. Esta modificación consta de dos bloques conectados a la última capa del decoder; el primer bloque consta de cinco *dense layers* que se utilizan para la tarea de clasificación, mientras que el segundo bloque tiene solo una *convolutional layer* dedicada a la tarea de segmentación. Asimismo, cabe señalar que la función de activación de algunas capas en los bloques finales que generan la salida de la red se cambió a ELU, ya que esto mostró experimentalmente un entrenamiento más estable.

.2. Tarea de segmentación. Para esta tarea el modelo fue entrenado durante 5 *epochs* usando la salida de la capa Out_{seg}, utilizando la función de perdida *Dice* y Adam como optimizados con un learning rate inicial de $1e^{-4}$.

Las segmentaciones logradas presentaron algunos artefactos alrededor del órgano segmentado principal, la razón de esto fue que el *ground truth* no fue realizado por el mismo radiólogo y algunos de ellos etiquetaron la sección exterior de la imagen con el propio órgano. Dado que la relevancia del modelo es su capacidad para capturar una aproximación del *ground truth* y asi lograr que la red aprenda las principales características del dominio, se consideró suficiente para que los experimentos prosiguieran.

.3. Tarea de clasificación. Posterior al entrenamiento en segmentación, el modelo fue re-entrenado por 5 *epochs* con el mismo optimizador y *learning rate* que fue

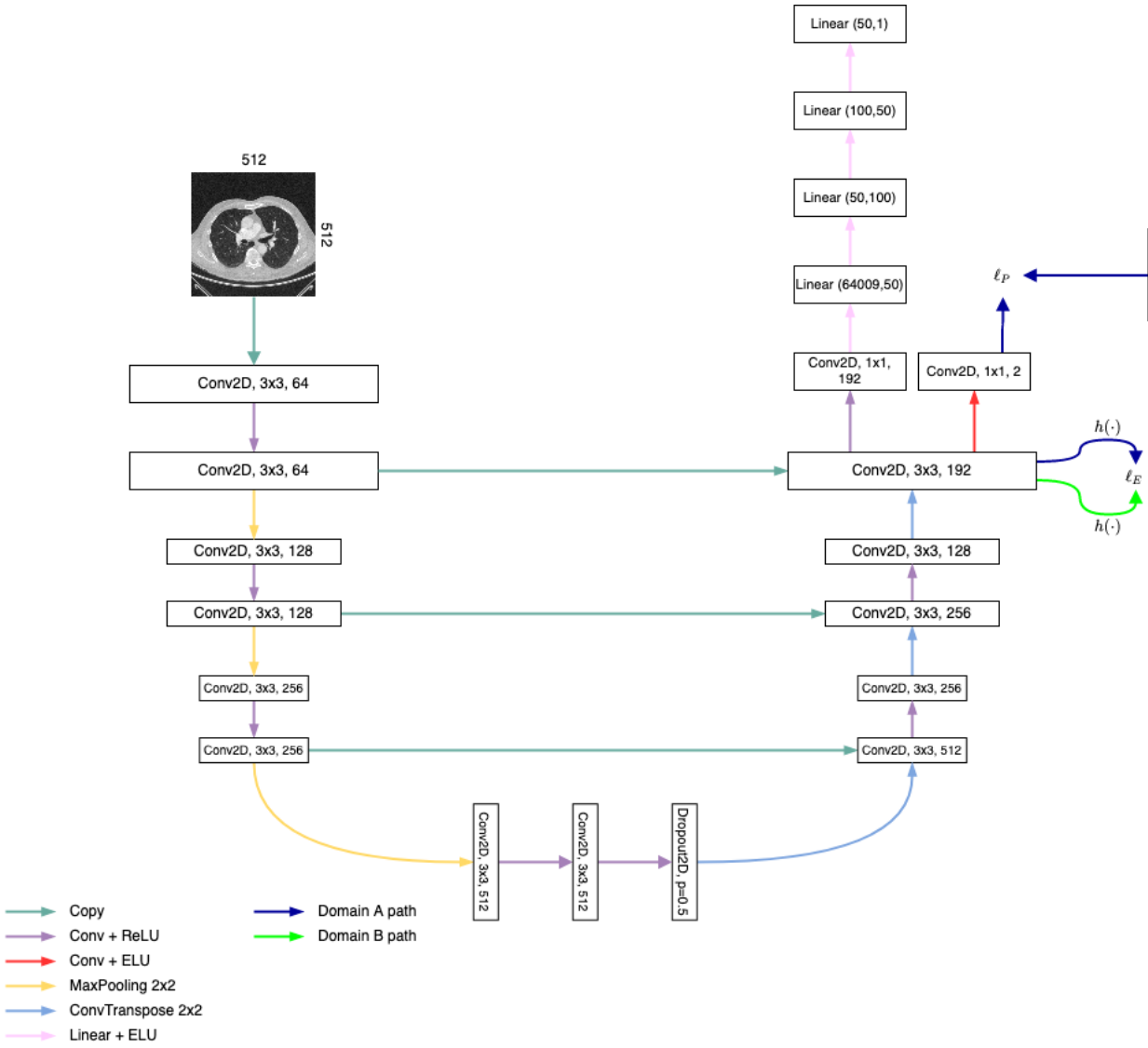


FIGURA 3.2. Arquitectura propuesta basado en el trabajo de Baur et al.[**BAN17**].

(Fuente: Original)

mencionado anteriormente, pero en esta ecuación se utilizó una perdida de *binary cross entropy* sobre la salida del modelo $Out_{classif}$. Esta actividad de entrenar el modelo en segmentación y posteriormente en clasificación crea el modelo denominado ***Unet Mixed Learning (Unet-ML)***

.4. Entrenamiento semi-supervisado con *manifold embedding*. Como se mencionó anteriormente, esta arquitectura fue entrenada con un esquema semi-supervisado,

utilizando una aproximación similar a [**BAN17**] donde se emplea una matriz de adyacencia que fue creada para representar el grado de similitud entre cada par de imágenes en el dataset. Idealmente esta matriz debe ser construida utilizando la metadata de los pacientes, pero al momento de la experimentación, dicha metadata no se encontraba disponible por lo cual se construyó la matriz considerando dos imágenes como "Una cerca de la otra", cuando el diagnóstico de las imágenes era el mismo.
Formalmente, sea A una matriz binaria de $N_U \times N_U$, donde N_U es la cantidad de imágenes del dataset usadas para el entrenamiento no supervisado:

$$a_{ij} = \begin{cases} 1, & \text{if } d_i = d_j \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Donde $a_{i,j}$ representa la matriz de adyacencia entre la *i-th* y *j-th* imagen, en nuestro caso, dado por el diagnóstico para dos imágenes (d_i y d_j respectivamente).
La función de pérdida utilizada para este experimento es la misma que la utilizada por Baur et al.[**BAN17**]. Esta función compara los *embeddings* creados por el modelo con los valores adyacentes entre dos imágenes codificadas en A y así, guía el modelo con la finalidad de crear representaciones latentes $h(\cdot)$ de las imágenes que siguen el patrón de adyacencia en A. La pérdida total del *embedding* calculada en la $l-th$ capa usada en el esquema no-supervisado se describe como la siguiente ecuación:

$$\mathcal{L}_{E_l}(X, A) = \begin{cases} d(h^l(x_i), h^l(x_j)) & \text{if } a_{ij} = 1, \\ max(0, m - d(h^l(x_i), h^l(x_j))) & \text{if } a_{ij} = 0 \end{cases} \tag{3.4}$$

donde $d(\cdot, \cdot) \in \mathbb{R}^1$ es la métrica de distancia calculada entre los *embeddings* generados por la *l-th* capa ($h^l(\cdot)$) para la *i-th* y *j-th* imagen. Como los autores del artículo original en el que se inspiró este modelo utilizaron la distancia angular de coseno.
Para nuestros experimentos la capa utilizada para la extracción de los *embeddings* fue la última capa del decodificador (Conv2D, 3x3, 192), ya que tiene suficiente dimensionalidad para capturar una rica representación de la información destilada en las capas anteriores.

Debido a los altos costos computacionales en los cálculos del *forwards pass*, solo un tamaño de *batch* de 2 nos permitió entrenar de manera efectiva el modelo. Esto significa que, a diferencia de los autores originales, nuestra aproximación no usa una agregación de los valores de la ecuación 3.4 para muchos elementos del *batch*.

Este modelo entrenado con la técnica de *Manifolds Embedding Loss* fue nombrado como ***Unet-ME***.

. Random Projection Tree

Para esta propuesta de tesis, se trabajó con ***Multiple Random Projection Tree*** (MRTP) [**HPT+16**] en conjunto con *Automatic hyperparameter tuning* [**JHR19**] con la finalidad de generar índices de manera eficiente, utilizando la cualidad multi-dimensionalidad de los *embeddings* obtenidos por las redes neuronales y la sencillez de la proyección aleatoria del algoritmo MRTP.

. Algoritmo Bridge para eliminación de Near-Duplicate

Se utilizó el algoritmo *Bridge* explicado anteriormente, como parte de los experimentos, para lograr comprobar si la reducción de imágenes *near-duplicate*, en conjunto al algoritmo MRTP, tienen efectos considerables o no en la búsqueda de imágenes médicas.

Capítulo 4

Experimentos

A continuación se presenta la evaluación experimental de la propuesta sobre distintos *datasets* pertenecientes a nódulos pulmonares y Covid-19.

. Medidas de evaluación

Para evaluar el desempeño de la propuesta, se utilizaron dos medidas de evaluación:

- ***Precision***: Es la relación entre el número de imágenes relevantes que se ha recuperado en la búsqueda y el número total de imágenes irrelevantes y relevantes recuperadas. En otras palabras, suponiendo que A es el número de imágenes relevantes recuperadas y B es el número total de imágenes irrelevantes recuperadas, el cálculo de precisión se realiza mediante la siguiente función:

$$Precisión = A/(A + B) \tag{4.1}$$

- ***Recall***: Esta métrica evalúa cuántas de las imágenes relevantes se han recuperado en la búsqueda actual de un total conocido dentro de la base de datos, que es el número total de imágenes relevantes que existen. Suponiendo que A fuese nuevamente el número total de imágenes relevantes que ha recuperado la búsqueda de la base de datos y C representa el número total de imágenes relevantes en la base de datos, el cálculo de *recall* se realiza mediante la siguiente función:

$$Recall = A/C \tag{4.2}$$

. Datasets

.1. Covid-19. Para producir un CBMIRS para el nuevo virus Sars-Cov-2 y mantener los experimentos lo más próximo a un ambiente clínico real, se utilizaron 3 *dataset* para entrenar y probar tanto CeNet, Xception, Unet-ML, Unet-ME y el nuevo sistema propuesto. El objetivo de usar 3 *dataset* distintos es entrenar los modelos de segmentación y clasificación sobre dos *dataset* y ocupar un tercer *dataset* como conjunto de pruebas para el sistema de búsquedas con el objetivo de simular imágenes obtenidas en ambientes clínicos reales caracterizadas por ser imágenes de distintas resoluciones y tamaños.

COVID-19 CT segmentation dataset: Disponible públicamente [1] este dataset contiene 100 CT axiales de 40 pacientes con Covid-19. Cada corte fue evaluado por un radiólogo con 3 tipos de etiquetas: vidrio esmerilado, consolidación y derrame pleural para el nivel de daño de Covid-19. Este *dataset* se utilizó para entrenar CeNet y Unet-ML en su módulo de segmentación. A pesar de que las máscaras de este *dataset* tienen distintas tonalidades de píxeles, medidos entre 0 y 1 para diferenciar los distintos daños provocados por el Covid-19, como CeNet originalmente se construyó para generar únicamente mascaras binarias de 0 y 1, solamente se le enseñó a la red a detectar si existe daño provocado por la enfermedad o no, sin tomar en cuenta las distintas tonalidad entre 0 y 1. Esta misma decisión se aplicó a Unet-ML para que los experimentos fueran consistentes. Un ejemplo del resultado de entrenamiento de los segmentadores se puede ver en la figura 4.1.
SARS-CoV-2 CT-scan dataset: También disponible públicamente [2], este *dataset* cuenta con 1252 cortes axiales de pacientes con Covid-19 y 1229 cortes de pacientes sin Covid-19. Este *dataset* fue utilizado para entrenar Xception, Unet-ML en su modulo de clasificación y Unet-ME.
the Covid-CT dataset: Dataset de Covid-19 de uso libre [3] que contiene 349 cortes axiales de pulmones con Covid-19, obtenido de 216 pacientes y también, un total de 463 cortes axiales de pulmones sin Covid-19, pero sin descartar otras enfermedades. Este *dataset* contiene imágenes adquiridas con diferentes medios, por ejemplo, cortes de TC post-procesados por cámaras de teléfonos celulares y algunas imágenes con muy baja resolución. Por estas razones, el conjunto de datos representa las condiciones reales de adquisición de imágenes para un sistema de este tipo, haciéndolo un *dataset* de prueba ideal para esta investigación. Algunos ejemplos de este *dataset* pueden ser observados en la figura 4.2.

[1] `http://medicalsegmentation.com/covid19/`
[2] `https://www.medrxiv.org/content/10.1101/2020.04.24.20078584v3`
[3] `https://github.com/UCSD-AI4H/COVID-CT`

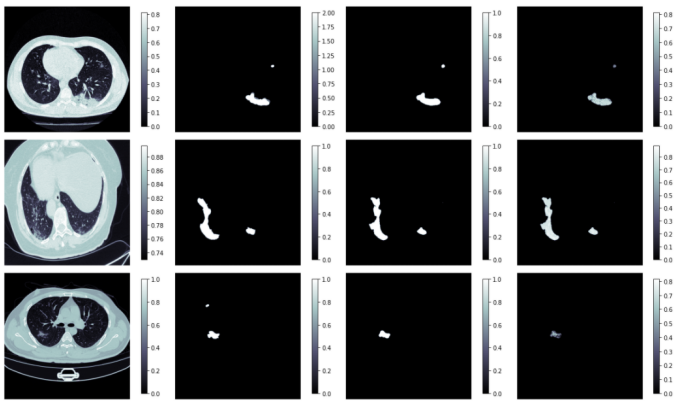


FIGURA 4.1. Resultado del entrenamiento de la red CeNet utilizando el *dataset* ***COVID-19 CT segmentation dataset***. Desde izquierda a derecha se puede ver la imagen original. La mascara binaria dictada por el radiólogo, la predicción de la red y finalmente la superposición entre la capa predicha por la red y la imagen original.

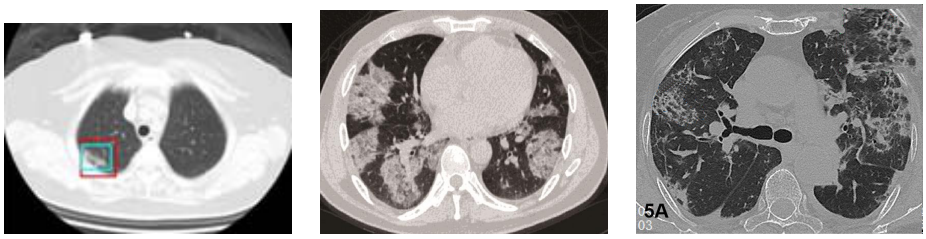


FIGURA 4.2. Ejemplos de imágenes de ***the Covid-CT dataset*** donde se aprecian las distintas resoluciones.

.2. Nódulos Pulmonares. Otra patología estudiada en esta tesis fue la búsqueda de imágenes con nódulos pulmonares. Como conseguir *dataset* públicos de esta enfermedad resulta bastante complejo, se utilizó un único *dataset* que tiene las características de segmentación y clasificación pertinentes, añadiendo *data augmentation* para solventar los problema de cantidad de imágenes en el conjunto de prueba del sistema de búsqueda.

LUng Nodule Analysis 2016: Más conocido como LUNA16 [**STDB**$^{+}$**17**], este *dataset* es una versión curada de LIDC/IDRI *dataset* [**AIMB**$^{+}$**11**] donde se utilizan únicamente cortes con un grosor superior a 2,5mm. También se cuenta con anotaciones compuestas por 4 radiólogos expertos. A su vez, este *dataset* contiene la segmentación pulmonar [4] haciendo este conjunto de datos un perfecto candidato a trabajar. Para procesar las 888 CT y encontrar los correctos cortes que contienen los nódulos

[4]https://www.kaggle.com/kmader/finding-lungs-in-ct-data

	Base de Datos	Conjunto Representativo	Datos Positivos	Datos Negativos
Covid-19	995	249	471	524
Nódulos Pulmonares	372	186	184	178

Tabla 4.1. Distribución de datos para experimento de cálculo de *precision* y *recall* de las enfermedades pulmonares de Covid19 y Nódulos Pulmonares.

pulmonares, se utilizó un pre-procesamiento similar a Zhu et al. [**ZLFX18**]. Este pre-procesamiento detecta que corte tienen la presencia de un nódulo pulmonar candidatos y si está diagnosticado con al menos 3 especialistas, se considera un caso positivo. Este pre-procesamiento nos deja con un total de 928 imágenes de pulmones en corte axial donde existe la presencia de nódulos pulmonares y mas de 22.000 imágenes sin nódulos pulmonares. Para compensar esta diferencia de datos, se utilizó sub-sampling en las imágenes sin presencia de nódulos pulmonares para tener un conjunto de datos de entrenamiento balanceado compuesto de 1856 imágenes donde posterior a eso, se dividió el conjunto en 1670 datos para entrenamiento y validación y 186 datos de prueba.

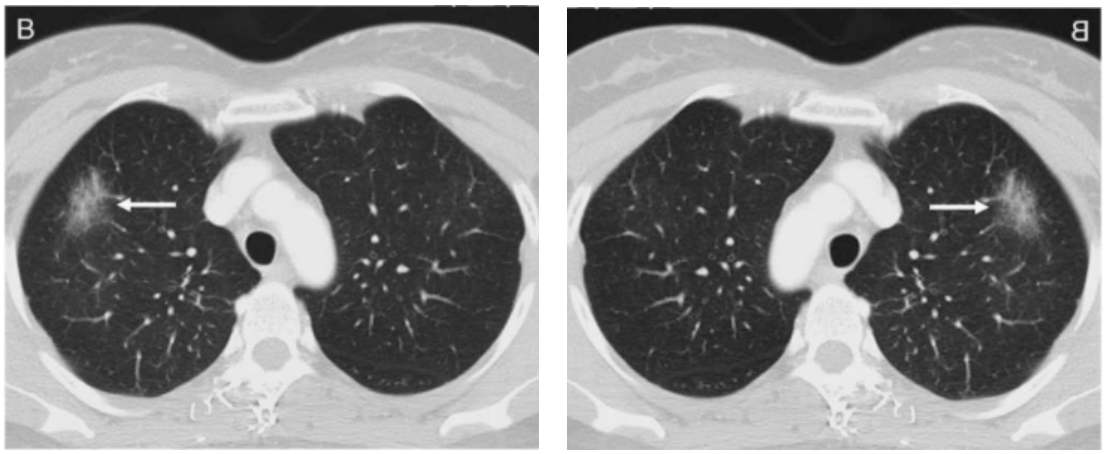
. Composición y distribución de dataset

Para realizar correctamente los experimentos en un CBIRS que simule un ambiente clínico real, se debe tener una clara separación del conjunto de datos utilizado para entrenar los modelos y el conjunto de datos al interior de la base de datos que se utilizará para calcular métricas de experimentación. Para realizar esto, se generó una composición de datos para los distintos *dataset* de Covid-19 y se utilizó *data augmentation* para el caso de nódulos pulmonares. Una tabla resumen se puede observar en la tabla 4.1

.1. SARS-CoV-2. Para trabajar con Covid-19 se extrajo una pequeña porción de los datos de *SARS-CoV-2 CT-scan dataset* en conjunto a la totalidad de *the Covid-CT dataset* con la finalidad de modelar la base de datos. El resto de datos de *SARS-CoV-2 CT-scan dataset* seran utilizados para entrenar los modelos. Como la sección que se extrae pertenece al mismo dominio de los datos de entrenamiento, pero a su vez, la red nunca a visto estas imágenes, se definirá este conjunto de datos como elementos relevantes en la base de datos. Se puede observar un resumen de la división y composición de *dataset* para la construcción de la base de datos y el entrenamiento de los modelos en la figura 4.2.

Dataset\Datos	Entrenamiento	Base de Datos	Tarea
COVID-19 CT segmentation dataset	2316	0	Segmentación
SARS-CoV-2 CT-scan dataset	2232	**249**	Clasificación
The Covid-CT dataset	0	746	Clasificación

TABLA 4.2. División y composición de datos para experimentación y entrenamiento, donde la cantidad marcada en rojo representa el conjunto de datos representativos en la base de datos.



(A) Presencia de nódulo pulmonar original. (B) Presencia de nódulo pulmonar reflejado.

FIGURA 4.3. Ejemplo de la factibilidad de generar nuevos datos de nódulos pulmonares mediante reflexión.

.2. Nódulos Pulmonares. Trabajar con esta enfermedad requirió otra aproximación, como solamente se tiene un *dataset* disponible con máscaras de segmentación e imágenes de clasificación, se utilizó *data augmentation* para construir una base de datos que contenga elementos relevantes, un ejemplo de esta operación se puede observar en la figura 4.3. Se extrajeron 186 datos de LUNA16 y se aplicó reflexión para duplicar esta cantidad. Como las 372 imágenes pertenecen al mismo dominio que los datos de entrenamiento, se consideró prudente utilizar la mitad de las imágenes como datos relevantes verificando que no se repitieran datos reflejados con su versión sin reflejar, un resumen de esto se puede observar en la tabla 4.3.

Dataset\Datos	Entrenamiento	Base de Datos	Tarea
Luna16	1670	**186**	Clasificación y Segmentación
Luna16 Reflejada	0	**186**	-

Tabla 4.3. División y composición de datos para experimentación y entrenamiento. Los datos representativos fueron tomados de manera aleatoria de tanto imágenes normales como reflejadas.

- ***Cálculo de Varianza en Precision y Recall en base de datos evolutiva***: En este experimento se comprobó la estabilidad de los modelos trabajados mediante el cálculo de métricas a medida que se incorporaron imágenes a la base de datos. Tanto los experimentos con Covid-19 y Nódulos Pulmonares se trabajaron inicialmente con una base de datos compuesta por el conjunto relevante C y se añadieron 5 imágenes por iteración hasta volver a completar el *dataset* compuesto por 995 y 372 imágenes respectivamente. Se utilizó un radio de búsqueda $K = 10$ constante y se realizaron 50 consultas por iteración, donde 25 consultas fueron de un paciente con la enfermedad a estudiar y 25 consultas con pacientes sin la enfermedad a estudiar, pero sin descartar otras patologías o pulmones completamente sanos. Por ejemplo, una búsqueda con Covid-19 negativo no descarta la presencia de otras enfermedades como nódulos pulmonares. Posterior a esto, se obtuvieron distintas mediciones de la varianza de *precision* y *recall* para cada iteración y se comprobó si existe dispersión al incorporar nuevos datos a una base de datos pre-existente.
- ***Cálculo de Precision y Recall en base de datos estática***:

  En este experimento se utilizó una base de datos compuesta por la totalidad de las imágenes a estudiar. Se generó una búsqueda con los mismos 50 elementos pero en esta ocasión se modificó el radio de búsqueda k partiendo de $k = 5$ hasta $k = 50$ con saltos sucesivos de cinco en cinco. El objetivo de este experimento fue observar como se comportan las métricas en los distintos modelos a medida que se realizan búsquedas cada vez más amplias en la base de datos.
- ***Cálculo de Precision y Recall en base de datos estática con eliminación de near-duplicates***:

  Para este experimento, se utilizó la misma configuración del experimento de ***Cálculo de Precision y Recall en base de datos estática*** pero en la búsqueda de resultado se incluyó la incorporación del algoritmo ***Bridge*** para solamente retornar los elementos más representativos y descartar los *near-duplicate*. Para lograr esto, se amplió la búsqueda en un factor de 10 y 3 unidades para Covid-19 y nódulos pulmonares respectivamente. El objetivo de este cambio es que al realizar una cierta búsqueda se genere un exceso de resultado, de esta forma, el algoritmo puede detectar y eliminar elementos *near-duplicates* sin llegar al riesgo de reducir la cantidad de elementos retornados. Por ejemplo, si se

Model	Total Average Variance
CeNet	7.449880e-03
Xception	4.817799e-05
Unet-ML	1.611723e-02
Unet-ME	1.082856e-04
Proposal-NoConcat	1.527221e-02
Proposal-Concat	9.468791e-05

Tabla 4.4. Distribución final de varianza al utilizar el total de muestras de resultados.

realiza una búsqueda de Covid-19 utilizando $K = 50$ se buscaran 500 imágenes para luego retornar las 50 más representativas.

. Resultados

A continuación se presentan los resultados obtenidos en la experimentación de los 4 modelos *baseline* y los 2 modelos propuestos.

.1. Covid-19:

.1.1. ***Cálculo de Varianza en Precision y Recall en base de datos evolutiva****:* El resultado del cálculo de varianza para analizar la estabilidad de los distintos *embeddings* en los sistemas se puede observar en la figura 4.4 y Tabla 4.4. Para este estudio, se realizó un total de 50 consultas a la base de datos con un área de búsqueda $k = 10$ donde 25 consultas son imágenes con la enfermedad presente y 25 consultas no poseen dicha enfermedad, se presento la varianza de las 50 consultas en base a la cantidad de elementos nuevos en la base de datos y a su vez, la varianza utilizando la muestra total para cada modelo.

Los resultados obtenidos muestran gran estabilidad en los modelos debido a que no se observan grandes cambios en la varianza. Como los valores de la varianza son cercanos a 0, se comprueba que el sistema es estable a medida que se incorporan nuevos datos que no pertenecen al mismo dominio de entrenamiento.

.1.2. ***Cálculo de Precision y Recall en base de datos estática****:* El resultado de las métricas para los distintos modelos trabajados se puede observar en la figura 4.5, 4.6 y 4.7. Como se comentó anteriormente, se realizó un total de 50 consultas a la base de datos donde 25 son imágenes con la enfermedad presente y 25 no poseen dicha enfermedad, presentándose el promedio en los gráficos de resultado.

Los resultados obtenidos demuestran una clara inclinación del modelo propuesto sin concatenación a priorizar la búsqueda de imágenes de Covid-19 sobre las imágenes sin

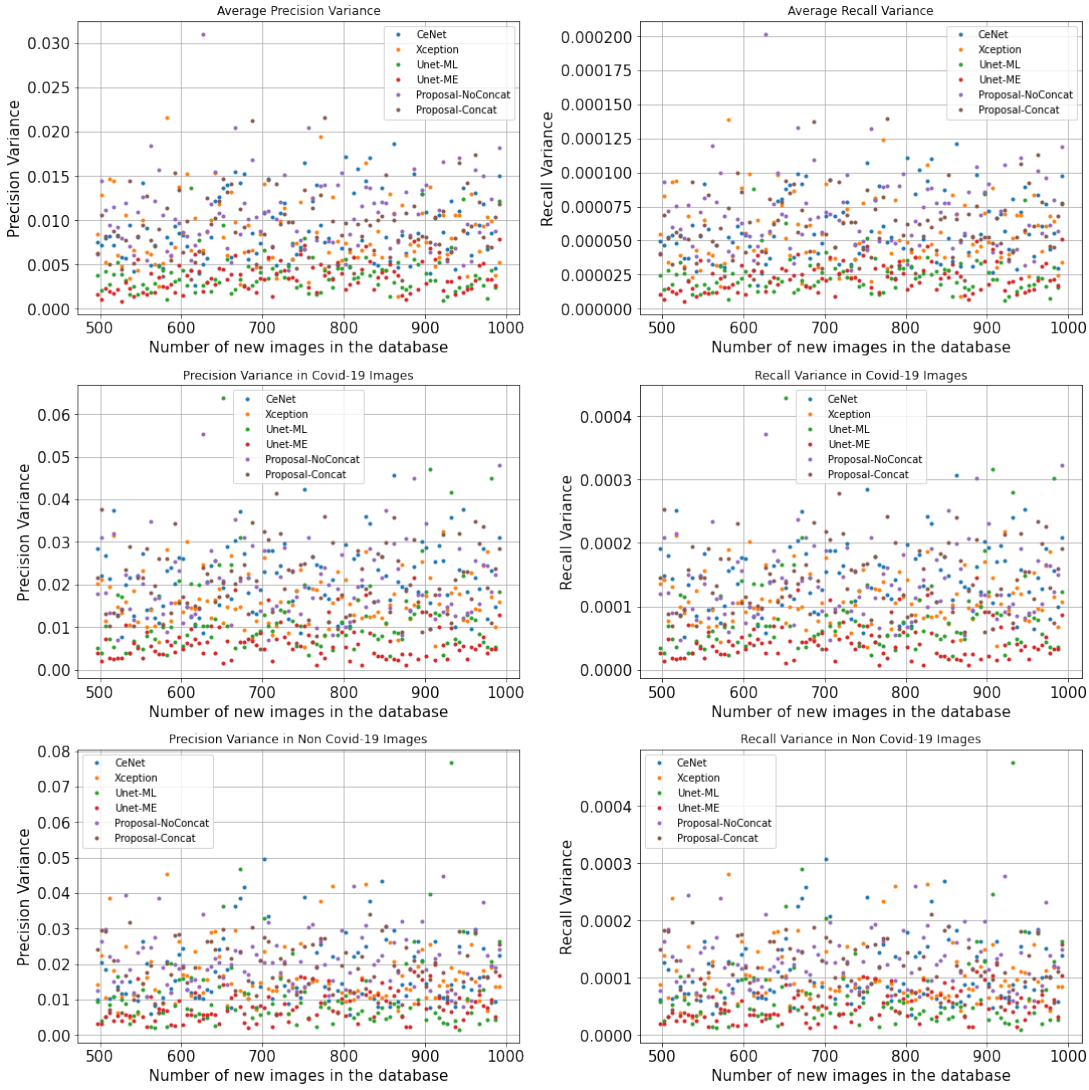


FIGURA 4.4. Distribución de varianza de precision y recall para la enfermedad de covid-19 en base de datos evolutiva.

Covid-19. El modelo propuesto sin concatenación y con concatenación también demuestran un comportamiento estable a medida que se aumenta el radio de búsqueda, donde no se ve una reducción considerable de *precision* en el modelo sin concatenación y se observa un aumento lento pero paulatino de *precision* del modelo con concatenación. En términos de *recall* los modelos propuestos superan al resto de los modelos pero por un margen muy menor, manteniendo un crecimiento similar a los sistemas que utilizan CeNet, Xception y Unet-ML. Respecto a la búsquedas sin Covid-19, podemos observar que el modelo Unet-ME sobrepasa con creces al resto de los modelos tanto en *precision* como en recall llegando a valores del 90 % y 40 % (considerando que el valor máximo de

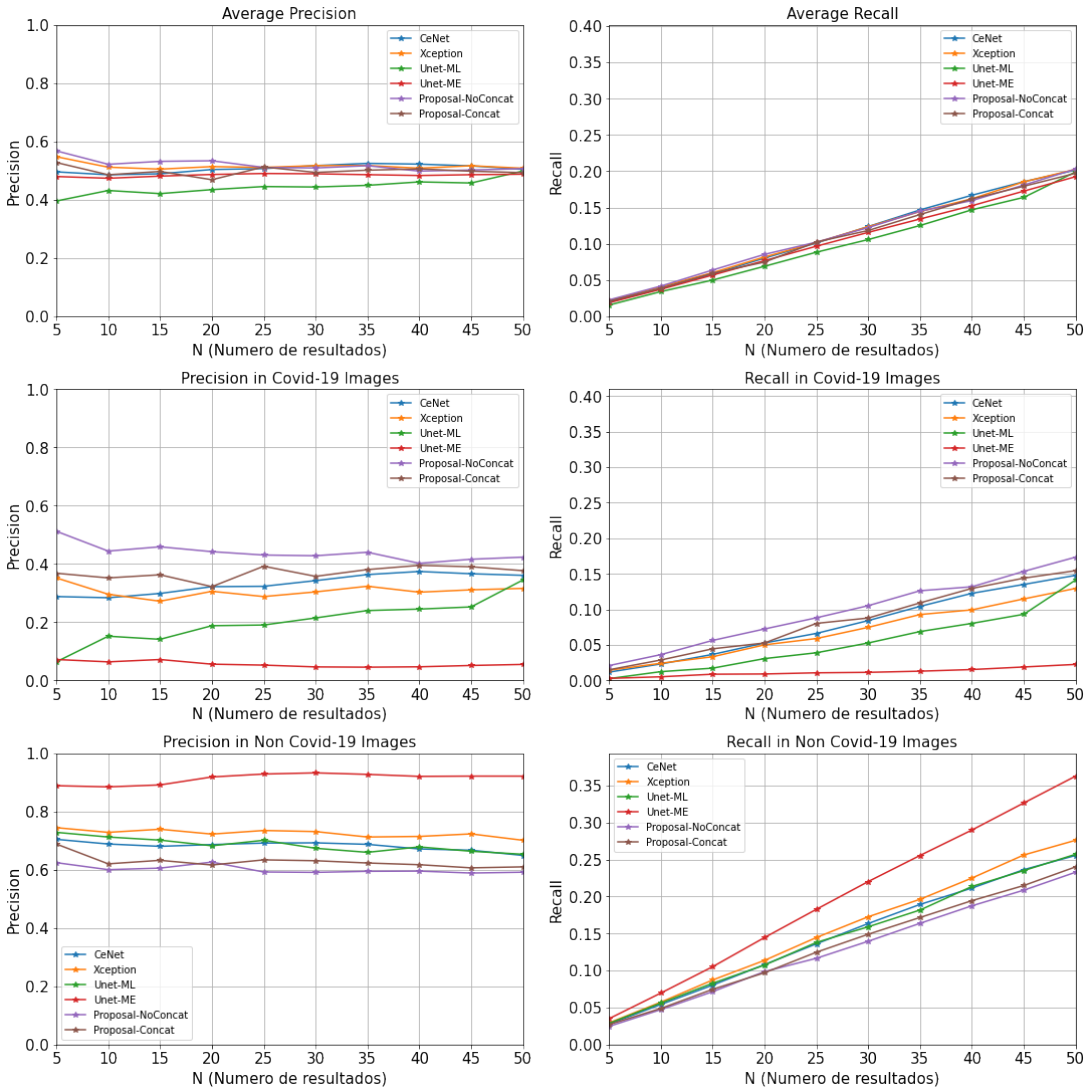


FIGURA 4.5. Resultados en el conjunto de prueba de los distintos modelos sobre el promedio de 25 búsquedas con covid-19, 25 búsquedas sin covid-19 y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

recall para Covid es de 41 % respectivamente). Los gráficos de *precision* vs *recall* nos otorgan más indicios de hasta que rango de búsqueda los modelos son más eficientes. Si observamos el gráfico *Precision-Recall Average* podemos observar que el modelo sin concatenación es el mejor modelo hasta un rango de búsqueda igual a 25, posterior a

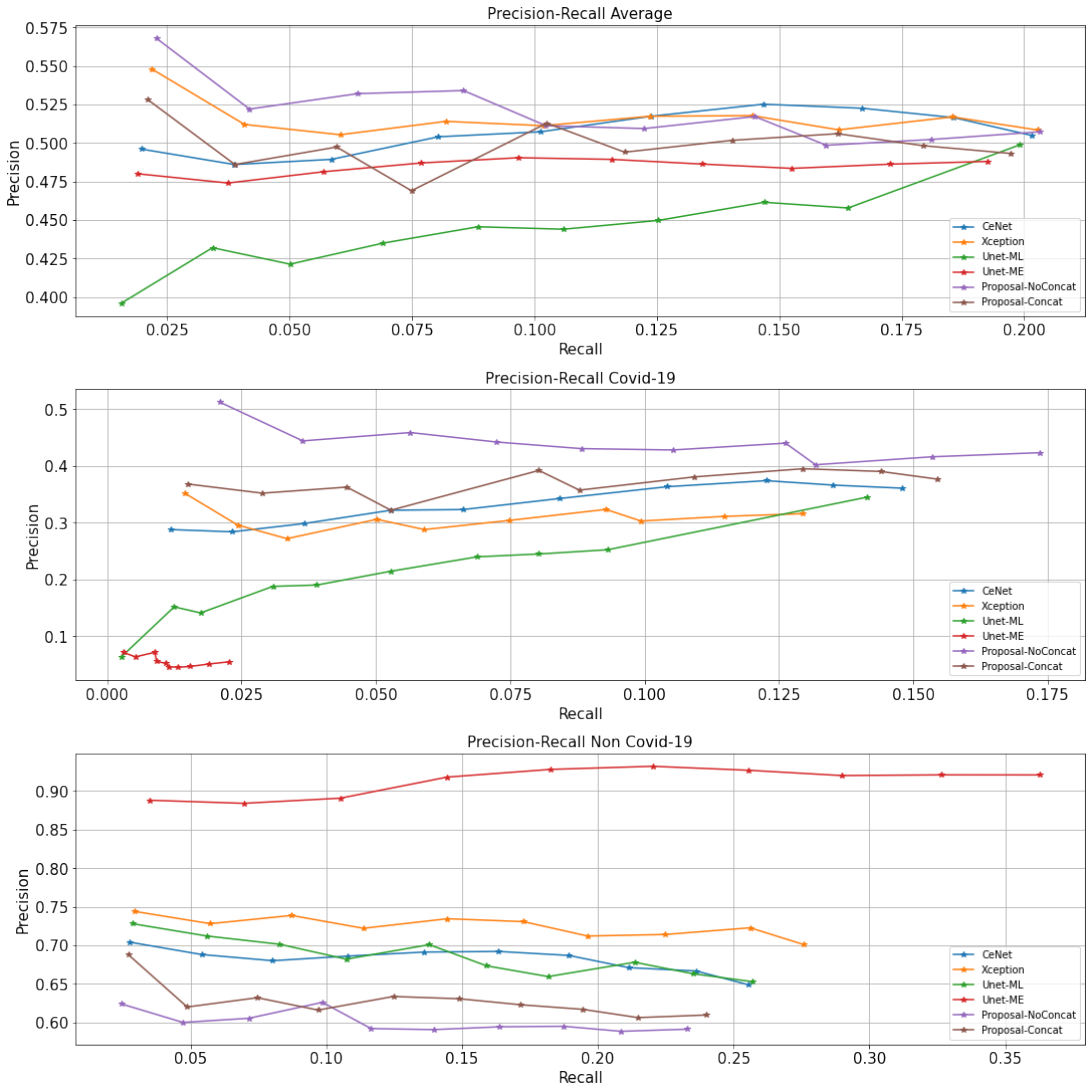


FIGURA 4.6. Gráficos *precision* vs *recall* del experimento 2 para la enfermedad Covid-19 sobre el promedio de 25 búsquedas con Covid-19, 25 búsquedas sin Covid-19 y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

esto otras redes toman el liderazgo. Los gráficos de F1-Score tienen un comportamiento esperado ya que a medida que se va aumentando el radio de búsqueda, el valor F1

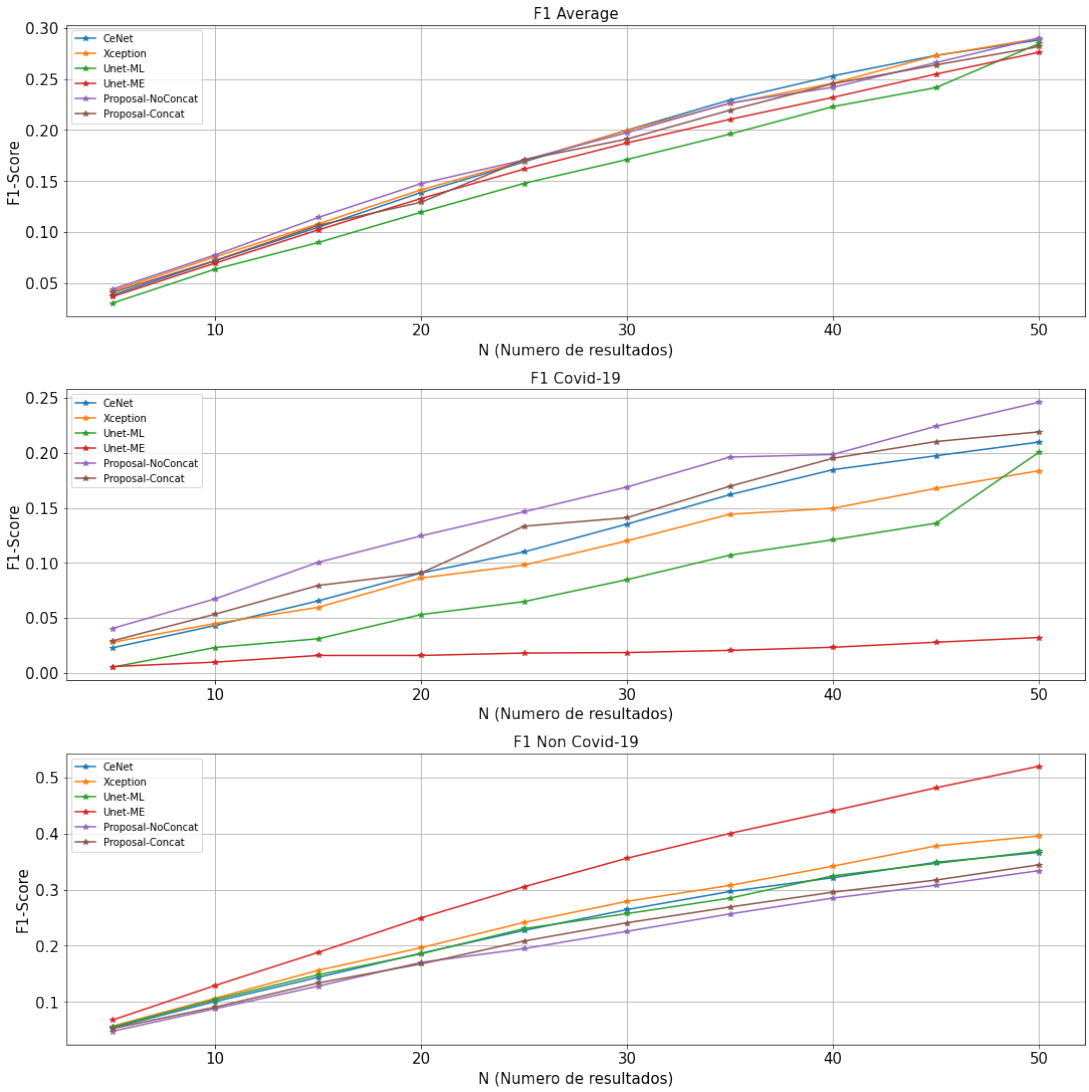


FIGURA 4.7. Gráficos *F1-Score* del experimento 2 para la enfermedad Covid-19 sobre el promedio de 25 búsquedas con Covid-19, 25 búsquedas sin Covid-19 y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

incrementa naturalmente, demostrando nuevamente que el modelo propuesto sin concatenación es el más eficiente para buscar imágenes de Covid-19 y el modelo Unet-ME es el más eficiente para buscar imágenes sin Covid-19.

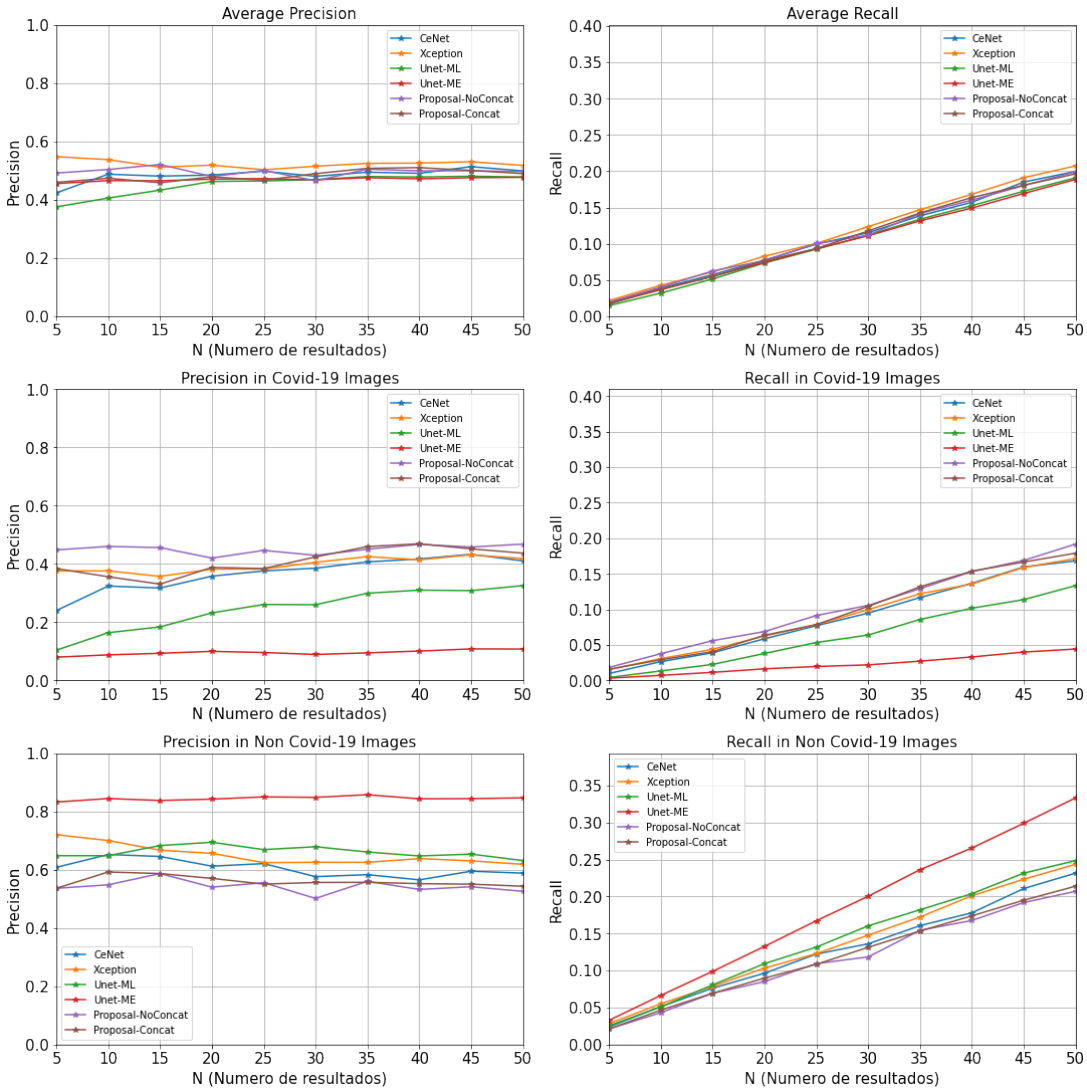


FIGURA 4.8. Resultados en el conjunto de prueba de los distintos modelos con la incorporación de eliminación de vectores *near-duplicates* por el algoritmo *Bridge* sobre el promedio de 25 búsquedas con covid-19, 25 búsquedas sin covid-19 y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

.1.3. ***Cálculo de Precision y Recall en base de datos estática con eliminación de near-duplicates.*** : El resultado para este experimento se puede observar en la figura 4.8, 4.9 y 4.10.

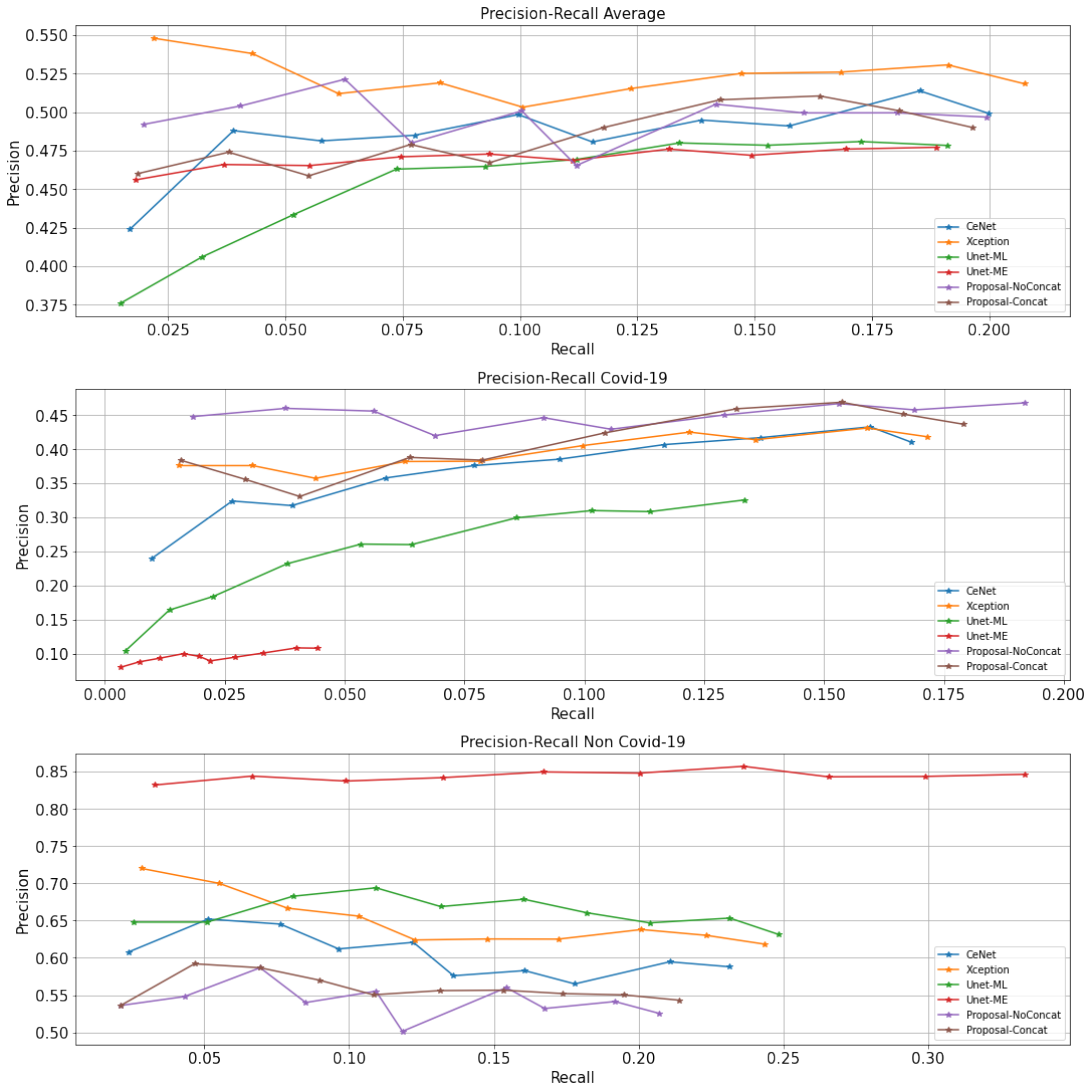


FIGURA 4.9. Gráficos *precision* vs *recall* del experimento 3 para la enfermedad Covid-19 sobre el promedio de 25 búsquedas con Covid-19, 25 búsquedas sin Covid-19 y a su vez, el promedio de las 50 búsquedas con la incorporación de la eliminación de vectores *near-duplicate* por el algoritmo *Bridge*. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

Los resultados obtenidos demuestran que para Covid-19 la eliminación de vectores *near-duplicate* tiene un efecto regularizador en el comportamiento del sistema. Podemos observar que con anterioridad el modelo propuesto con concatenación no lograba

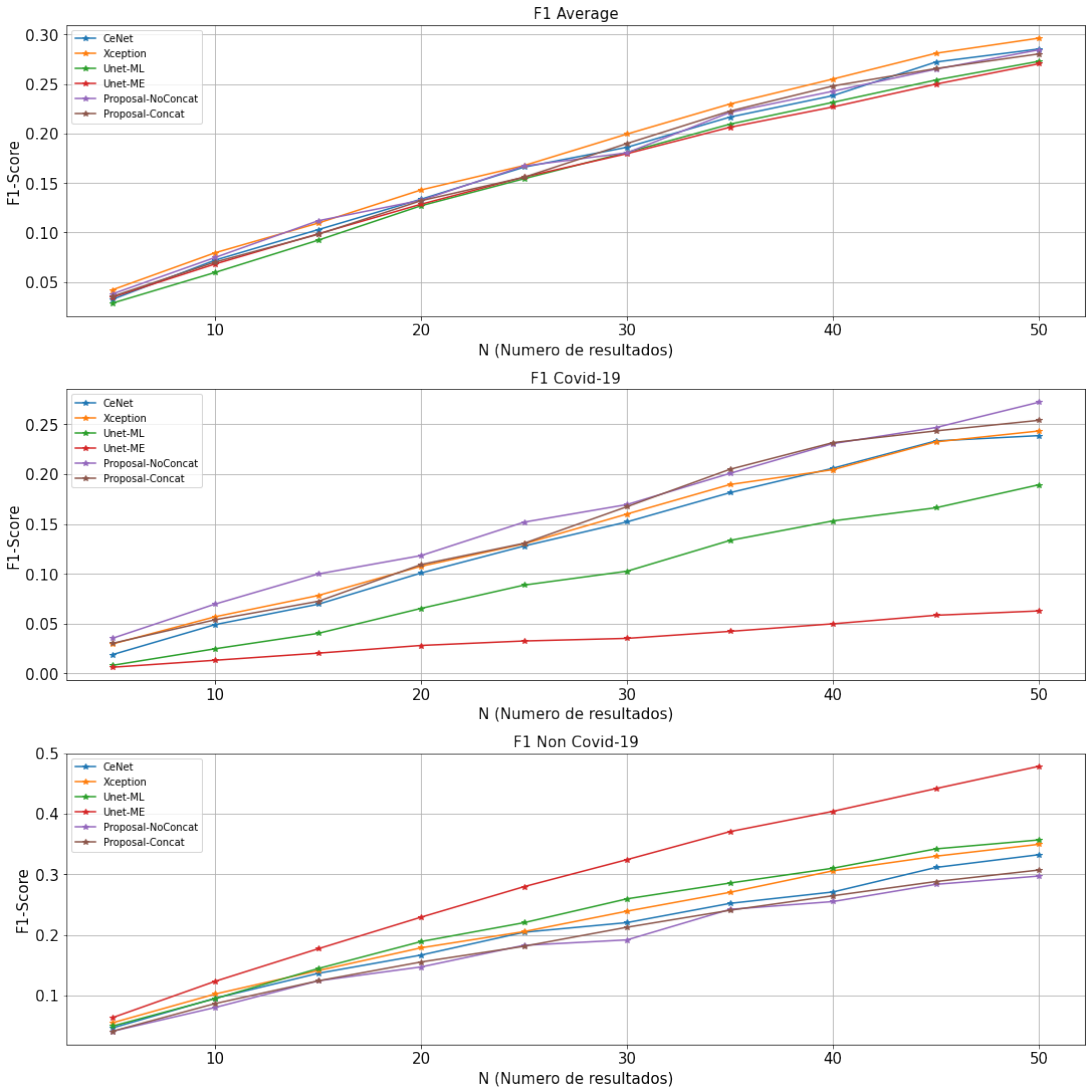


FIGURA 4.10. Gráficos *F1-Score* del experimento 3 para la enfermedad Covid-19 sobre el promedio de 25 búsquedas con Covid-19, 25 búsquedas sin Covid-19 y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

superar al modelo sin concatenación, pero con la incorporación de *Bridge* podemos observar como ambos modelos son competitivos respecto a la búsqueda de imágenes con Covid-19. También se puede observar que el comportamiento de CeNet y Xception aumentó considerablemente, lo cual implica que *Bridge* es un excelente método para

Model	Total Average Variance
CeNet	5.086982e-03
Xception	5.877320e-05
Unet-ML	1.111425e-02
Unet-ME	1.181236e-04
Proposal-NoConcat	9.615018e-03
Proposal-Concat	1.213864e-04

Tabla 4.5. Distribución final de varianza al utilizar el total de muestras de resultados.

potenciar modelos que no se comportan competitivamente en la búsqueda de imagen con Covid-19. Además podemos observar que en *average*, el sistema que utiliza *Xception* logra imponerse a la cabeza del resto de los modelos. El gráfico de F1-score prueba el comportamiento demostrado anteriormente pero se observa con mayor claridad la potencia de búsqueda de imágenes con Covid-19 de Xception y CeNet. Como el algoritmo Bridge potencia los resultados obtenidos con *embeddings* de Xception y CeNet, podemos decir que este método es una opción recomendable si es que se desea implementar un CBMIRS utilizando únicamente un sistema de *embedding* basado en segmentación o clasificación por separado.

.2. Nódulos Pulmonares:

.2.1. ***Cálculo de Precision y Recall en base de datos evolutiva:*** El resultado del cálculo de varianza para analizar la estabilidad de los distintos *embeddings* en los sistemas se puede observar en la figura 4.11 y en la Tabla 4.5. Para este estudio, se realizó un total de 50 consultas a la base de datos con un área de búsqueda $k = 10$ donde 25 consultas son imágenes con la enfermedad presente y 25 consultas no poseen dicha enfermedad. Presentándose la varianza de las 50 consultas en base a la cantidad de elementos nuevos en la base de datos y a su vez, la varianza utilizando la muestra total para cada modelo.

Al igual que los resultados de Covid-19, los experimentos de varianza para nódulos pulmonares son altamente estables. Ya que hubo varianzas cercanas a 0, lo que confirma la estabilidad de los modelos en el sistema. Como se tiene una menor cantidad de datos, se puede observar que la curva de varianza de la propuesta con concatenación se superpone sobre la curva de CeNet, dando a entender que existe transferencia de información en el *embedding* de parte de CeNet hacia el sistema. Respecto a Unet se comprobó que ambos modelos están sobreajustados al no generar cambio alguno en los resultados.

.2.2. ***Cálculo de Precision y Recall en base de datos estática:*** Los resultados de este experimento para nódulos pulmonares lo podemos observar en la figura 4.12, figura 4.13 y figura 4.14.

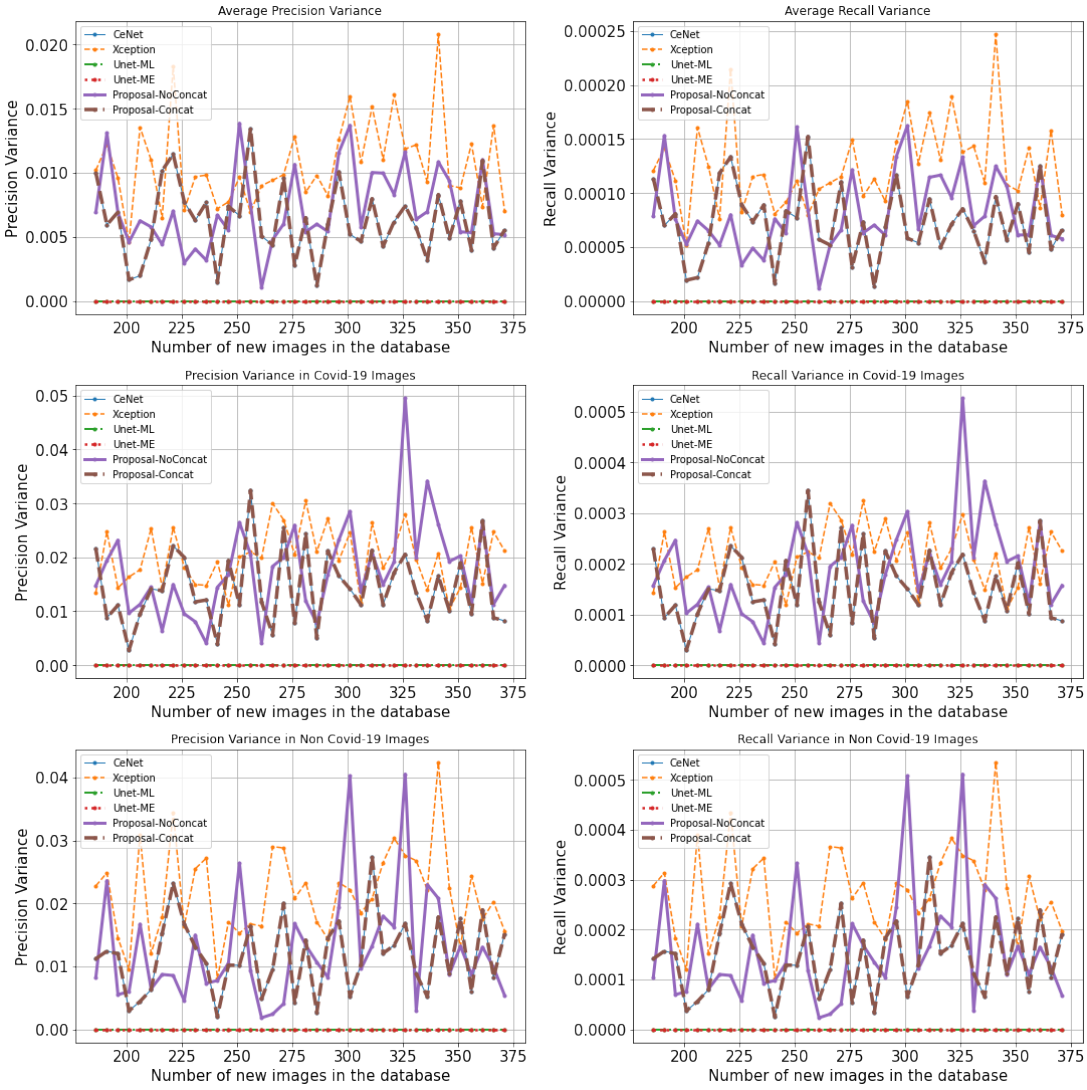


FIGURA 4.11. Distribución de varianza de precision y recall para la enfermedad de nódulos pulmonares en base de datos evolutiva.

Lo primero que se puede observar en los resultados es el perfecto rendimiento de Unet-ME en imágenes con nódulos pulmonares pero con un desastroso rendimiento en imágenes sin la enfermedad presente. Este comportamiento se debe a un claro sobre ajuste del modelo, evidencia en los gráficos de *precision* vs *recall* donde podemos observar que *precision* nunca disminuye a lo largo de *recall* en el caso de imágenes con la enfermedad, mientras que en el caso contrario, todos los valores son bastante aproximados a 0, por lo cual, este modelo se descarta en su funcionamiento.

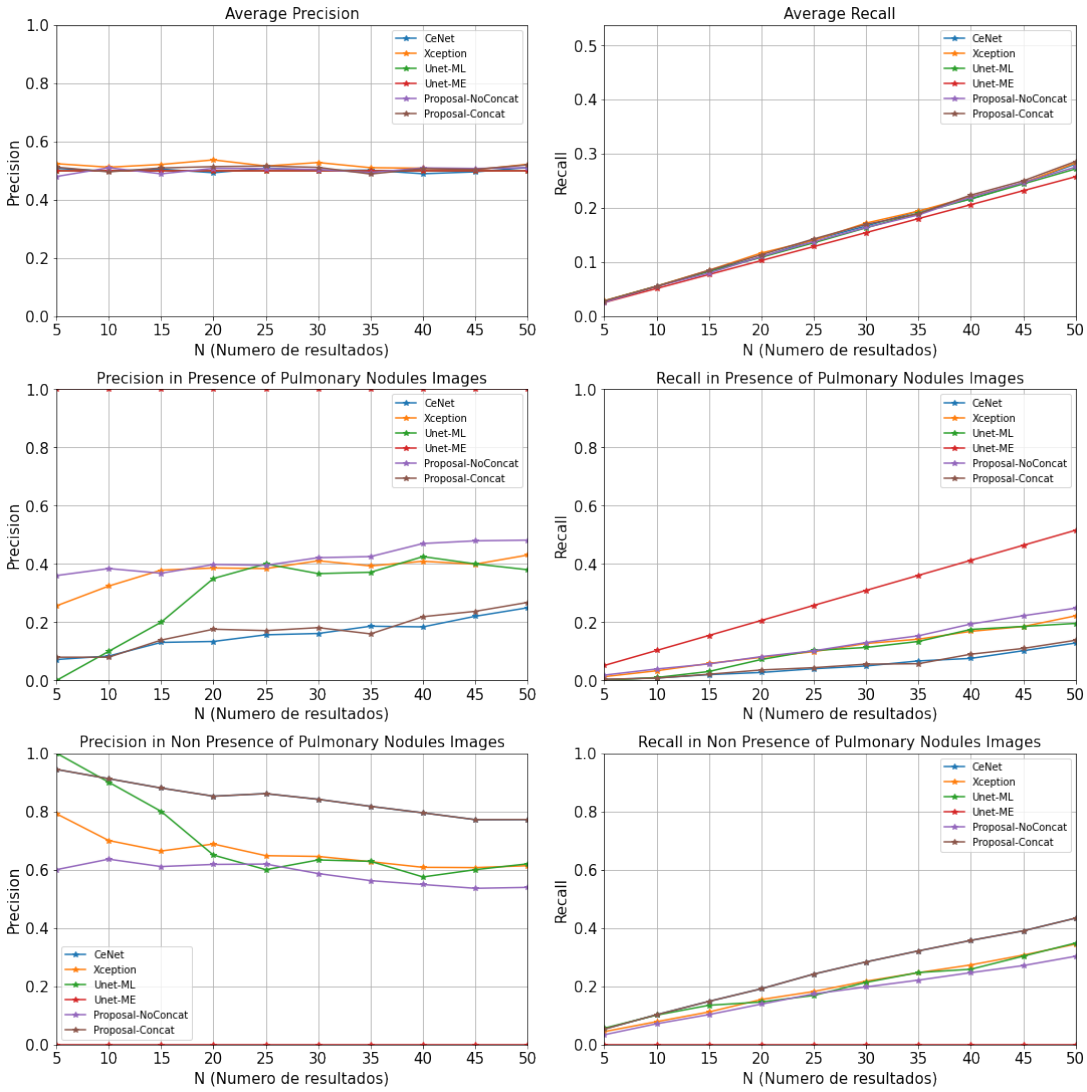


FIGURA 4.12. Resultados en el conjunto de prueba de los distintos modelos sobre el promedio de 25 búsquedas con nódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

Respecto a los otros modelos, la propuesta sin concatenación sigue teniendo los mejores resultados en comparación a su competencia, dominando tanto los gráficos de *precision* como *recall* en relación al número de imágenes retornadas. En el caso de la ausencia de nódulos pulmonares, se observó que la propuesta con concatenación se eleva

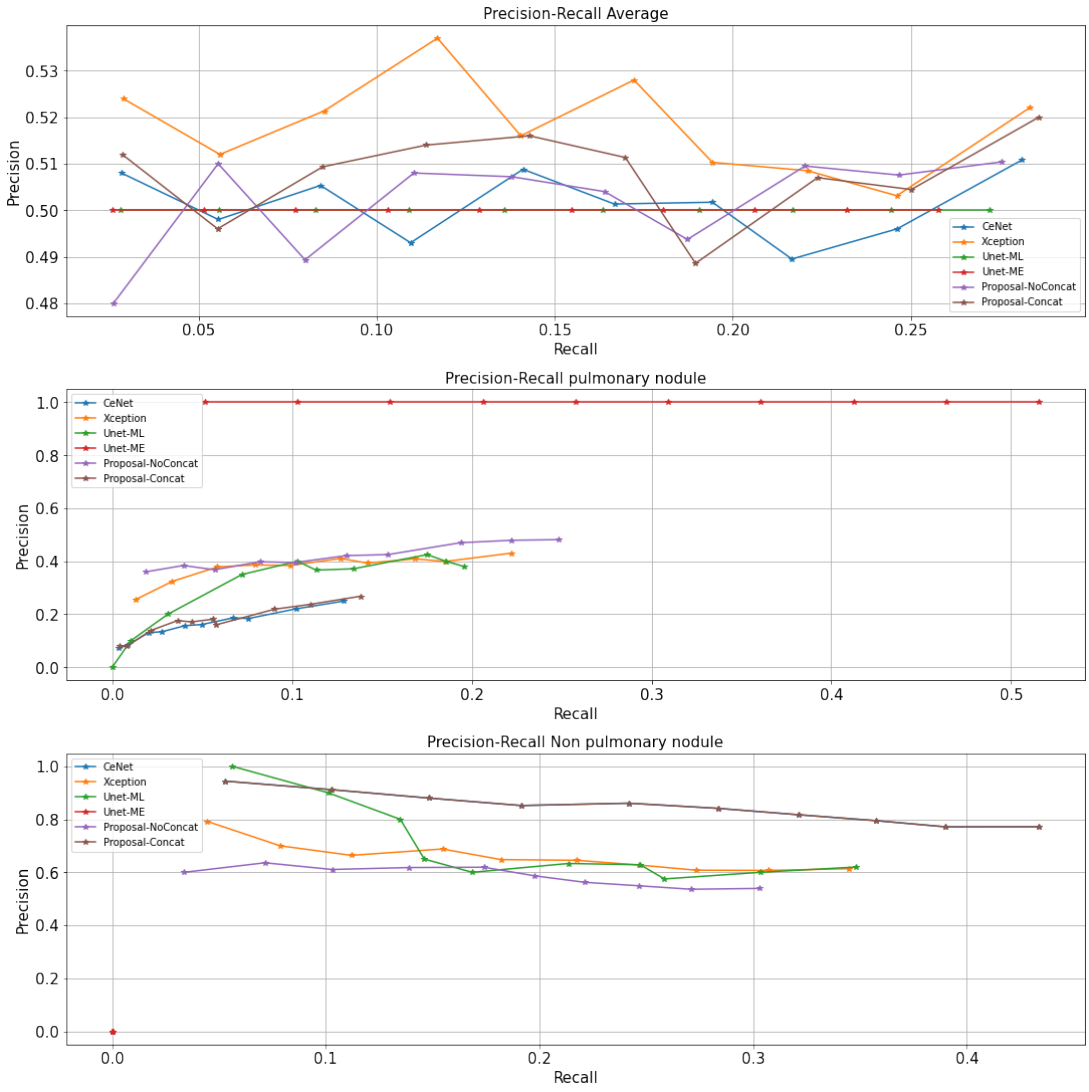


FIGURA 4.13. Gráficos *precision* vs *recall* del experimento 2 para la enfermedad de nódulos pulmonares sobre el promedio de 25 búsquedas con nódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

como el mejor modelo, esto debido a la presencia adicional de información otorgada al concatenar los resultados del segmentador. Al no existir nódulos pulmonares la segmentación de todo el pulmón aporta la suficiente información para definir intrínsecamente

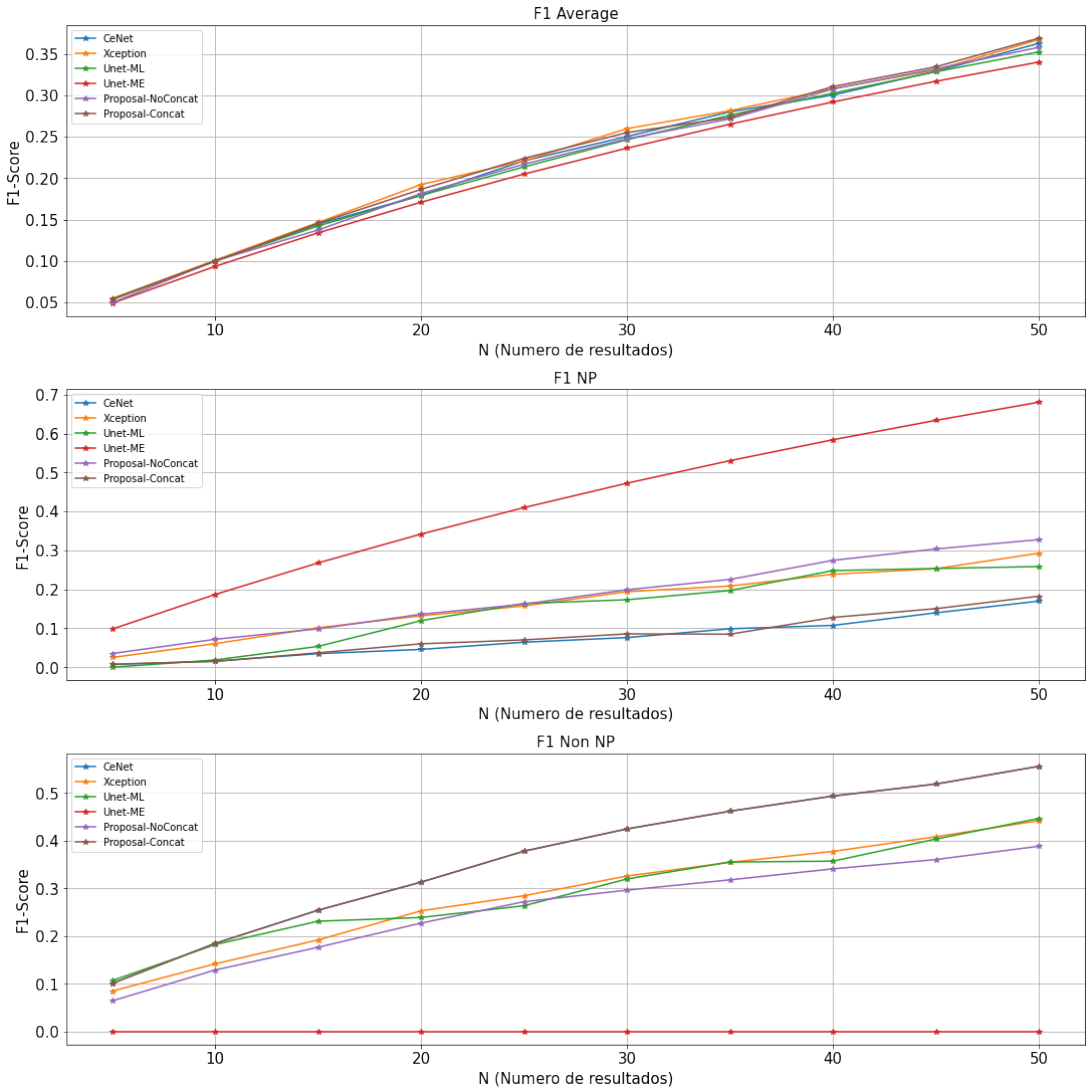


FIGURA 4.14. Gráficos *F1-Score* del experimento 3 para la enfermedad de nódulos pulmonares sobre el promedio de 25 búsquedas con cnódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

que la imagen no contiene la enfermedad. Podemos observar en la figura 4.13 que la propuesta con concatenación también logró superar a la propuesta sin concatenación cuando la cantidad de elementos retornados k es igual a $k = 30$, 35 y 40 dando un buen indicio de la factibilidad de acoplar esta información al *embedding* de búsqueda.

.2.3. ***Cálculo de Precision y Recall en base de datos estática con eliminación de near-duplicates.*** :

El efecto de la incorporación de eliminación de *near-duplicate* para el caso de nódulos pulmonares puede ser observado en la figura 4.15, 4.16 y 4.17.

Los resultados fueron bastante más pronunciados respecto a los experimentos con Covid-19. Se observó que el efecto regularizador en el sistema fué bastante mas pronunciado en este caso, donde el modelo Unet-ME logró estabilizar sus resultados considerando que es la misma red utilizada para el experimento 2, comprobando así la eficiencia del algoritmo Bridge. El comportamiento de mejorar respecto a la cantidad de elementos retornados en la búsqueda viene de la mano del mismo algoritmo, ya que *Bridge* utiliza hyper-esferas mas grandes a medida que nos alejamos del centro de búsqueda en el espacio hyper-dimensional. Este comportamiento asegura y también limita la funcionalidad de *Bridge* a bases de datos con un mínimo de elementos de casos de estudios. No se recomienda el uso de este algoritmo si se tiene un sistema con pocos estudios, porque la búsqueda resultante terminará entregando solamente imágenes similares, negando el objetivo del mismo algoritmo.

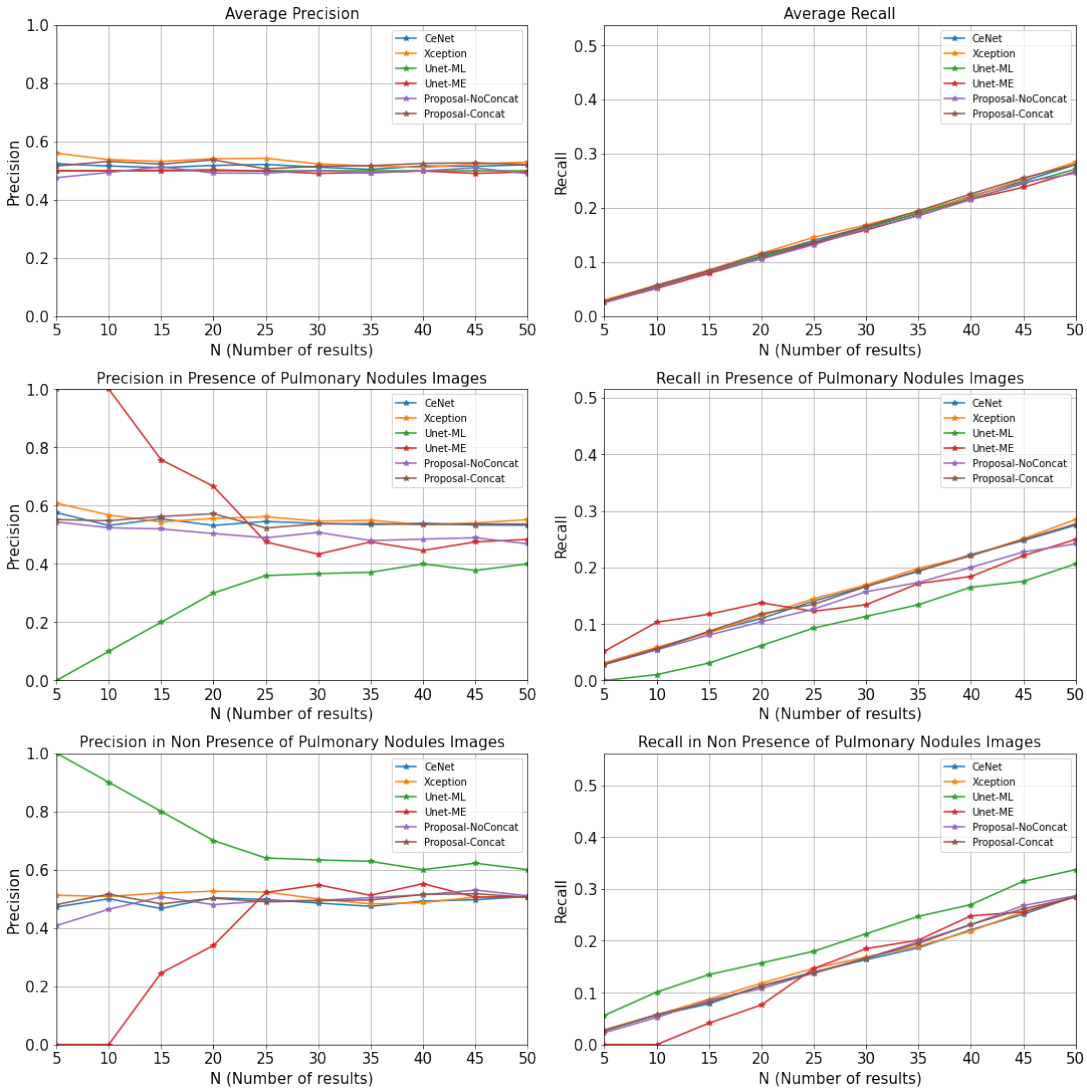


FIGURA 4.15. Resultados en el conjunto de prueba de los distintos modelos con la incorporación de eliminación de vectores *near-duplicates* por el algoritmo *Bridge* sobre el promedio de 25 búsquedas con nódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

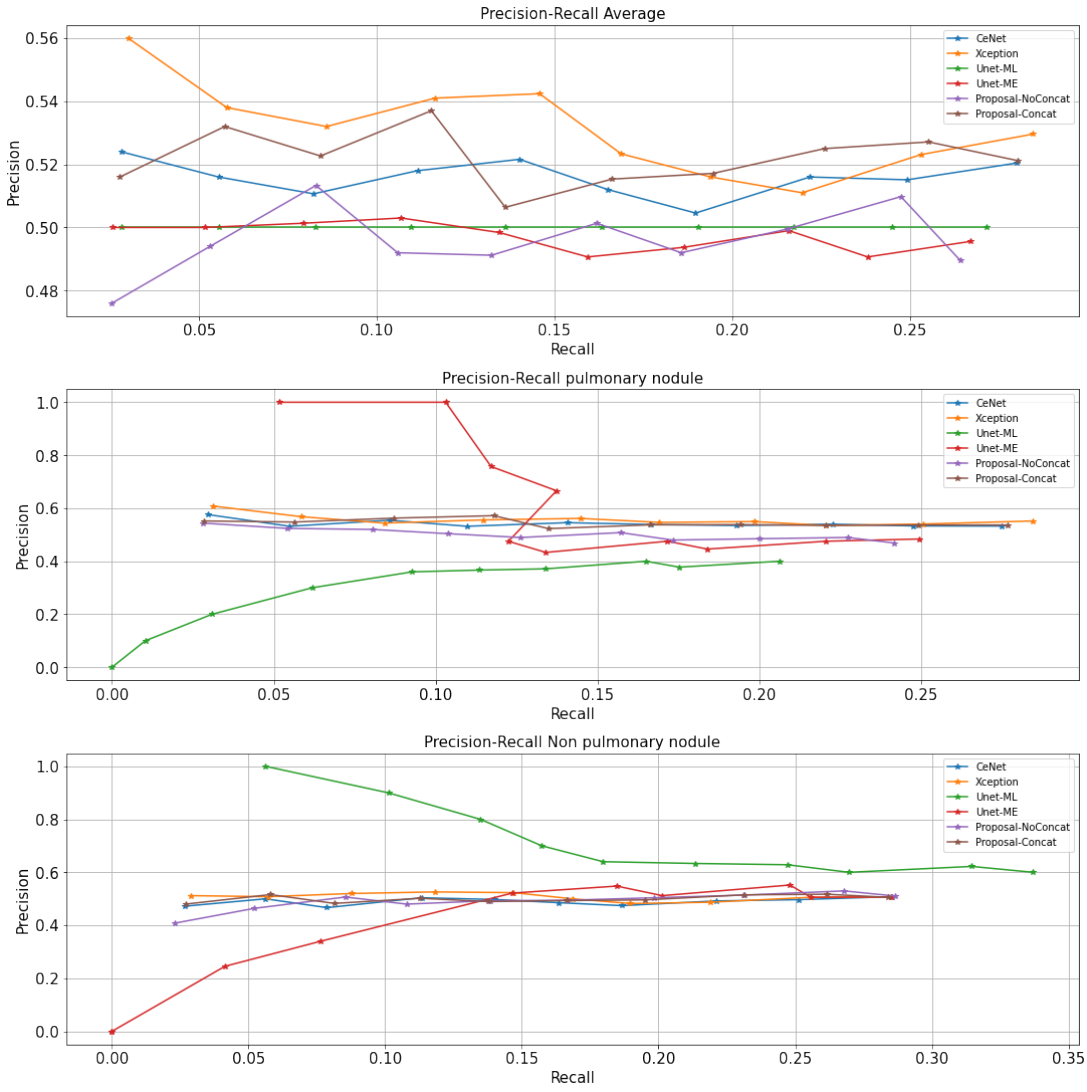


FIGURA 4.16. Gráficos *precision* vs *recall* del experimento 3 para la enfermedad nódulos pulmonares sobre el promedio de 25 búsquedas con nódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas con la incorporación de la eliminación de vectores *near-duplicate* por el algoritmo *Bridge*. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

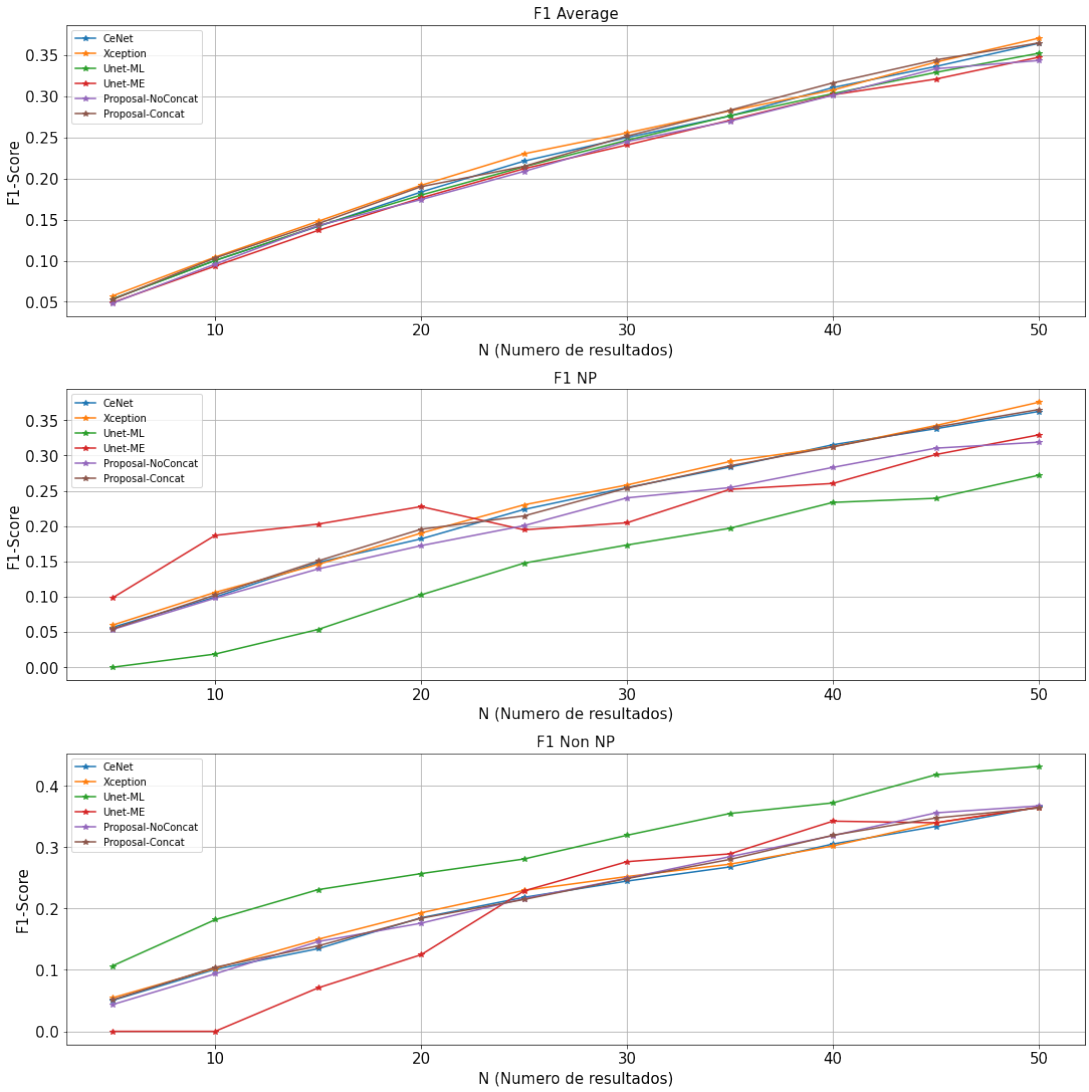


FIGURA 4.17. Gráficos F1-score del experimento 3 para la enfermedad nódulos pulmonares sobre el promedio de 25 búsquedas con nódulos pulmonares, 25 búsquedas sin nódulos pulmonares y a su vez, el promedio de las 50 búsquedas con la incorporación de la eliminación de vectores *near-duplicate* por el algoritmo *Bridge*. Los colores azul, amarillo, verde, rojo, morado y café representan los modelos CeNet, Xception, Unet-ML, Unet-ME, propuesta sin concatenación y propuesta con concatenación respectivamente que se utilizaron para la construcción del *embedding* de búsqueda.

Capítulo 5

Conclusiones

Los resultados principales y contribuciones son presentadas en este capítulo, comentando respecto a los objetivos y las hipótesis planteada en un inicio.

. Contribuciones y Alcance del Trabajo

En el presente trabajo se propusieron dos métodos principales para la resolución del problema de CBMIRS con el propósito de mejorar los métodos actuales de generación de *image embeddings* utilizados en estos sistemas. El primer método se basó en la combinación de información de CeNet y Xception, construyendo un modelo mixto que combina la información a través de los vectores resultantes de Xception, red entrenada con los *embeddings* de Cenet y con una concatenación de los mismos datos de entrada, generando un vector más rico en información. El segundo modelo propuesto tiene el mismo proceso que el primero pero no se concatenan los *embeddings* resultantes con los datos de entrada. El comportamiento de estos modelos se comparo a través de métricas establecidas en el estado del arte, siendo estas *precision* y *recall* y cuatro baseline los cuales son CBMIRS que utilizan embeddings de búsqueda construidos por las redes CeNet, Xception, Unet-ML y Unet-ME.

La capacidad de enriquecer los *embedding* que se utilizan en los CBIRS empleando *deep learning*, resulta de gran utilidad para mejorar los resultados obtenidos en búsquedas y potenciar la escalabilidad de estos sistemas, en especial en el ámbito de la medicina.

Los experimentos que se decidió llevar a cabo sobre *dataset* de Covid-19 y nódulos pulmonares, buscó comprobar el comportamiento de *precision* y *recall* en cuatro *baselines* y dos propuestas de CBMIRS simulando el ambiente real que se vive en clínicas y hospitales. Sobre esto se demostró experimentalmente que los métodos de construcción

de *embeddings* tienen un mejor desempeño en los CBMIRS si se combina información mixta de tareas diferentes que para este caso fue segmentación y clasificación. Se comprobó una mejora en la búsqueda de imágenes de una cierta enfermedad. Se demostró que la eliminación de elementos *near-duplicates* es un excelente proceso para regularizar sistemas de búsquedas de este tipo, siempre y cuando, se tenga una gran cantidad de elementos en la base de datos.

. Conclusiones del Trabajo

Los dos métodos propuestos, a diferencia de los métodos existentes en el estado del arte, presentan cualidades de generación de *image embedding* lo suficientemente altas para que un CBMIRS basado en estos modelos pueda ser escalable en un sistema clínico real, donde no todas las imágenes son similares a las usadas en el entrenamiento de los modelos. Esto permite que las propuestas puedan retornar imágenes que aporten más información al especialista consultor, permitiendo que el equipo médico pueda decidir con mayor facilidad el tratamiento o condición de un paciente. La evidencia experimental valida la hipótesis de que un CBMIRS, el cual su módulo de generación de *embeddings* es basado en un clasificador entrenado con los *embeddings* resultantes de un segmentador bajo un dominio similar, genera un sistema con vectores más ricos en información, permitiendo indexar y buscar mediante vecinos cercanos, imágenes médicas más acordes a la imagen de consulta.
La hipótesis tiene un fuerte *bias* hacia imágenes con una cierta enfermedad presente, siendo esta, Covid-19 o nódulos pulmonares, donde se observa un mejor resultado. Para el caso de Covid-19, el vector de segmentación no aporta información al clasificador ya que el modelo no reconocerá una imagen que no tenga la enfermedad presente, transformando este vector en un "*junk embedding*", explicando el mal comportamiento de la propuesta en imágenes sin Covid-19. Para nódulos pulmonares, se utilizó una segmentación de pulmón permitiendo la transferencia de información en cualquier caso, lo que demuestra el por qué para esta enfermedad los modelos propuestos superaron al resto de los *baseline* en búsqueda de la enfermedad. Esto prueba que la transferencia de información entre dos modelos es posible y potencia un sistema de búsqueda de imágenes, pero al mismo tiempo el *embedding* puede ser volátil si el segmentador se encuentra con una imagen que no puede segmentar.
En el caso de la incorporación del algoritmo *Bridge* para filtrar los elementos *near-duplicate*, se demostró que este proceso es una manera efectiva de regularizar el CBMIRS de tal forma que sistemas basados en otros modelos puedan retorna una mayor cantidad de imágenes relevantes. A su vez, se pueden generar CBMIRS en base a modelos de tareas simples por el efecto regularizante del algoritmo. Se demostró además que este algoritmo no puede ser implementado en base de datos con pocos estudios o

en búsquedas con k muy pequeños, por su naturaleza de incrementar su eficiencia en búsquedas mas amplias y a mejorar los resultados en grandes bases de datos.

. Observaciones Finales y Trabajo Futuro

Todos los modelos propuestos en el problema de CBMIRS fueron trabajados y entrenados como clasificadores binarios dada la complejidad de encontrar suficiente información para realizar un entrenamiento multi-clase, por lo cual, un estudio a futuro es unir distintos *dataset* de imágenes médicas, considerando también distintos tipos de cortes (axiales, coronales y sagitales) para entrenar redes en múltiples clases, teniendo en cuenta la incorporación de distintos modelos de segmentación.
Un factor crucial en este estudio es que todos los CT fueron descompuestos para trabajar con imágenes individuales, ya que de esta forma es más conveniente del punto de vista técnico. Sin embargo en un ambiente clínico real, esto tiene que cambiar al uso del CT completo mediante técnicas de análisis 3D, por lo tanto, un trabajo futuro será realizar búsquedas de CT completos permitiendo al equipo médico buscar por enfermedad y por corte.
Otro aspecto que podría complementar este trabajo sería un estudio completo de distintos CBMIRS y simular su comportamiento en un ambiente clínico real, para así, tener un estudio considerable de como realmente se comportan estos sistemas a lo largo del tiempo y no sobre una base de datos pseudo estática.

Bibliografía

[AIMB+11] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al., *The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans*, Medical physics **38** (2011), no. 2, 915–931.

[AKF+19] Euijoon Ahn, Ashnil Kumar, Michael Fulham, Dagan Feng, and Jinman Kim, *Convolutional sparse kernel network for unsupervised medical image analysis*, Medical image analysis **56** (2019), 140–151.

[AKG+15] Yaron Anavi, Ilya Kogan, Elad Gelbart, Ofer Geva, and Hayit Greenspan, *A comparative study for chest radiograph image retrieval using binary texture and deep learning classification*, 2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, 2015, pp. 2940–2943.

[BAN17] Christoph Baur, Shadi Albarqouni, and Nassir Navab, *Semi-supervised deep learning for fully convolutional networks*, International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 311–319.

[BKS+] Morteza Babaie, Shivam Kalra, Aditya Sriram, Christopher Mitcheltree, Shujin Zhu, Amin Khatami, Shahryar Rahnamayan, and H. R. Tizhoosh, *Classification and Retrieval of Digital Pathology Scans: A New Dataset*.

[CdBP19] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim, *Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis*, Medical image analysis **54** (2019), 280–296.

[CGC+18] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al., *Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)*, 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE, 2018, pp. 168–172.

[CHBJ+18] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al., *Opportunities and obstacles for deep learning in biology and medicine*, Journal of The Royal Society Interface **15** (2018), no. 141, 20170387.

[Cho17] François Chollet, *Xception: Deep learning with depthwise separable convolutions*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[CLQ+19] Yiheng Cai, Yuanyuan Li, Changyan Qiu, Jie Ma, and Xurong Gao, *Medical image retrieval based on convolutional neural network and supervised hashing*, IEEE Access **7** (2019), 51877–51885.

[CRC+14] Roshi Choudhary, Nikita Raina, Neeshu Chaudhary, Rashmi Chauhan, and RH Goudar, *An integrated approach to content based image retrieval*, 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2014, pp. 2404–2410.

[CZ05] Olivier Chapelle and Alexander Zien, *Semi-supervised classification by low density separation.*, AISTATS, vol. 2005, Citeseer, 2005, pp. 57–64.

[DAMMSR17] Paulo Mazzoncini De Azevedo-Marques, Arianna Mencattini, Marcello Salmeri, and Rangaraj M Rangayyan, *Medical image analysis and informatics: Computer-aided diagnosis and therapy*, CRC Press, 2017.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, *Imagenet: A large-scale hierarchical image database*, 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[DHBSM16] Alba G Seco De Herrera, Stefano Bromuri, Roger Schaer, and Henning Müller, *Overview of the medical tasks in imageclef 2016*, CLEF Working Notes. Evora, Portugal (2016).

[Gan21] Arun Gandhi, *Data augmentation: How to use deep learning when you have limited data*, Mar 2021.

[GCF+19] Zaiwang Gu, Jun Cheng, Huazhu Fu, Kang Zhou, Huaying Hao, Yitian Zhao, Tianyang Zhang, Shenghua Gao, and Jiang Liu, *Ce-net: Context encoder network for 2d medical image segmentation*, IEEE transactions on medical imaging **38** (2019), no. 10, 2281–2292.

[Ham] Daniel Hammmack, *2nd place solution to 2017 dsb - daniel hammack and julian de wit.*

[HPT+16] Ville Hyvönen, Teemu Pitkänen, Sotiris Tasoulis, Elias Jääsaari, Risto Tuomainen, Liang Wang, Jukka Corander, and Teemu Roos, *Fast nearest neighbor search through sparse random projections and voting*, Big Data (Big Data), 2016 IEEE International Conference on, IEEE, 2016, pp. 881–888.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2** (1989), no. 5, 359–366.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[Ire20] Victor E. Irekponor, *Mathematical prerequisites for understanding autoencoders and variational autoencoders (vaes)*, May 2020.

[JHR19] Elias Jääsaari, Ville Hyvönen, and Teemu Roos, *Efficient autotuning of hyperparameters in approximate nearest neighbor search*, Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, p. In press.

[Kha19] Renu Khandelwal, *Word embeddings for nlp*, Dec 2019.

[Kno18] Kasper Knol, *Creating a movie recommender using convolutional neural networks*, Feb 2018.

[Kri14] Alex Krizhevsky, *One weird trick for parallelizing convolutional neural networks*, arXiv preprint arXiv:1404.5997 (2014).

[LKB+17] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez, *A survey on deep learning in medical image analysis*, Medical image analysis **42** (2017), 60–88.

[LSK+03] Thomas Martin Lehmann, Henning Schubert, Daniel Keysers, Michael Kohnen, and Berthold B Wein, *The irma code for unique classification of medical images*, Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation, vol. 5033, International Society for Optics and Photonics, 2003, pp. 440–451.

[McG20] Kevin McGuinneess, *Transfer learning*, URL: www.slideshare.net/xavigiro/transfer-learning-d2l4-insightdcu-machine-learning-workshop-2017 (2017 (accessed July, 2020)).

[MP43] Warren S. McCulloch and Walter Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5** (1943), no. 4, 115–133 (en).

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems, 2013, pp. 3111–3119.

[Nay20] Sunita Nayak, *Image classification using transfer learning in pytorch: Learn opencv*, Dec 2020.

[OACP19] Muhammad Owais, Muhammad Arsalan, Jiho Choi, and Kang Ryoung Park, *Effective diagnosis and treatment through content-based medical image retrieval (cbmir) by using artificial intelligence*, Journal of clinical medicine **8** (2019), no. 4, 462.

[Özt20] Şaban Öztürk, *Stacked auto-encoder based tagging with deep features for content-based medical image retrieval*, Expert Systems with Applications **161** (2020), 113693.

[PSC19] Eduardo Pinho, João Figueira Silva, and Carlos Costa, *Volumetric feature learning for query-by-example in medical imaging archives*, 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), IEEE, 2019, pp. 138–143.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning, *Glove: Global vectors for word representation*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[PY09] Sinno Jialin Pan and Qiang Yang, *A survey on transfer learning*, IEEE Transactions on knowledge and data engineering **22** (2009), no. 10, 1345–1359.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, *U-net: Convolutional networks for biomedical image segmentation*, International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[RHW85] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, *Learning Internal Representations by Error Propagation*, Tech. Report ICS-8506, CALIFORNIA UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE, CALIFORNIA UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE, September 1985.

[SBPS+14] Lúcio FD Santos, Marcos VN Bedo, Marcelo Ponciano-Silva, Agma JM Traina, and Caetano Traina, *Being similar is not enough: How to bridge usability gap through diversity in medical images*, 2014 IEEE 27th International Symposium on Computer-Based Medical Systems, IEEE, 2014, pp. 287–293.

[SOF+13] Lucio FD Santos, Willian D Oliveira, Monica RP Ferreira, Agma JM Traina, and Caetano Traina Jr, *Parameter-free and domain-independent similarity search with diversity*, Proceedings of the 25th International Conference on Scientific and Statistical Database Management, 2013, pp. 1–12.

[SPK19] R Rani Saritha, Varghese Paul, and P Ganesh Kumar, *Content based image retrieval using deep learning process*, Cluster Computing **22** (2019), no. 2, 4187–4200.

[STDB+17] Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira SN Berens, Cas van den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, Bram Geurts, et al., *Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge*, Medical image analysis **42** (2017), 1–13.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, *Rethinking the inception architecture for computer vision*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[SZ14] Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556 (2014).

[Tsa19] Sik-Ho Tsang, *Review: Xception - with depthwise separable convolution, better than inception-v3 (image...*, Mar 2019.

[WMQQ16] Guohui Wei, He Ma, Wei Qian, and Min Qiu, *Similarity measurement of lung masses for medical image retrieval using kernel based semisupervised distance metric*, Medical physics **43** (2016), no. 12, 6259–6269.

[YZL+20] Pengshuai Yang, Yupeng Zhai, Lin Li, Hairong Lv, Jigang Wang, Chengzhan Zhu, and Rui Jiang, *A deep metric learning approach for histopathological image retrieval*, Methods (2020).

[ZLFX18] Wentao Zhu, Chaochun Liu, Wei Fan, and Xiaohui Xie, *Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification*, 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 673–681.

[ZLT17] Wengang Zhou, Houqiang Li, and Qi Tian, *Recent advance in content-based image retrieval: A literature survey*, arXiv preprint arXiv:1706.06064 (2017).

Apéndice A

Arquitectura Xception

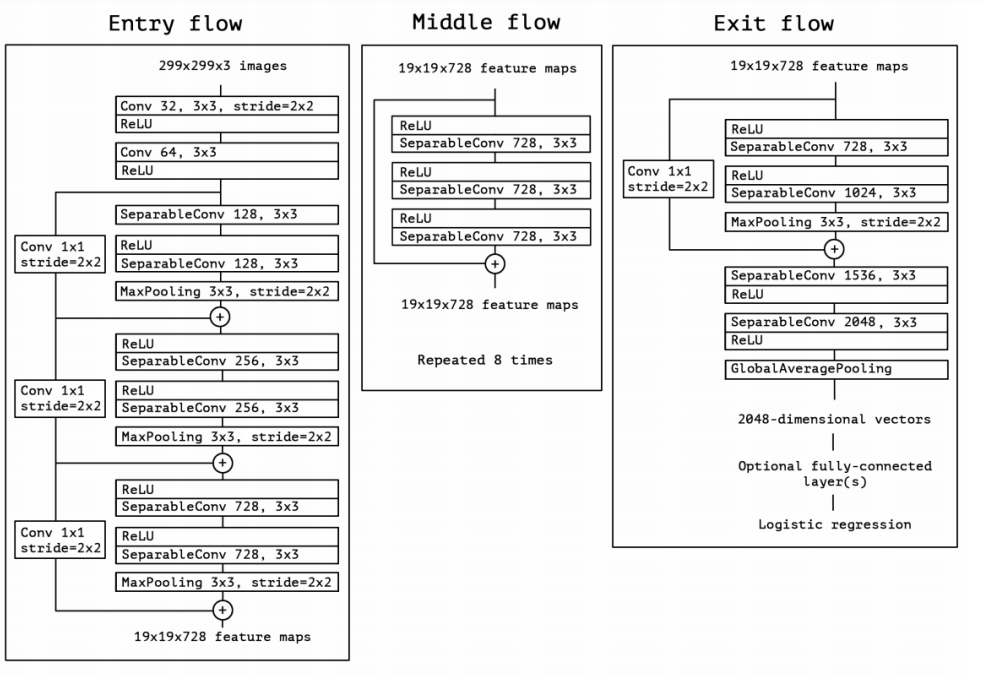


Figura A.1. Modelo completo de Xception
(Fuente: Paper [**Cho17**])

. Derivaciones de Modelo **C-MoA**

Apéndice B

Detalles de Experimentación

. Experimento 2 Covid-19

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.4960	0.4860	0.4893	0.5040	0.5072	0.5173	0.5251	0.5225	0.5164	0.5048
Xception	0.5480	0.5120	0.5053	0.5140	0.5112	0.5173	0.5177	0.5085	0.5169	0.5084
Unet-ML	0.3960	0.4320	0.4213	0.4350	0.4456	0.4440	0.4497	0.4615	0.4578	0.4988
Unet-ME	0.4800	0.4740	0.4813	0.4870	0.4904	0.4893	0.4863	0.4835	0.4862	0.4880
Proposal-NoConcat	0.5680	0.5220	0.5320	0.5340	0.5112	0.5093	0.5171	0.4985	0.5022	0.5072
Proposal-Concat	0.5280	0.4860	0.4973	0.4690	0.5128	0.4940	0.5017	0.5060	0.4982	0.4932

Tabla B.1. Promedio de *Precision* para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0198	0.0387	0.0585	0.0804	0.1011	0.1239	0.1468	0.1670	0.1857	0.2017
Xception	0.0219	0.0408	0.0603	0.0819	0.1018	0.1237	0.1445	0.1621	0.1854	0.2027
Unet-ML	0.0156	0.0343	0.0501	0.0691	0.0885	0.1059	0.1253	0.1469	0.1640	0.1992
Unet-ME	0.0190	0.0374	0.0570	0.0769	0.0967	0.1158	0.1343	0.1526	0.1727	0.1926
Proposal-NoConcat	0.0228	0.0418	0.0639	0.0855	0.1024	0.1224	0.1450	0.1596	0.1810	0.2031
Proposal-Concat	0.0211	0.0388	0.0596	0.0749	0.1025	0.1184	0.1404	0.1619	0.1794	0.1972

Tabla B.2. Promedio de *Recall* para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.2880	0.2840	0.2987	0.3220	0.3232	0.3427	0.3634	0.3740	0.3662	0.3608
Xception	0.3520	0.2960	0.2720	0.3060	0.2880	0.3040	0.3234	0.3030	0.3111	0.3160
Unet-ML	0.0640	0.1520	0.1413	0.1880	0.1904	0.2147	0.2400	0.2450	0.2524	0.3448
Unet-ME	0.0720	0.0640	0.0720	0.0560	0.0528	0.0467	0.0457	0.0470	0.0516	0.0552
Proposal-NoConcat	0.5120	0.4440	0.4587	0.4420	0.4304	0.4280	0.4400	0.4020	0.4160	0.4232
Proposal-Concat	0.3680	0.3520	0.3627	0.3220	0.3920	0.3573	0.3806	0.3950	0.3902	0.3768

TABLA B.3. Promedio de *Precision* para 25 búsquedas de covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0118	0.0233	0.0367	0.0528	0.0662	0.0843	0.1043	0.1226	0.1351	0.1479
Xception	0.0144	0.0243	0.0334	0.0502	0.0590	0.0748	0.0928	0.0993	0.1148	0.1295
Unet-ML	0.0026	0.0125	0.0174	0.0308	0.0390	0.0528	0.0689	0.0803	0.0931	0.1413
Unet-ME	0.0030	0.0052	0.0089	0.0092	0.0108	0.0115	0.0131	0.0154	0.0190	0.0226
Proposal-NoConcat	0.0210	0.0364	0.0564	0.0725	0.0882	0.1052	0.1262	0.1318	0.1534	0.1734
Proposal-Concat	0.0151	0.0289	0.0446	0.0528	0.0803	0.0879	0.1092	0.1295	0.1439	0.1544

TABLA B.4. Promedio de *Recall* para 25 búsquedas de covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.7040	0.6880	0.6800	0.6860	0.6912	0.6920	0.6869	0.6710	0.6667	0.6488
Xception	0.7440	0.7280	0.7387	0.7220	0.7344	0.7307	0.7120	0.7140	0.7227	0.7008
Unet-ML	0.7280	0.7120	0.7013	0.6820	0.7008	0.6733	0.6594	0.6780	0.6631	0.6528
Unet-ME	0.8880	0.8840	0.8907	0.9180	0.9280	0.9320	0.9269	0.9200	0.9209	0.9208
Proposal-NoConcat	0.6240	0.6000	0.6053	0.6260	0.5920	0.5907	0.5943	0.5950	0.5884	0.5912
Proposal-Concat	0.6880	0.6200	0.6320	0.6160	0.6336	0.6307	0.6229	0.6170	0.6062	0.6096

TABLA B.5. Promedio de *Precision* para 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0277	0.0542	0.0803	0.1080	0.1361	0.1635	0.1893	0.2113	0.2362	0.2554
Xception	0.0293	0.0573	0.0872	0.1137	0.1446	0.1726	0.1962	0.2249	0.2561	0.2759
Unet-ML	0.0287	0.0561	0.0828	0.1074	0.1380	0.1591	0.1817	0.2135	0.2350	0.2570
Unet-ME	0.0350	0.0696	0.1052	0.1446	0.1827	0.2202	0.2554	0.2898	0.3263	0.3625
Proposal-NoConcat	0.0246	0.0472	0.0715	0.0986	0.1165	0.1395	0.1638	0.1874	0.2085	0.2328
Proposal-Concat	0.0271	0.0488	0.0746	0.0970	0.1247	0.1490	0.1717	0.1943	0.2148	0.2400

TABLA B.6. Promedio de *Recall* para 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0380	0.0717	0.1045	0.1387	0.1687	0.1999	0.2294	0.2531	0.2731	0.2882
Xception	0.0420	0.0756	0.1078	0.1413	0.1698	0.1996	0.2259	0.2458	0.2729	0.2898
Unet-ML	0.0301	0.0635	0.0896	0.1193	0.1476	0.1710	0.1960	0.2229	0.2415	0.2847
Unet-ME	0.0365	0.0694	0.1020	0.1328	0.1616	0.1873	0.2104	0.2320	0.2548	0.2762
Proposal-NoConcat	0.0438	0.0774	0.1142	0.1474	0.1706	0.1974	0.2265	0.2418	0.2661	0.2901
Proposal-Concat	0.0405	0.0719	0.1065	0.1292	0.1709	0.1910	0.2194	0.2453	0.2638	0.2818

Tabla B.7. F1-Score para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0227	0.0430	0.0654	0.0907	0.1099	0.1353	0.1620	0.1847	0.1974	0.2098
Xception	0.0277	0.0448	0.0596	0.0862	0.0980	0.1200	0.1442	0.1496	0.1677	0.1837
Unet-ML	0.0050	0.0230	0.0309	0.0530	0.0648	0.0847	0.1070	0.1210	0.1360	0.2005
Unet-ME	0.0057	0.0097	0.0158	0.0158	0.0180	0.0184	0.0204	0.0232	0.0278	0.0321
Proposal-NoConcat	0.0403	0.0673	0.1004	0.1245	0.1464	0.1689	0.1962	0.1985	0.2242	0.2460
Proposal-Concat	0.0290	0.0533	0.0794	0.0907	0.1333	0.1411	0.1697	0.1951	0.2103	0.2191

Tabla B.8. F1-Score para 25 búsquedas con covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0533	0.1004	0.1437	0.1867	0.2274	0.2645	0.2968	0.3214	0.3488	0.3666
Xception	0.0564	0.1063	0.1561	0.1965	0.2416	0.2792	0.3077	0.3420	0.3781	0.3959
Unet-ML	0.0552	0.1039	0.1482	0.1856	0.2305	0.2573	0.2849	0.3248	0.3470	0.3688
Unet-ME	0.0673	0.1291	0.1882	0.2498	0.3053	0.3562	0.4005	0.4407	0.4819	0.5202
Proposal-NoConcat	0.0473	0.0876	0.1279	0.1703	0.1947	0.2257	0.2568	0.2850	0.3079	0.3340
Proposal-Concat	0.0521	0.0905	0.1335	0.1676	0.2084	0.2410	0.2691	0.2956	0.3172	0.3444

Tabla B.9. F1-Score para 25 búsquedas sin covid-19.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.4240	0.4880	0.4813	0.4850	0.4984	0.4807	0.4949	0.4910	0.5138	0.4992
Xception	0.5480	0.5380	0.5120	0.5190	0.5032	0.5153	0.5251	0.5260	0.5307	0.5184
Unet-ML	0.3760	0.4060	0.4333	0.4630	0.4648	0.4693	0.4800	0.4785	0.4809	0.4784
Unet-ME	0.4560	0.4660	0.4653	0.4710	0.4728	0.4687	0.4760	0.4720	0.4760	0.4772
Proposal-NoConcat	0.4920	0.5040	0.5213	0.4800	0.5008	0.4653	0.5051	0.4995	0.4996	0.4968
Proposal-Concat	0.4600	0.4740	0.4587	0.4790	0.4672	0.4900	0.5080	0.5105	0.5009	0.4900

TABLA B.10. Promedio de *Precision* para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0169	0.0389	0.0576	0.0775	0.0996	0.1154	0.1387	0.1573	0.1852	0.1998
Xception	0.0219	0.0430	0.0613	0.0830	0.1006	0.1237	0.1471	0.1683	0.1912	0.2075
Unet-ML	0.0149	0.0322	0.0516	0.0737	0.0925	0.1121	0.1340	0.1527	0.1726	0.1910
Unet-ME	0.0180	0.0368	0.0552	0.0745	0.0935	0.1111	0.1317	0.1493	0.1694	0.1887
Proposal-NoConcat	0.0197	0.0404	0.0627	0.0769	0.1004	0.1120	0.1418	0.1603	0.1803	0.1994
Proposal-Concat	0.0184	0.0379	0.0550	0.0767	0.0935	0.1178	0.1426	0.1638	0.1808	0.1964

TABLA B.11. Promedio de *Recall* para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.2400	0.3240	0.3173	0.3580	0.3760	0.3853	0.4069	0.4170	0.4329	0.4104
Xception	0.3760	0.3760	0.3573	0.3820	0.3824	0.4053	0.4251	0.4140	0.4311	0.4184
Unet-ML	0.1040	0.1640	0.1840	0.2320	0.2608	0.2600	0.2994	0.3100	0.3084	0.3256
Unet-ME	0.0800	0.0880	0.0933	0.1000	0.0960	0.0893	0.0949	0.1010	0.1084	0.1080
Proposal-NoConcat	0.4480	0.4600	0.4560	0.4200	0.4464	0.4293	0.4503	0.4670	0.4578	0.4680
Proposal-Concat	0.3840	0.3560	0.3307	0.3880	0.3840	0.4240	0.4594	0.4690	0.4516	0.4368

TABLA B.12. Promedio de *Precision* para 25 búsquedas de covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0098	0.0266	0.0390	0.0587	0.0770	0.0948	0.1167	0.1367	0.1597	0.1682
Xception	0.0154	0.0308	0.0439	0.0626	0.0784	0.0997	0.1220	0.1357	0.1590	0.1715
Unet-ML	0.0043	0.0134	0.0226	0.0380	0.0534	0.0639	0.0859	0.1016	0.1138	0.1334
Unet-ME	0.0033	0.0072	0.0115	0.0164	0.0197	0.0220	0.0272	0.0331	0.0400	0.0443
Proposal-NoConcat	0.0184	0.0377	0.0561	0.0689	0.0915	0.1056	0.1292	0.1531	0.1689	0.1918
Proposal-Concat	0.0157	0.0292	0.0407	0.0636	0.0787	0.1043	0.1318	0.1538	0.1666	0.1790

Tabla B.13. Promedio de *Recall* para 25 búsquedas de covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.6080	0.6520	0.6453	0.6120	0.6208	0.5760	0.5829	0.5650	0.5947	0.5880
Xception	0.7200	0.7000	0.6667	0.6560	0.6240	0.6253	0.6251	0.6380	0.6302	0.6184
Unet-ML	0.6480	0.6480	0.6827	0.6940	0.6688	0.6787	0.6606	0.6470	0.6533	0.6312
Unet-ME	0.8320	0.8440	0.8373	0.8420	0.8496	0.8480	0.8571	0.8430	0.8436	0.8464
Proposal-NoConcat	0.5360	0.5480	0.5867	0.5400	0.5552	0.5013	0.5600	0.5320	0.5413	0.5256
Proposal-Concat	0.5360	0.5920	0.5867	0.5700	0.5504	0.5560	0.5566	0.5520	0.5502	0.5432

Tabla B.14. Promedio de *Precision* para 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0239	0.0513	0.0762	0.0964	0.1222	0.1361	0.1606	0.1780	0.2107	0.2315
Xception	0.0283	0.0551	0.0787	0.1033	0.1228	0.1477	0.1723	0.2009	0.2233	0.2435
Unet-ML	0.0255	0.0510	0.0806	0.1093	0.1317	0.1603	0.1820	0.2038	0.2315	0.2485
Unet-ME	0.0328	0.0665	0.0989	0.1326	0.1672	0.2003	0.2362	0.2655	0.2989	0.3332
Proposal-NoConcat	0.0211	0.0432	0.0693	0.0850	0.1093	0.1184	0.1543	0.1676	0.1918	0.2069
Proposal-Concat	0.0211	0.0466	0.0693	0.0898	0.1083	0.1313	0.1534	0.1739	0.1950	0.2139

Tabla B.15. Promedio de *Recall* para 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0325	0.0721	0.1029	0.1337	0.1661	0.1861	0.2166	0.2383	0.2722	0.2854
Xception	0.0421	0.0796	0.1096	0.1431	0.1677	0.1995	0.2299	0.2551	0.2811	0.2963
Unet-ML	0.0286	0.0597	0.0923	0.1271	0.1544	0.1810	0.2095	0.2315	0.2541	0.2730
Unet-ME	0.0347	0.0683	0.0987	0.1286	0.1561	0.1797	0.2063	0.2269	0.2499	0.2705
Proposal-NoConcat	0.0379	0.0749	0.1119	0.1326	0.1672	0.1805	0.2214	0.2428	0.2650	0.2845
Proposal-Concat	0.0354	0.0702	0.0982	0.1322	0.1558	0.1899	0.2227	0.2480	0.2657	0.2804

Tabla B.16. F1-Score para 25 búsquedas de covid-19 y 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0189	0.0491	0.0695	0.1008	0.1279	0.1521	0.1814	0.2059	0.2333	0.2386
Xception	0.0296	0.0570	0.0782	0.1076	0.1301	0.1600	0.1896	0.2044	0.2323	0.2433
Unet-ML	0.0082	0.0248	0.0403	0.0654	0.0887	0.1026	0.1335	0.1531	0.1662	0.1893
Unet-ME	0.0063	0.0133	0.0204	0.0282	0.0327	0.0353	0.0423	0.0499	0.0584	0.0628
Proposal-NoConcat	0.0353	0.0697	0.0999	0.1183	0.1518	0.1695	0.2008	0.2306	0.2467	0.2721
Proposal-Concat	0.0302	0.0539	0.0724	0.1093	0.1306	0.1674	0.2048	0.2316	0.2434	0.2540

Tabla B.17. F1-Score para 25 búsquedas de covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0461	0.0952	0.1363	0.1665	0.2042	0.2201	0.2519	0.2707	0.3112	0.3322
Xception	0.0545	0.1022	0.1408	0.1785	0.2053	0.2390	0.2701	0.3056	0.3298	0.3494
Unet-ML	0.0491	0.0946	0.1442	0.1888	0.2200	0.2594	0.2854	0.3099	0.3419	0.3566
Unet-ME	0.0630	0.1232	0.1769	0.2291	0.2795	0.3241	0.3704	0.4038	0.4414	0.4782
Proposal-NoConcat	0.0406	0.0800	0.1239	0.1469	0.1826	0.1916	0.2420	0.2549	0.2833	0.2969
Proposal-Concat	0.0406	0.0864	0.1239	0.1551	0.1811	0.2125	0.2405	0.2644	0.2879	0.3069

Tabla B.18. F1-Score para 25 búsquedas sin covid-19 con la inclusión del algoritmo Bridge para detectar near duplicates.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.5080	0.4980	0.5053	0.4930	0.5088	0.5013	0.5017	0.4895	0.4960	0.5108
Xception	0.5240	0.5120	0.5213	0.5370	0.5160	0.5280	0.5103	0.5085	0.5031	0.5220
Unet-ML	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Unet-ME	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Proposal-NoConcat	0.4800	0.5100	0.4893	0.5080	0.5072	0.5040	0.4937	0.5095	0.5076	0.5104
Proposal-Concat	0.5120	0.4960	0.5093	0.5140	0.5160	0.5113	0.4886	0.5070	0.5044	0.5200

TABLA B.19. Promedio de *Precision* para 25 búsquedas de nódulos pulmonares y 25 búsquedas sin nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0284	0.0556	0.0843	0.1095	0.1411	0.1667	0.1943	0.2166	0.2462	0.2812
Xception	0.0288	0.0560	0.0852	0.1171	0.1405	0.1723	0.1943	0.2210	0.2461	0.2833
Unet-ML	0.0281	0.0557	0.0829	0.1091	0.1358	0.1634	0.1906	0.2168	0.2445	0.2721
Unet-ME	0.0258	0.0515	0.0773	0.1031	0.1289	0.1546	0.1804	0.2062	0.2320	0.2577
Proposal-NoConcat	0.0261	0.0555	0.0799	0.1105	0.1379	0.1640	0.1873	0.2203	0.2466	0.2756
Proposal-Concat	0.0286	0.0554	0.0849	0.1139	0.1430	0.1698	0.1895	0.2238	0.2501	0.2859

TABLA B.20. Promedio de *Recall* para 25 búsquedas de nódulos pulmonares y 25 búsquedas sin nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0720	0.0840	0.1307	0.1340	0.1568	0.1613	0.1863	0.1840	0.2204	0.2496
Xception	0.2560	0.3240	0.3787	0.3860	0.3840	0.4107	0.3931	0.4090	0.3991	0.4304
Unet-ML	0.0000	0.1000	0.2000	0.3500	0.4000	0.3667	0.3714	0.4250	0.4000	0.3800
Unet-ME	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Proposal-NoConcat	0.3600	0.3840	0.3680	0.3980	0.3952	0.4213	0.4251	0.4700	0.4791	0.4816
Proposal-Concat	0.0800	0.0800	0.1387	0.1760	0.1712	0.1813	0.1600	0.2190	0.2373	0.2680

TABLA B.21. Promedio de *Precision* para 25 búsquedas de nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0037	0.0087	0.0202	0.0276	0.0404	0.0499	0.0672	0.0759	0.1023	0.1287
Xception	0.0132	0.0334	0.0586	0.0796	0.0990	0.1270	0.1419	0.1687	0.1852	0.2219
Unet-ML	0.0000	0.0103	0.0309	0.0722	0.1031	0.1134	0.1340	0.1753	0.1856	0.1959
Unet-ME	0.0515	0.1031	0.1546	0.2062	0.2577	0.3093	0.3608	0.4124	0.4639	0.5155
Proposal-NoConcat	0.0186	0.0396	0.0569	0.0821	0.1019	0.1303	0.1534	0.1938	0.2223	0.2482
Proposal-Concat	0.0041	0.0082	0.0214	0.0363	0.0441	0.0561	0.0577	0.0903	0.1101	0.1381

TABLA B.22. Promedio de *Recall* para 25 búsquedas de nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0530	0.1025	0.1483	0.1915	0.2418	0.2836	0.3213	0.3573	0.3901	0.4337
Xception	0.0445	0.0787	0.1119	0.1546	0.1820	0.2175	0.2467	0.2733	0.3070	0.3447
Unet-ML	0.0562	0.1011	0.1348	0.1461	0.1685	0.2135	0.2472	0.2584	0.3034	0.3483
Unet-ME	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Proposal-NoConcat	0.0337	0.0715	0.1029	0.1389	0.1739	0.1978	0.2211	0.2467	0.2710	0.3029
Proposal-Concat	0.0530	0.1025	0.1483	0.1915	0.2418	0.2836	0.3213	0.3573	0.3901	0.4337

TABLA B.24. Promedio de *Recall* para 25 búsquedas sin nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.9440	0.9120	0.8800	0.8520	0.8608	0.8413	0.8171	0.7950	0.7716	0.7720
Xception	0.7920	0.7000	0.6640	0.6880	0.6480	0.6453	0.6274	0.6080	0.6071	0.6136
Unet-ML	1.0000	0.9000	0.8000	0.6500	0.6000	0.6333	0.6286	0.5750	0.6000	0.6200
Unet-ME	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Proposal-NoConcat	0.6000	0.6360	0.6107	0.6180	0.6192	0.5867	0.5623	0.5490	0.5360	0.5392
Proposal-Concat	0.9440	0.9120	0.8800	0.8520	0.8608	0.8413	0.8171	0.7950	0.7716	0.7720

TABLA B.23. Promedio de *Precision* para 25 búsquedas sin nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.5080	0.4980	0.5053	0.4930	0.5088	0.5013	0.5017	0.4895	0.4960	0.5108
Xception	0.5240	0.5120	0.5213	0.5370	0.5160	0.5280	0.5103	0.5085	0.5031	0.5220
Unet-ML	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Unet-ME	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Proposal-NoConcat	0.4800	0.5100	0.4893	0.5080	0.5072	0.5040	0.4937	0.5095	0.5076	0.5104
Proposal-Concat	0.5120	0.4960	0.5093	0.5140	0.5160	0.5113	0.4886	0.5070	0.5044	0.5200

TABLA B.25. Promedio de F1 para 25 búsquedas con nódulos pulmonares y 25 búsquedas sin nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0284	0.0556	0.0843	0.1095	0.1411	0.1667	0.1943	0.2166	0.2462	0.2812
Xception	0.0288	0.0560	0.0852	0.1171	0.1405	0.1723	0.1943	0.2210	0.2461	0.2833
Unet-ML	0.0281	0.0557	0.0829	0.1091	0.1358	0.1634	0.1906	0.2168	0.2445	0.2721
Unet-ME	0.0258	0.0515	0.0773	0.1031	0.1289	0.1546	0.1804	0.2062	0.2320	0.2577
Proposal-NoConcat	0.0261	0.0555	0.0799	0.1105	0.1379	0.1640	0.1873	0.2203	0.2466	0.2756
Proposal-Concat	0.0286	0.0554	0.0849	0.1139	0.1430	0.1698	0.1895	0.2238	0.2501	0.2859

TABLA B.26. Promedio de F1 para 25 búsquedas con nódulos pulmonares.

Modelos \ K	5	10	15	20	25	30	35	40	45	50
CeNet	0.0720	0.0840	0.1307	0.1340	0.1568	0.1613	0.1863	0.1840	0.2204	0.2496
Xception	0.2560	0.3240	0.3787	0.3860	0.3840	0.4107	0.3931	0.4090	0.3991	0.4304
Unet-ML	0.0000	0.1000	0.2000	0.3500	0.4000	0.3667	0.3714	0.4250	0.4000	0.3800
Unet-ME	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Proposal-NoConcat	0.3600	0.3840	0.3680	0.3980	0.3952	0.4213	0.4251	0.4700	0.4791	0.4816
Proposal-Concat	0.0800	0.0800	0.1387	0.1760	0.1712	0.1813	0.1600	0.2190	0.2373	0.2680

Tabla B.27. Promedio de F1 para 25 sin búsquedas con nódulos pulmonares.

. Experimento 3 Nódulos Pulmonares

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.5080	0.4980	0.5053	0.4930	0.5088	0.5013	0.5017	0.4895	0.4960	0.5108
Xception	0.5240	0.5120	0.5213	0.5370	0.5160	0.5280	0.5103	0.5085	0.5031	0.5220
Unet-ML	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Unet-ME	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Proposal-NoConcat	0.4800	0.5100	0.4893	0.5080	0.5072	0.5040	0.4937	0.5095	0.5076	0.5104
Proposal-Concat	0.5120	0.4960	0.5093	0.5140	0.5160	0.5113	0.4886	0.5070	0.5044	0.5200

Tabla B.28. Promedio de *Precision* para 25 búsquedas de nódulos pulmonares y 25 búsquedas sin nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0284	0.0556	0.0843	0.1095	0.1411	0.1667	0.1943	0.2166	0.2462	0.2812
Xception	0.0288	0.0560	0.0852	0.1171	0.1405	0.1723	0.1943	0.2210	0.2461	0.2833
Unet-ML	0.0281	0.0557	0.0829	0.1091	0.1358	0.1634	0.1906	0.2168	0.2445	0.2721
Unet-ME	0.0258	0.0515	0.0773	0.1031	0.1289	0.1546	0.1804	0.2062	0.2320	0.2577
Proposal-NoConcat	0.0261	0.0555	0.0799	0.1105	0.1379	0.1640	0.1873	0.2203	0.2466	0.2756
Proposal-Concat	0.0286	0.0554	0.0849	0.1139	0.1430	0.1698	0.1895	0.2238	0.2501	0.2859

Tabla B.29. Promedio de *Recall* para 25 búsquedas de nódulos pulmonares y 25 búsquedas sin nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0720	0.0840	0.1307	0.1340	0.1568	0.1613	0.1863	0.1840	0.2204	0.2496
Xception	0.2560	0.3240	0.3787	0.3860	0.3840	0.4107	0.3931	0.4090	0.3991	0.4304
Unet-ML	0.0000	0.1000	0.2000	0.3500	0.4000	0.3667	0.3714	0.4250	0.4000	0.3800
Unet-ME	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Proposal-NoConcat	0.3600	0.3840	0.3680	0.3980	0.3952	0.4213	0.4251	0.4700	0.4791	0.4816
Proposal-Concat	0.0800	0.0800	0.1387	0.1760	0.1712	0.1813	0.1600	0.2190	0.2373	0.2680

TABLA B.30. Promedio de *Precision* para 25 búsquedas de nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0037	0.0087	0.0202	0.0276	0.0404	0.0499	0.0672	0.0759	0.1023	0.1287
Xception	0.0132	0.0334	0.0586	0.0796	0.0990	0.1270	0.1419	0.1687	0.1852	0.2219
Unet-ML	0.0000	0.0103	0.0309	0.0722	0.1031	0.1134	0.1340	0.1753	0.1856	0.1959
Unet-ME	0.0515	0.1031	0.1546	0.2062	0.2577	0.3093	0.3608	0.4124	0.4639	0.5155
Proposal-NoConcat	0.0186	0.0396	0.0569	0.0821	0.1019	0.1303	0.1534	0.1938	0.2223	0.2482
Proposal-Concat	0.0041	0.0082	0.0214	0.0363	0.0441	0.0561	0.0577	0.0903	0.1101	0.1381

TABLA B.31. Promedio de *Recall* para 25 búsquedas de nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.9440	0.9120	0.8800	0.8520	0.8608	0.8413	0.8171	0.7950	0.7716	0.7720
Xception	0.7920	0.7000	0.6640	0.6880	0.6480	0.6453	0.6274	0.6080	0.6071	0.6136
Unet-ML	1.0000	0.9000	0.8000	0.6500	0.6000	0.6333	0.6286	0.5750	0.6000	0.6200
Unet-ME	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Proposal-NoConcat	0.6000	0.6360	0.6107	0.6180	0.6192	0.5867	0.5623	0.5490	0.5360	0.5392
Proposal-Concat	0.9440	0.9120	0.8800	0.8520	0.8608	0.8413	0.8171	0.7950	0.7716	0.7720

TABLA B.32. Promedio de *Precision* para 25 búsquedas sin nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0530	0.1025	0.1483	0.1915	0.2418	0.2836	0.3213	0.3573	0.3901	0.4337
Xception	0.0445	0.0787	0.1119	0.1546	0.1820	0.2175	0.2467	0.2733	0.3070	0.3447
Unet-ML	0.0562	0.1011	0.1348	0.1461	0.1685	0.2135	0.2472	0.2584	0.3034	0.3483
Unet-ME	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Proposal-NoConcat	0.0337	0.0715	0.1029	0.1389	0.1739	0.1978	0.2211	0.2467	0.2710	0.3029
Proposal-Concat	0.0530	0.1025	0.1483	0.1915	0.2418	0.2836	0.3213	0.3573	0.3901	0.4337

TABLA B.33. Promedio de *Recall* para 25 búsquedas sin nódulos pulmonares con la inclusión del algoritmo Bridge para detectar near duplicates.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0533	0.1002	0.1416	0.1835	0.2210	0.2497	0.2758	0.3103	0.3360	0.3641
Xception	0.0570	0.1044	0.1479	0.1916	0.2299	0.2553	0.2820	0.3073	0.3413	0.3704
Unet-ML	0.0532	0.1003	0.1422	0.1798	0.2142	0.2464	0.2760	0.3029	0.3288	0.3520
Unet-ME	0.0490	0.0935	0.1368	0.1763	0.2118	0.2406	0.2707	0.3015	0.3207	0.3472
Proposal-NoConcat	0.0484	0.0959	0.1428	0.1743	0.2085	0.2448	0.2696	0.3011	0.3333	0.3432
Proposal-Concat	0.0526	0.1034	0.1450	0.1899	0.2146	0.2514	0.2828	0.3160	0.3440	0.3646

Tabla B.34. Promedio de F1 para 25 con búsquedas con nódulos pulmonares y 25 búsquedas sin nódulos pulmonares.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0565	0.0994	0.1486	0.1819	0.2236	0.2545	0.2836	0.3153	0.3380	0.3624
Xception	0.0596	0.1062	0.1457	0.1901	0.2302	0.2583	0.2915	0.3124	0.3425	0.3755
Unet-ML	0.0000	0.0187	0.0536	0.1026	0.1475	0.1732	0.1970	0.2336	0.2394	0.2721
Unet-ME	0.0980	0.1869	0.2029	0.2277	0.1948	0.2047	0.2521	0.2604	0.3014	0.3293
Proposal-NoConcat	0.0533	0.0979	0.1393	0.1723	0.2007	0.2400	0.2545	0.2832	0.3104	0.3189
Proposal-Concat	0.0541	0.1024	0.1507	0.1956	0.2144	0.2539	0.2855	0.3124	0.3403	0.3652

Tabla B.35. Promedio de F1 para 25 con búsquedas con nódulos pulmonares.

change	5	10	15	20	25	30	35	40	45	50
CeNet	0.0502	0.1010	0.1346	0.1850	0.2182	0.2447	0.2677	0.3051	0.3337	0.3655
Xception	0.0545	0.1026	0.1500	0.1930	0.2295	0.2521	0.2723	0.3020	0.3397	0.3649
Unet-ML	0.1064	0.1818	0.2308	0.2569	0.2807	0.3193	0.3548	0.3721	0.4179	0.4317
Unet-ME	0.0000	0.0000	0.0708	0.1248	0.2288	0.2763	0.2890	0.3423	0.3397	0.3649
Proposal-NoConcat	0.0434	0.0937	0.1462	0.1761	0.2161	0.2494	0.2845	0.3188	0.3558	0.3672
Proposal-Concat	0.0511	0.1042	0.1392	0.1842	0.2147	0.2487	0.2800	0.3194	0.3475	0.3637

Tabla B.36. Promedio de F1 para 25 sin búsquedas con nódulos pulmonares.