Segmentación del Ventrículo Izquierdo en Ecocardiogramas

Diego Sú Gómez, Estefanía Pérez Yeo, Vanessa Méndez Palacios, Francisco Javier Sánchez Panduro & Isaí Ambrocio

29 de noviembre de 2023

Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Guadalajara

Resumen Este estudio aborda la tarea de segmentación del ventrículo izquierdo del corazón en imágenes de ecocardiogramas en un plano apical, con el objetivo de evaluar la función cardíaca. El enfoque propuesto busca comparar la eficiencia de dos metodologías: el uso de máscaras y el uso de *landmarks*. La tarea principal consiste en comparar el rendimiento de ambas metodologías en función del tamaño del conjunto de entrenamiento. Para solucionar dicho reto se planea utilizar una arquitectura de segmentación, como *U-Net*, y realizar experimentos con diferentes metodologías, tanto empleando *máscaras* como el uso de *landmarks*. La resolución de este reto tiene potencial de mejorar significativamente la eficiencia y precisión de la segmentación del ventrículo izquierdo en ecocardiogramas, lo que podría tener un impacto positivo en la evaluación de la función cardíaca clínica.

Keywords: Segmentación · U-Net · Landmarks · Máscaras · Dice Score

1. Introducción

En el complejo mundo de la salud cardiovascular, donde cada latido cuenta y cada detalle puede ser la clave entre la vida y la muerte, la evaluación de la función cardíaca tiene un papel crucial en el diagnóstico y tratamiento de enfermedades cardiovasculares. Una herramienta que se ha vuelto fundamental para esta evaluación es la segmentación del ventrículo izquierdo del corazón en ecocardiogramas en un plano apical. Tradicionalmente, esta tarea se ha realizado de manera manual, lo que es laborioso y propenso a errores. Con el objetivo de mejorar y agilizar este proceso, este estudio se centra en la exploración de dos enfoques alternativos, que es la segmentación basada en máscaras y la segmentación basada en *landmarks*.

El enfoque de segmentación basada en máscaras involucra la utilización de redes neuronales profundas, como *U-Net*, para segmentar las imágenes, seguido de la evaluación del rendimiento mediante el *Dice Score*. Mientras que el enfoque alternativo propone el uso de *landmarks* y etiquetado probabilístico para generar máscaras detalladas que proporcionan información más precisa sobre la relación de cada píxel con los *landmarks*. Esta técnica tiene el potencial de mejorar la calidad de las imágenes segmentadas.

El objetivo principal de este estudio es comparar el rendimiento de ambas metodologías en función del tamaño del conjunto de entrenamiento. Se tiene la hipótesis de que el enfoque basado en *landmarks* será más eficaz con conjuntos de entrenamiento más pequeños, pero se desconoce cuál de las dos metodologías ofrecerá un mejor rendimiento a medida que el conjunto de datos crece.

Con esto se tiene como objetivo contribuir a una evaluación cardíaca más eficiente y precisa, lo que podría tener un impacto significativo en la atención clínica de pacientes con afecciones cardiovasculares.

2. Metodología

2.1. Estructura del Conjunto de Datos

El conjunto de datos utilizado para este estudio consiste en un conjunto de 10,030 videos de ecocardiogramas, cada video compuesto por una serie de *frames*, con un tamaño de 112x112 pixeles. Además, el conjunto se divide en otros tres conjuntos: entrenamiento, validación y prueba, esta división se realiza para poder evaluar el rendimiento del modelo en diferentes escenarios.

Adicionalmente, se tiene un archivo llamado *'FileList'* que contiene información relevante, como el nombre del archivo de video y la división a la que pertenece. Como también, el archivo *'VolumeTracings'* proporciona anotaciones detalladas para cada cuadro en términos de coordenadas (X1, Y1, X2, Y2) que definen la región del ventrículo izquierdo.

2.2. Protección de la Privacidad y Gestión de Datos

La inteligencia artificial destaca por su capacidad para manejar una gran cantidad de datos sensibles, desempeñando un papel crucial en la industria médica.

En el desarrollo de proyectos en este ámbito, es de gran importancia considerar minuciosamente la sensibilidad de los datos recopilados de diversos pacientes en intervalos de tiempo específicos. El conjunto de datos utilizado en este trabajo es obtenido de la Universidad de Medicina de Stanford, cuyo departamento de investigación *EchoNet-Dynamic* cuenta con un acuerdo de uso en el cual proporcionan un limite legal del uso del conjunto de datos. El listado de normas detalla los permisos tanto para la lectura como para el uso por parte de terceros, y se pueden encontrar en el enlace https://echonet.github.io/dynamic/ en la sección de *Accessing Dataset*. Esto verifica que los datos que se utilizarán durante el proyecto están anonimizados y no representan una amenaza al incumplimiento de las leyes de protección de los datos personales. Esto se ve reflejado en la naturaleza del conjunto de datos, ya que no se revela ningún tipo de información sensible.

De igual manera, se implementó un control de acceso para garantizar la confidencialidad y seguridad de los documentos que se desarrollaron. Todos los documentos generados para este trabajo, tanto reportes como códigos fuente, están designados con acceso restringido y solo los miembros del equipo tienen permiso para acceder a ellos. Este control de acceso garantiza que la documentación interna del proyecto este protegida de usuarios no autorizados a información confidencial. Además de eso, los repositorios donde se guardan los archivos del proyecto también están protegidos con acceso únicamente para los miembros del equipo, y si se quisiera hacer uso de alguno de estos códigos se deberá pedir acceso a ellos.

2.3. Método de Implementación

Para abordar la tarea de segmentación del ventrículo izquierdo en los ecocardiogramas, se emplea la arquitectura de red neuronal convolucional conocida como *U-Net*. Esta elección se basa en la capacidad demostrada de *U-Net* para tareas de segmentación de imágenes médicas.

La arquitectura *U-Net* se destaca por su capacidad para capturar características a distintos niveles de resolución mediante la combinación de bloques de codificación y decodificación. La sección de codificación sirve para extraer características relevantes de la imagen, mientras que la sección de decodificación facilita la reconstrucción precisa de las estructuras segmentadas.

En cuanto a la implementación del modelo, se utiliza la combinación de *frameworks* de *Deep Learning*, específicamente *TensorFlow* y *Keras*. Estos *frameworks* proporcionan herramientas robustas y flexibles para la construcción, entrenamiento y evaluación de modelos de aprendizaje profundo.

En el enfoque basado en máscaras, se utiliza un conjunto de datos para entrenar la red, permitiéndole predecir máscaras de segmentación binarias. Este método es efectivo para la identificación de regiones de interés en las imágenes. Por otro lado, en el enfoque de *landmarks*, se adopta un enfoque de etiquetado probabilístico mediante distribuciones gaussianas para representar la ubicación de *landmarks*. Esta estrategia posibilita la generación de máscaras más detalladas al tener en cuenta la información probabilística sobre la posición de puntos

de referencia específicos, mejorando así la precisión en la segmentación del ventrículo izquierdo en los ecocardiogramas.

2.4. Uso de Big Data

En lo que respecta al manejo de *Big Data*, el modelo de almacenamiento y la gestión de datos, se decide prescindir de tecnologías escalables por diversas razones.

En primer lugar, la naturaleza de los datos, presentados en forma de videos en formato '.avi', obligaría a emplear un lector de búfer que los convertiría a bits. Esto implicaría la realización de procesos adicionales para el manejo de los archivos, los cuales resultan innecesarios para abordar la problemática en cuestión.

Además, no se cuenta con archivos que demanden bases de datos relacionales, ya que no se trata de documentos ni tablas. Por lo tanto, el uso de tecnologías como *Cassandra o MongoDB* no resulta necesario en esta solución. Considerando también que el volumen de datos es relativamente pequeño y que se trata de un ETL (*Extract*, *Transform*, *Load*) sencillo, no tendría sentido emplear tecnologías que podrían introducir complejidad innecesaria.

Finalmente, en lo que respecta al entrenamiento de modelos de aprendizaje automático, el enfoque se centra en la extracción de características de las señales. Este tipo de procesamiento es relativamente simple y no requiere el uso de técnicas de procesamiento distribuido. Tomando en cuenta estas consideraciones, se opta por no utilizar tecnologías como *Hadoop*, *Spark*, *Cassandra* ni *MongoDB*, ya que no aportan beneficios sustanciales al reto.

Por las razones mencionadas anteriormente, se determinó utilizar únicamente el almacenamiento en la nube para guardar todos los archivos que sean necesarios para el reto, ya sean los videos del conjunto de datos, las imágenes procesadas almacenadas, y los modelos entrenados para poder tener acceso a ellos rápidamente y no requerir de unidades físicas para almacenar los materiales necesarios para desarrollar la solución. El proceso a seguir para poder llevar a cabo el proyecto se explicará a continuación, pero la idea principal fue tener todos los archivos que se iban a utilizar, además de los archivos generados, tanto los modelos y las máscaras, para poder acceder a todos y cada uno de ellos y que todos los miembros del equipo puedan utilizar todos los recursos sin dificultad alguna.

3. Experimentos

Después de haber explicado la metodología que se siguió para poder llevar a cabo la realización del proyecto, lo siguiente será detallar los pasos que se siguieron para poder obtener los resultados deseados. A lo largo de la implementación de la solución del proyecto se realizaron cinco actividades principales que permitieron obtener los resultados obtenidos.

A continuación se mostrarán cada una de las etapas, junto con una explicación a profundidad de los procedimientos que se realizaron y una justificación sobre la decisión de hacer cada uno de estos pasos.

3.1. Preprocesamiento del conjunto de datos

Lo primero que se realizó para poder llevar a cabo cada uno de los experimentos necesarios fue entender a fondo el conjunto de datos y su estructura. Como ya se mencionó previamente en el documento, el conjunto de datos contiene 10,030 videos de ecocardiogramas de corazones etiquetados. Cada video cuenta con dos *frames* seleccionados específicamente para entrenamiento de redes neuronales. Un *frame* es cuando el corazón se encuentra en su tamaño más grande, mientras que otro es cuando el corazón se encuentra en su punto más pequeño. La idea de este conjunto de datos es aislar el ventrículo izquierdo de cada uno de los *frames*, y los archivos contienen coordenadas de dicha máscara.

Adicional a estos videos, el conjunto de datos contiene dos archivos .csv, el primero el cual contiene la información de las máscaras para cada uno de los *frames*, mientras que el segundo contiene el nombre del archivo y el *split* al que pertenece ese video (*Train*, *Test*, *Validation*). Teniendo esto en mente, se decidió utilizar *Google Drive* para almacenar los archivos de video y poder tener acceso a ellos sin necesidad de precargarlos en el entorno virtual con el que se iba a trabajar. Se cargaron todos los archivos en una unidad compartida, permitiendo que todos los miembros del equipo tuvieran acceso a ellos.

Después de haber definido la manera de trabajar con los archivos que iban a ser utilizados, lo siguiente fue entender la relación entren los archivos .csv y los videos. Se observó que al cargar cada video se podía extraer el *frame* necesario, y en base a ese *frame* se pueden utilizar las coordenadas en el conjunto de datos para poder formar las máscaras de la imagen.

Con este conocimiento se generó una función la cual toma el archivo de video, extrae los *frames* en base al conjunto de datos, y utiliza las coordenadas de ese *frame* y video para formar las máscaras.

3.2. Estructura de los archivos a utilizar

Tras haber visto como funcionan las coordenadas en relación a los *frames* obtenidos, se generó una función para poder crear las máscaras de cada uno de los *frames* por video. Esta función actúa tal y como se explicó en la sección anterior, más se añadió la función de guardar tanto el *frame* como la máscara generada en distintas carpetas almacenadas en *Google Drive*. Las carpetas fueron organizadas en base al *split* al que pertenece cada uno de los videos, y la distribución quedó de la siguiente manera.

- Frames Entrenamiento
- Frames Prueba
- Frames Validación
- Máscaras Entrenamiento
- Máscaras Prueba
- Máscaras Validación

Los splits de los 10,030 videos se encuentran de esta forma:

- Entrenamiento: 7465
- Prueba: 1288
- Validación: 1277

Después de haber generado estas carpetas, se definió una función la cual itera a través de todos los videos del conjunto de datos, obtiene los *frames* en base al archivo .csv, genera las máscaras en base a las coordenadas y luego guarda tanto el *frame* como la máscara en sus respectivas carpetas dependiendo del *split* al que pertenecen. Tras haber terminado con este proceso, todos los *frames* y las máscaras se encuentran guardados en sus respectivas carpetas y se pueden acceder fácilmente sin necesidad de generarlas de nuevo.

3.3. Especificaciones acerca de las máscaras y los frames

Ya teniendo todas las máscaras y los *frames* organizados en sus respectivas carpetas, se pueden acceder de forma sencilla con el directorio de las carpetas de *Google Drive*. Para esto, se generó una función con el objetivo de seleccionar un directorio y un número específico de *frames*/máscaras, para poder entrenar un modelo posteriormente. Con esta función generada es muy simple el poder obtener una cantidad específica de *frames* y máscaras para poder usarlas en el entrenamiento de una red neuronal.

Esta función opera mediante el directorio especificado, se seleccionan los archivos del mismo, y luego se seleccionan únicamente los primeros **N** archivos de la carpeta. Después de haber realizado eso, se normalizan cada una de las imágenes y se guardan en un arreglo para su fácil acceso y manipulación.

Sin embargo, hay algunas cosas que se deben tomar en cuenta y procesos que hacer para que las máscaras estén listas para el entrenamiento. El proceso faltante es la normalización de las máscaras, el cual consiste en hacer que los valores sean únicamente 0 y 1, dejando una máscara binaria que permite que pueda ser utilizada en el proceso de entrenamiento. Esto se realiza fácilmente con una línea de código, pero es esencial para el entrenamiento exitoso de los modelos. Con esto realizado, los datos que se utilizarán para entrenar las redes neuronales son las siguientes.

- Arreglo de Frames (Entrenamiento/Prueba/Validación), de forma (N,112,112) en escala de grises, donde N es el número de frames por utilizar determinado en la función de carga de los frames.
- Arreglo de Máscaras (Entrenamiento/Prueba/Validación), de forma (N,112,112), en blanco y negro (0,1), donde N es el número de máscaras por utilizar determinado en la función de carga de los frames.

Para la realización de este proyecto, y en base a las limitaciones del entorno computacional que se utilizó, se decidió que los *splits* para la implementación de las redes neuronales fueran de la siguiente manera:

- Entrenamiento: 2000 videos del conjunto de entrenamiento

- Prueba: 1000 videos del conjunto de prueba
- Validación: 1000 videos del conjunto de validación

Con esto ya definido, lo siguiente a realizar fue realizar la red neuronal, definir su arquitectura, funciones de pérdida, optimización y las métricas utilizadas para monitorear el entrenamiento.

3.4. Implementación de la U-Net en base a máscaras

El siguiente proceso realizado fue generar la red neuronal para poder realizar predicciones. El primer método que se utilizó fue el de realizar las predicciones en base a las máscaras obtenidas, enviando como entradas los *frames* con sus respectivas máscaras, y buscando que la salida de la red neuronal sean máscaras similares sin necesidad de aplicar ningún postprocesamiento a los resultados.

La arquitectura elegida fue la *U-Net*. La *U-Net* es una arquitectura de redes neuronales utilizadas para segmentación de imágenes. Está compuesta por dos partes principales: contracción y expansión. Es una red convolucional que va reduciendo la imagen mientras la estira horizontalmente y viceversa. Con esto se busca encontrar relaciones entre los valores de entrada y poder empezar a definir criterios para las predicciones.

Esta red fue entrenada con los *frames* y máscaras de entrenamiento, utilizando el optimizador *Adam*, mientras que la función de pérdida fue tanto la *Binary Crossentropy* y el *Mean Squared Error*, estas dos métricas fueron útiles para obtener los resultados más acertados con las máscaras obtenidas. En cuanto a las métricas utilizadas respecto al rendimiento de los resultados, se usaron dos métricas diferentes: *Accuracy* y *Dice Score*.

La métrica *Accuracy* se refiere a la fracción de predicciones que son verdaderas. Esta métrica mide la precisión del modelo al evaluar la proporción de predicciones correctas en relación con el conjunto total de instancias. Básicamente nos ayuda a tener una visión más general del éxito del modelo al momento de hacer las clasificaciones.

La segunda métrica hace mención en cuanto a la mejora de segmentación de las imágenes dentro de la red, donde se valida que tan similares son las predicciones obtenidas de las máscaras establecidas. Esta similitud es conocida como el tamaño del *overlap* de las dos segmentaciones entre el tamaño total de ambas imágenes; dentro del contexto de ecocardiogramas, se conoce que tomando en cuenta las máscaras del ventrículo se observan imágenes compuestas entre pixeles de 1 y 0 ó blanco y negro respectivamente.

En base a esta métrica el *Dice Score* es conocido como:

$$F_1 = \frac{2 \cdot numero\,de\,verdaderos\,positivos}{2 \cdot numero\,de\,verdaderos\,positivos + numero\,de\,falsos\,positivos + numero\,de\,falsos\,negativos}$$

Tomando en cuenta la perspectiva de segmentación de imágenes, esta ecuación puede ser traducida a una posición donde **A** es la máscara establecida y **B** es la máscara predicha por la red; los positivos son conocidos como el número total de pixeles en 1 (blanco) dentro de **A**, los verdaderos positivos son el número

total de pixeles en 1 tanto en **A** y **B**, y por último aquellos falsos negativos son el número total de pixeles que se muestran en 1 en **B**, pero 0 en **A**.

$$Dice\,Score = \frac{2 \cdot |A \cap B|}{2 \cdot |A \cap B| + |B \setminus A| + |A \setminus B|} = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

En cuanto a los valores de pérdida se toma en cuenta que en la metodología de máscaras utiliza la métrica de *Binary Crossentropy* de la librería de *Keras*, la cual calcula a como su nombre los menciona, la pérdida *cross-entropy* entre la máscara verdadera y su correspondiente predicción, donde *cross-entropy* hace referencia a una pérdida logarítmica donde se mide el rendimiento del modelo cuyo output es una probabilidad de entre 0 y 1.

La arquitectura de la U-Net seleccionada fue la siguiente:

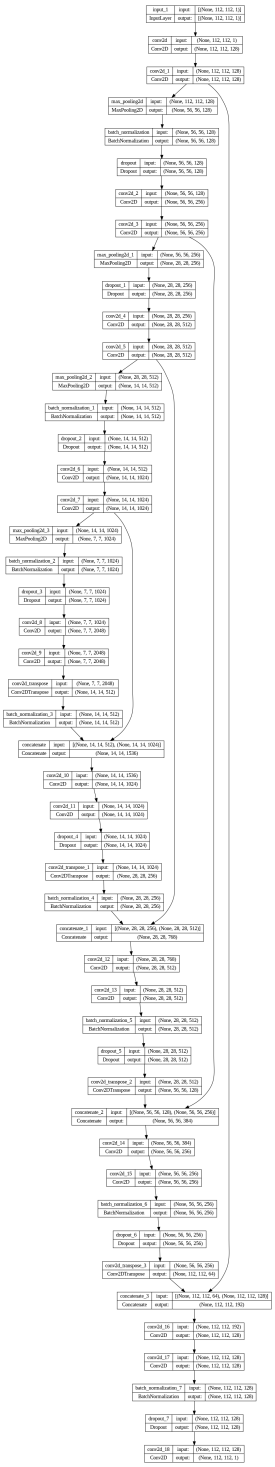

Figura 1. Arquitectura de la U-Net entrenada mediante máscaras.

Como se puede observar, la primera capa definida es la capa de entrada, la cual recibe entradas con forma de (112,112), que es el tamaño de las imágenes tanto de los *frames* como de las máscaras. Después de haber recibido las entradas, se encuentran tres capas diferentes para cada una de las convoluciones realizadas. Cada capa de convolución tiene una convolución en dos dimensiones, con el número de neuronas variando dependiendo del número de convolución, empezando por 128 neuronas.

Además, cada capa también tiene un *MaxPooling*, *BatchNormalization* para evitar *overfitting*, y finalmente un *dropout*, para eliminar aleatoriamente cierto porcentaje de las neuronas y mejorar el entendimiento de las entradas y evitar el riesgo de la memorización de la red. En total, se generaron 4 capas de convolución con número variante de neuronas, además también de alterar el porcentaje de neuronas eliminadas en el *dropout*. Estas convoluciones forman parte de la fase de contracción de las imágenes, reduciendo su tamaño y encontrando relaciones entre los *frames* y máscaras.

Después de haber concluido con la etapa de contracción, lo siguiente es la fase conocida como el cuello de botella, donde se reduce al máximo el tamaño de las entradas y de ahí prosigue la etapa de expansión. La etapa de cuello de botella incluye también dos capas de convolución adicionales. En esta etapa de expansión se utilizan convoluciones transpuestas, también conocidas como des-convoluciones, debido a que esta vez se van aumentando las dimensiones de las entradas. Estas capas utilizan dos argumentos importantes, una función de activación la cual permite definir el proceso de interpretación de las neuronas. La función para todas las capas de la red es *relu"*, mientras el segundo argumento es el de los márgenes, el cual se declara como *"same"*, lo que indica que no hay desplazamiento entre los *frames* y las máscaras.

Finalmente, la última capa es la capa de salida, en donde se define la estructura de las salidas del modelo, que en este caso son imágenes de tamaño (112,112) compuesta por dos clases distintas, las cuales son valores compuestos por 1 o 0, indicando así la realización de las máscaras.

Por último, cabe aclarar que todas las capas a excepción de la de entrada y salida tienen *Batch Normalization* y un *dropout*.

3.5. Implementación de la U-Net en base a landmarks

Después de haber entrenado la *U-Net* en base a las máscaras obtenidas, se decidió tomar otro enfoque distinto. Además de haber formado las máscaras, lo siguiente que se hizo fue usar las máscaras para generar *landmarks*, que constan de puntos clave alrededor de cada una de las máscaras, para generar ocho *landmarks* alrededor de la silueta de cada máscara, y posterior a eso se aplica una dilatación y un *blur* gaussiano para incrementar la tolerancia en los resultados obtenidos. Finalmente, también se normalizan los resultados, para que el valor más alto en cada canal sea de 1, y el resto sean menores, generando un mapa de probabilidad con los valores de cada uno de los canales, facilitando el proceso de entrenamiento del modelo. Con estos *landmarks* definidos, se generaron imágenes de ocho canales, en donde cada canal consta de un punto específico, y

se entrenará la misma arquitectura con la intención de que al entrenar se genere cada *landmark* en cada uno de los canales, para después formar la máscara resultante.

Para este enfoque, la función de pérdida utilizada fue el *Mean Squared Error*, y no se utilizaron métricas adicionales para determinar la calidad de los resultados.

La arquitectura para esta *U-Net* es la siguiente:

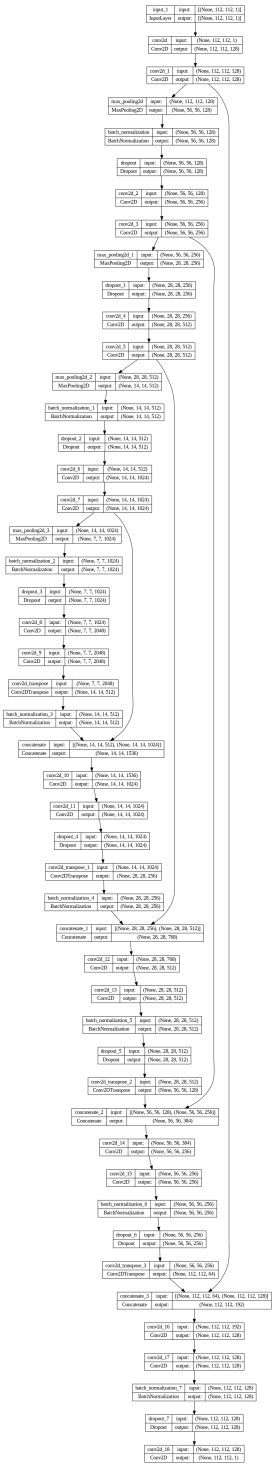

Figura 2. Arquitectura de la U-Net entrenada mediante landmarks.

Cómo se puede observar en esta imagen, la arquitectura de la *U-Net* es prácticamente idéntica a la utilizada para la predicción de máscaras, con únicamente tres diferencias. La primera de ellas es la capa de salida. Como se observa, la capa de salida contiene ocho neuronas en lugar de una. Esto se debe a que, como en este caso se quieren predecir cada uno de los *landmarks* por separado, la salida debe ser un punto por cada canal, y al tener ocho canales diferentes la idea es generar ocho puntos distintos.

La segunda diferencia es la utilización del *UpSampling* en lugar del *MaxPooling*. Esto también es a causa de que la salida se quiere utilizar en ocho canales diferentes en lugar de uno solo. Y por último, la otra diferencia es la función de activación en la capa de salida. En este caso, se utiliza una función sigmoide, la cual genera que las salidas se encuentren en valores entre 0 y 1, y estos valores representan la probabilidad de que el punto en ese canal sea el *landmark* en sí.

Tomando en cuenta los valores de pérdida, se decidió utilizar Mean Squared Error de la librería de *Keras* como métrica dentro de esta metodología, la cual sigue la fórmula *loss = mean(square(ytrue - ypred), axis=-1)*, donde se obtiene un output en forma al *batch size* correspondiente, lo cual para nuestra implementación es de 63.

Tras haber explicado la arquitectura de ambas redes neuronales y el proceso que se llevó a cabo para haber llegado hasta este punto, lo siguiente por realizar es evaluar los resultados obtenidos para cada uno de los métodos, compararlos y encontrar conclusiones en base a lo que se haya obtenido.

4. Resultados

Previo a mostrar los resultados obtenidos, se debe explicar cómo se realizó el proceso de entrenamiento. Para ambas metodologías el proceso de entrenamiento fue el mismo, con la única variación que fue el utilizar las métricas adicionales para las máscaras.

El entrenamiento se realizó durante 100 épocas, con el tamaño del *batch* siendo de 63 imágenes seleccionadas al azar tanto del conjunto de entrenamiento como el de validación. Además de esto, se declaró un *checkpoint*, el cual permite guardar el modelo óptimo en base a un criterio en específico. En este caso, el criterio fue la pérdida obtenida durante la validación, y se guardará el modelo con el menor valor en esta métrica.

Cada época funciona de manera diferente. Lo primero que se hace es seleccionar las imágenes que conforman el *batch*. En este caso, se seleccionan 63 *frames* y sus respectivas máscaras del conjunto de entrenamiento. Después se introducen a la red neuronal y se van calculando las funciones de pérdida y las métricas adicionales, si es que hay. Después, con el modelo entrenado con ese *batch*, se evalúa en un *batch* del mismo tamaño pero con el conjunto de validación, y se calculan sus respectivas métricas, para después pasar a la siguiente época e ir cambiando los pesos dependiendo de los resultados.

Después de haber concluido con la etapa de entrenamiento, se pueden realizar distintos gráficos para ir observando los cambios en las métricas a través de las épocas, además de poder observar los resultados obtenidos con los actuales.

4.1. U-Net en base a máscaras

Después de haber entrenado el modelo mediante máscaras, se descargó el modelo final almacenado y se probó en el conjunto de prueba. Después de haber guardado las predicciones, se generó una tabla en donde se hicieron gráficas tanto del *frame* original, como de la máscara generada con las coordenadas, y de la máscara generada por el modelo. Los resultados se muestran a continuación.

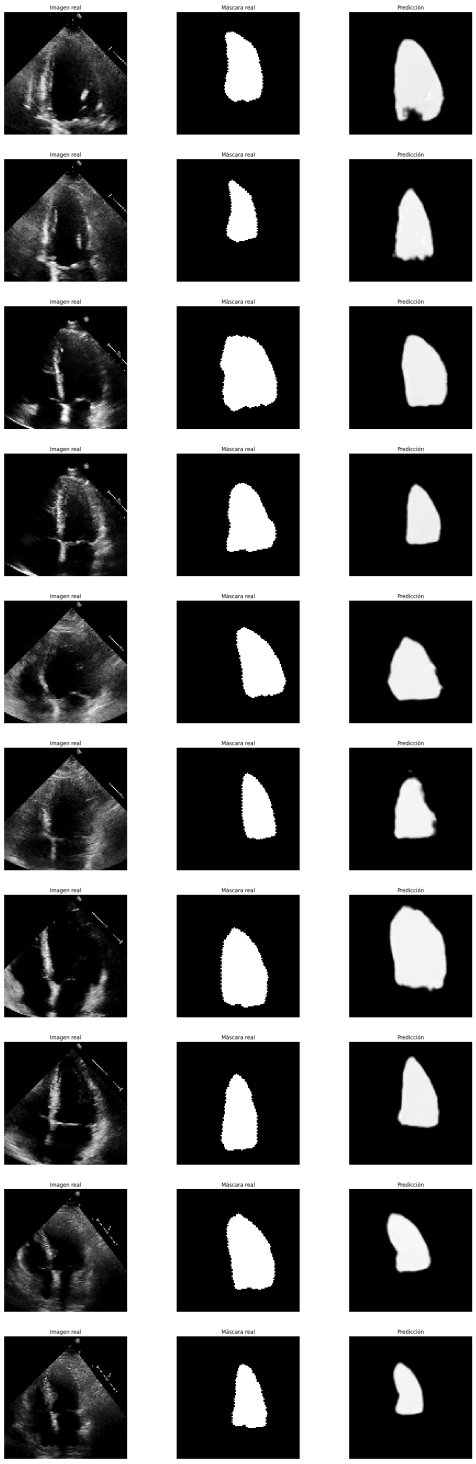

Figura 3. Predicciones de la U-Net entrenada mediante máscaras.

Como se puede observar en los resultados, la gran mayoría de las máscaras generadas corresponden con las originales, aunque algunos de los resultados tienen un poco de ruido adicional. Sin embargo, estos resultados son capaces de identificar correctamente el ventrículo izquierdo en los *frames* con una gran precisión en los resultados. Se pueden observar claramente las curvas y los puntos que posiblemente eran difíciles de identificar, tales como los que se encuentran entre otras partes del corazón. Sin embargo, se puede ver claramente que las formas se distinguen y son muy similares a las originales, por lo que se puede decir que el modelo es capaz de predecir correctamente los resultados.

Además de eso, las métricas obtenidas a lo largo del entrenamiento se muestran a continuación.

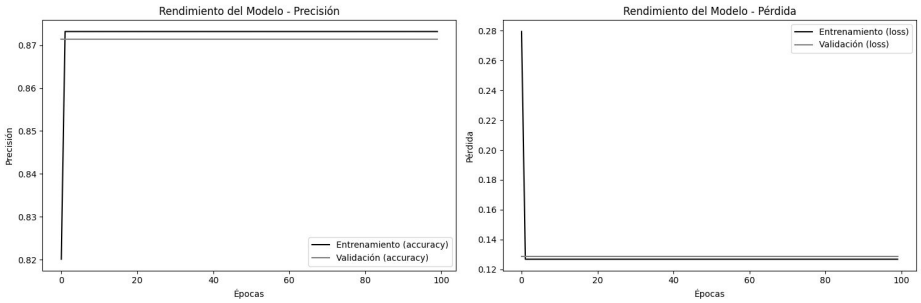

Figura 4. Métricas capturadas durante el entrenamiento de la U-Net en base a máscaras.

Estas métricas ya se explicaron en las secciones anteriores, por lo que se pueden interpretar para entender a profundidad cómo se llevó a cabo el entrenamiento de esta red. Como se puede observar, a lo largo de las épocas de entrenamiento la función de pérdida disminuyó exponencialmente en las primeras 5 épocas, y después estuvo prácticamente constante, disminuyendo en cantidades muy pequeñas. Sin embargo, la pérdida durante la validación fue siempre pequeña, oscilando entre 0.12 y 0.14 e igual sufriendo muy pocos cambios a lo largo del entrenamiento. Por otro lado, la precisión en el entrenamiento de este modelo arrancó aproximadamente en 0.82, incrementó exponencialmente durante las primeras 5 épocas, y luego se mantuvo constante durante el resto de las épocas, incrementando en magnitudes muy pequeñas. Esto se puede decir igualmente de la precisión durante la validación, empezando en 0.88 aproximadamente y manteniéndose en ese rango a lo largo del entrenamiento.

4.2. U-Net en base a landmarks

Al igual que con el enfoque de máscaras, después de haber entrenado el modelo, se descargó el óptimo y se realizaron predicciones en base al conjunto de prueba. De igual manera, se generó la gráfica con el *frame* original, la máscara

generada por las coordenadas, y la generada por el modelo. Sin embargo, debido a la estructura de la red neuronal, donde la salida es una imagen de forma (112,112,8), se tiene que aplicar un postprocesamiento a los resultados antes de poder obtener las máscaras.

Este postprocesamiento consiste en tomar las predicciones obtenidas, y seleccionar el pixel con el valor más grande, lo cual se puede interpretar como el pixel con la mayor probabilidad de ser el *landmark* seleccionado. Se selecciona este pixel para cada uno de los canales, luego se unen en una imagen de forma (112,112), y se genera un polígono en base a los puntos seleccionados.

Después de aplicar este procesamiento, los resultados quedan de la siguiente manera.

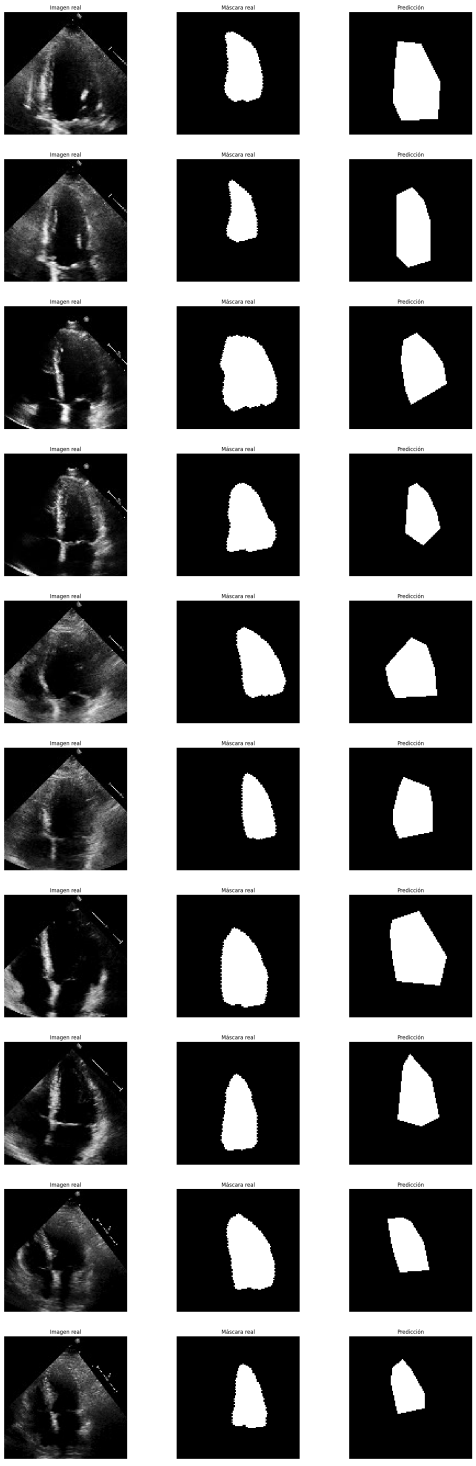

Figura 5. Predicciones de la U-Net entrenada mediante landmarks.

Los resultados obtenidos, como se puede ver, tienen distintas precisiones. La primer diferencia que se observa es la incapacidad del modelo de identificar y generar curvas. Los resultados de los puntos son bastante acertados, sin embargo, la formación de los polígonos es bastante limitada al entender la forma de las máscaras. Aún así, la forma general es bastante adecuada, y se podría decir que los resultados son aceptables más están lejanos a ser considerados buenos. Algunas formas en las que estos resultados se pueden mejorar sería incrementando el número de *landmarks* a utilizar, o definir criterios más estrictos para la selección de los *landmarks*. Además, también se podría crear una función mucho más específica para la generación de polígonos, en lugar de usar una función tan trivial como la que se utilizó.

Los parámetros del entrenamiento de este modelo se observan aquí.

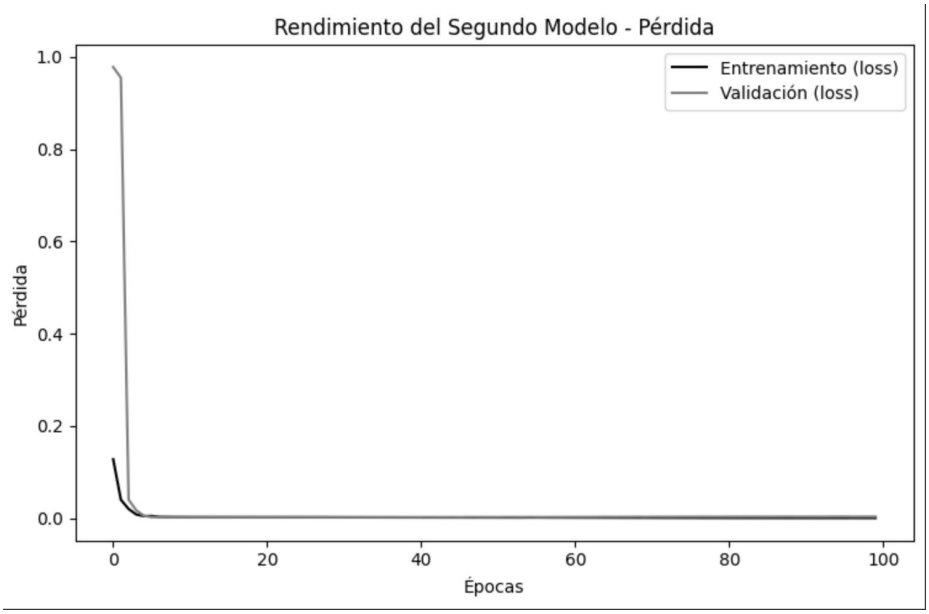

Figura 6. Pérdida capturada en el entrenamiento de la U-Net en base a landmarks.

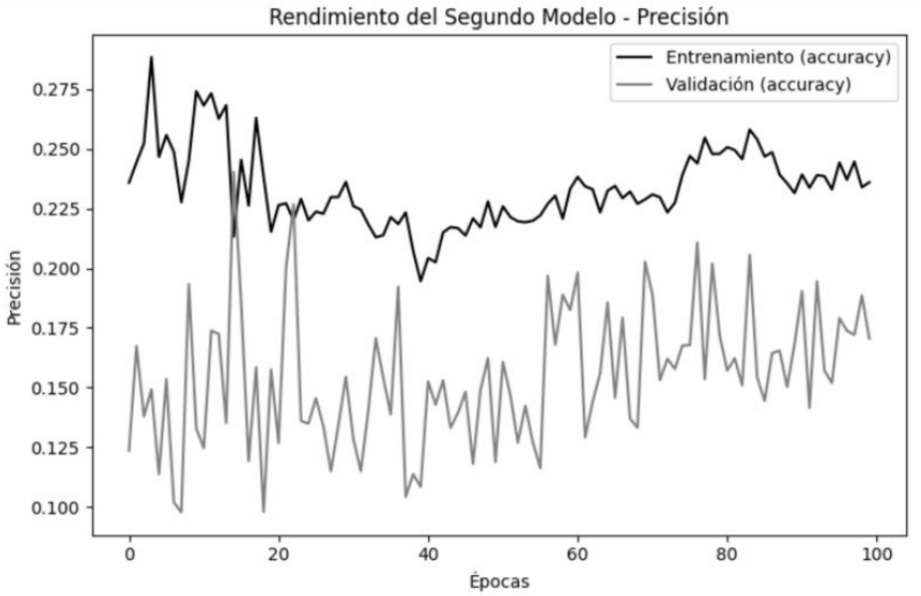

Figura 7. Precisión capturada en el entrenamiento de la U-Net en base a landmarks.

En esta gráfica, se observa que la función de pérdida durante el entrenamiento no sufrió tantas modificaciones a lo largo de las épocas de entrenamiento. la función de pérdida empezó en 0.2 aproximadamente, y redujo hasta números muy cercanos a 0, y se mantuvo oscilando por ese rango el resto de las épocas de entrenamiento. Esto se puede deber a que se está entrenando en base a *landmarks*, y la mayoría de la imagen resultante estará en color negro (0), o en valores muy cercanos a él. Por otro lado, la precisión fue variando significativamente a lo largo de las épocas, tanto en el entrenamiento como la validación. Los valores oscilaban entre 0.125 y 0.275, cambiando drásticamente durante las épocas. Se puede ver que los valores de entrenamiento fueron superiores a los de validación. Sin embargo, esta métrica no es tan reveladora, teniendo en cuenta que cada canal de las imágenes únicamente va a revelar un punto en específico, los resultados de cada uno de los canales serían líneas de probabilidad circulando el punto ideal para ser parte de la máscara, por lo que no es nada alarmante.

4.3. Comparación de los resultados obtenidos

Para poder realizar una comparación efectiva entre los resultados generados por el modelo de *landmarks* con el de máscaras, se decidió utilizar una función de la librería *Sci-Kit Learn*, la cual se llama *Structural Similarity Index*, o por sus siglas SSI, la cual consiste en una función que evalúa la estructura de ambas imágenes y encuentra un índice de similiaridad, el cual puede estar entre 0 y 1. Esto permitirá evaluar las máscaras generadas por cada uno de los modelos,

y definirá un valor de similaridad para cada una de ellas, permitiendo generar información descriptiva sobre la efectividad de los modelos en comparación con los resultados reales, sin necesidad de utilizar la subjetividad de la percepción de un ojo humano. Los resultados obtenidos son los siguientes.

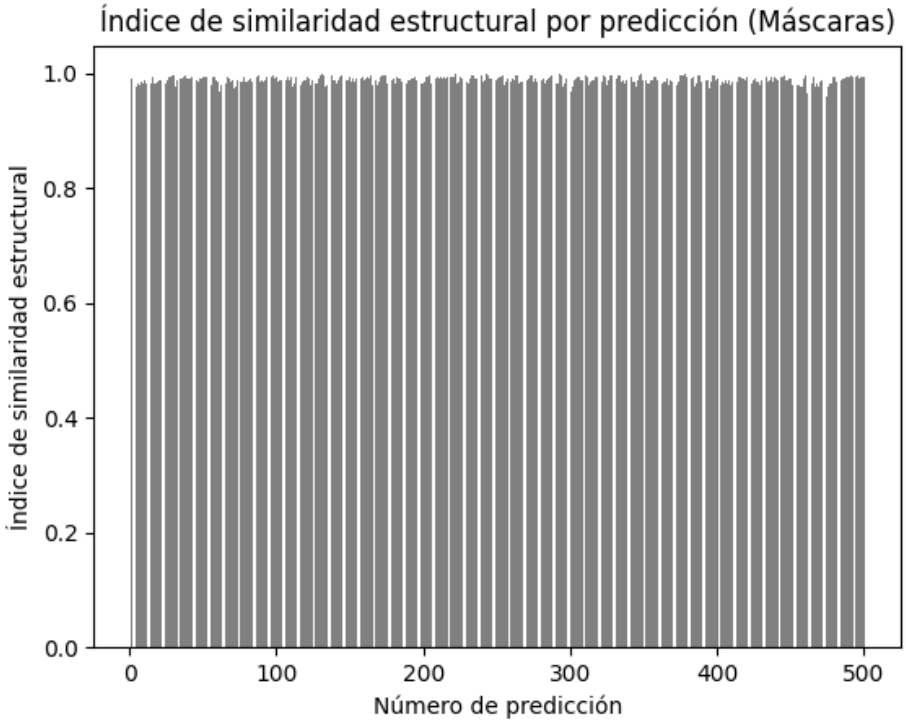

Figura 8. SSI del modelo entrenado por máscaras

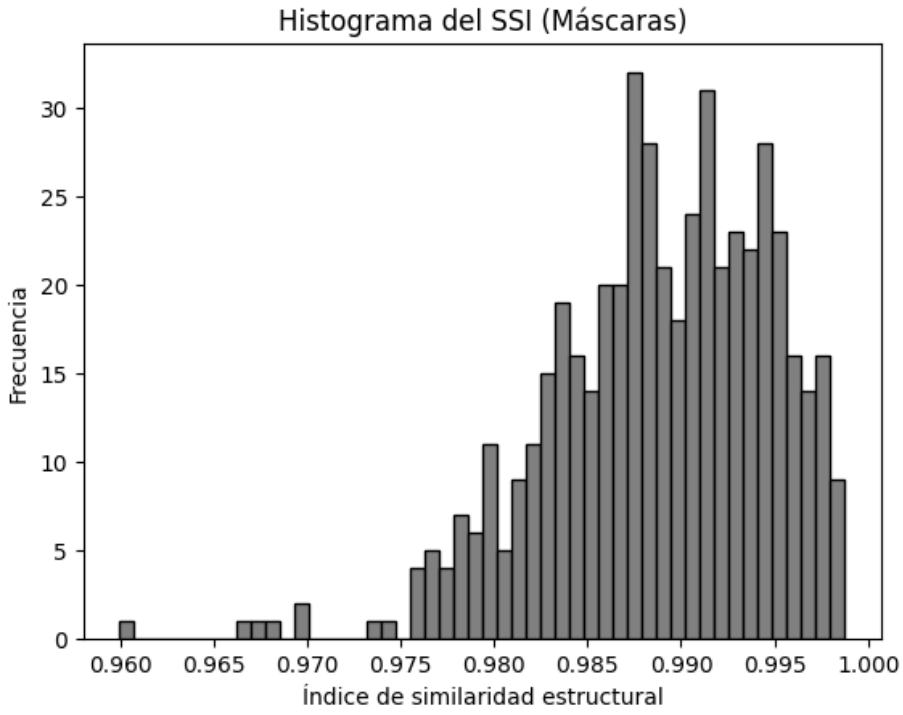

Figura 9. Histograma del SSI del modelo entrenado por máscaras

Como se puede observar en estas dos gráficas, las medidas obtenidas por el índice de similaridad estructural en las predicciones generadas por el modelo en base a máscaras son bastante buenas, ya que se puede ver que la gran mayoría de las predicciones están entre el 0.8 y el 1, es decir entre un 80 y 100 por ciento de similaridad. Esto se comprueba en el histograma respectivo, donde se observa que la gran mayoría de las predicciones están entre 80 y 90 por ciento de similaridad. Esto comprueba que este modelo es bastante bueno al predecir máscaras mediante un *frame* de un ecocardiograma.

Por otro lado, los resultados del modelo entrenado mediante *landmarks* se observan a continuación.

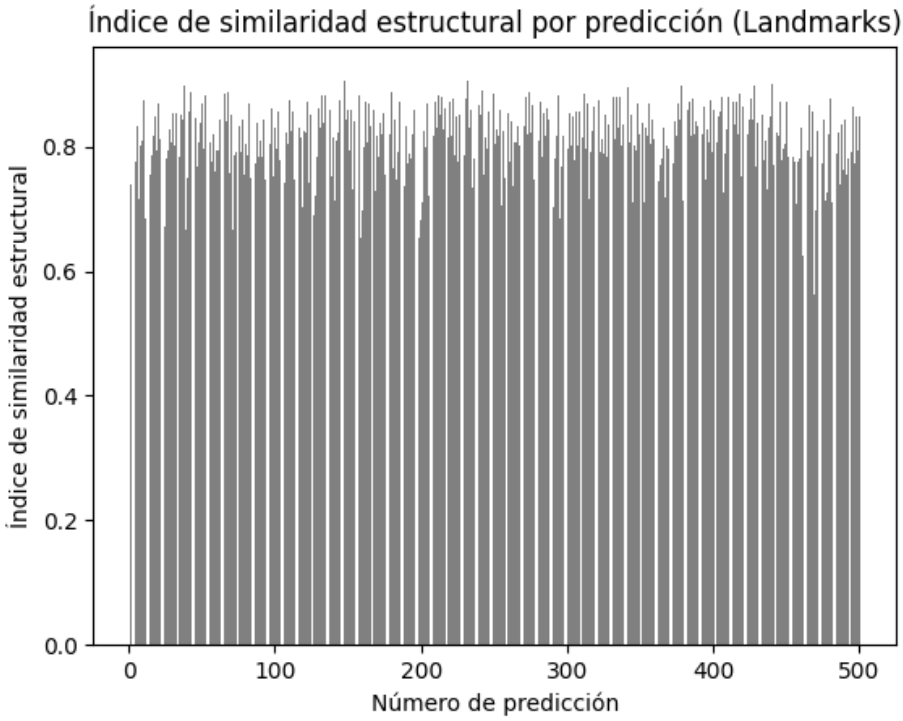

Figura 10. SSI del modelo entrenado por landmarks

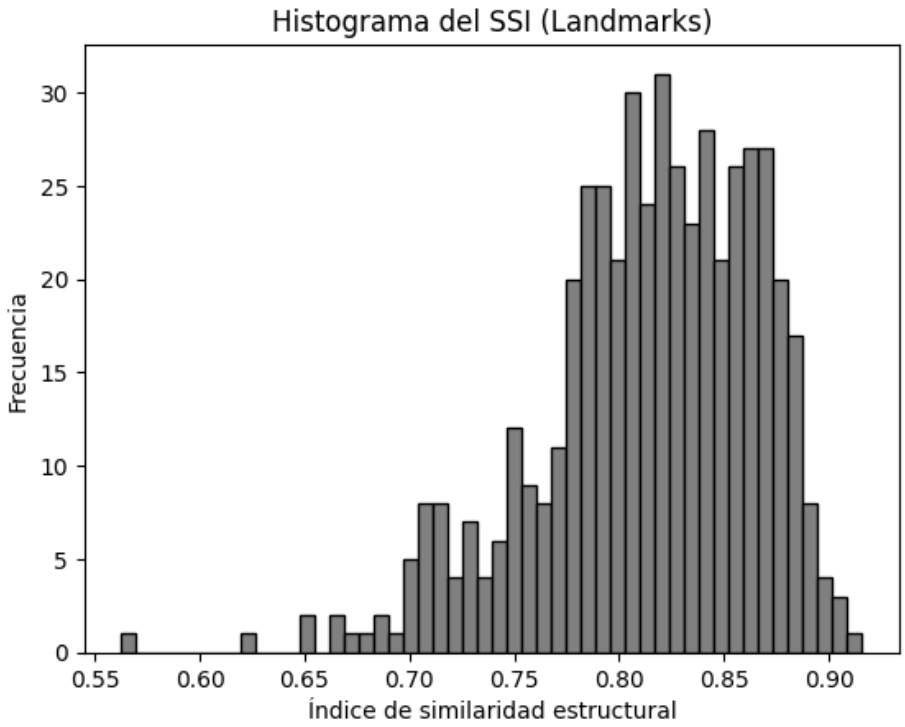

Figura 11. Histograma del SSI del modelo entrenado por máscaras

Como se ve en los gráficos, el SSI en este caso es mucho más variante, oscilando entre el 0.7 y el 0.9, teniendo muchas menos predicciones en valores superiores al 80 %. Se puede ver que el modelo funciona un poco peor que el entrenado por las máscaras, y esto es respaldado por su respectivo histograma, donde se observa que la mayoría de los valores oscilan entre el 80 y el 85 % de similaridad. Si bien es cierto que las diferencias en cuanto a los valores no son notorias, si se observan ambos tipos de máscaras generadas, se puede ver que las generadas por el modelo de *landmarks* no distinguen curvaturas significativas, lo que hace que su efectividad se vea comprometida. En conclusión, se puede decir que tanto en base a los resultados visuales, como en las similitudes con las máscaras reales, el modelo entrenado en base a máscaras es mucho más efectivo que el de los *landmarks*.

5. Conclusiones

5.1. Diego Sú Gómez

Tras haber finalizado el proceso de desarrollo y solución del proyecto, puedo concluir que el modelo que mejor resultados tuvo fue el de las máscaras. Los

resultados fueron mucho más precisos que lo esperado, además de que el modelo se desenvolvió bastante bien con el conjunto de prueba, y si se mejora mediante los hiper parámetros o algún método de retroalimentación podría llegar a ser bastante competitivo en los resultados obtenidos. Sin embargo, la desventaja principal de este modelo es la generación de máscaras. En este escenario se contaban con las máscaras ya generadas, mientras que en un caso general o más realista, se deberían generar manualmente las máscaras, lo que sería un proceso más tedioso y podría llevar mucho más tiempo. Sin embargo, entrenar este modelo con todo el conjunto de entrenamiento podría hacer que el modelo sea lo suficientemente competitivo para tener resultados muy buenos, con la única desventaja de que el tiempo de entrenamiento podría ser bastante tardado, pero el resultado sería muy bueno.

Si bien es cierto que el modelo en base a landmarks fue menos efectivo, esta metodología de implementación tiene un mayor potencial en un escenario más realista, debido a que es mucho más fácil seleccionar puntos que generar máscaras por completo. Si se pudiera mejorar el algoritmo de formación de máscaras en base a puntos, o encontrar un mejor criterio para determinar los puntos seleccionados, el potencial de ese modelo es mucho mayor. Al igual que con el modelo en base a máscaras, entrenar este con todo el conjunto de entrenamiento podría mejorar significativamente los resultados, pero con las limitaciones del entorno virtual con el que se trabajó, esto no fue posible, pero el hacerlo haría que el modelo mejore significativamente y poder obtener resultados mucho más precisos y adecuados con la situación.

En conclusión, se puede decir que el modelo con mejores resultados fue el de las máscaras, pero el de los *landmarks* puede ser mucho más efectivo si se perfecciona más a detalle.

5.2. Estefanía Pérez Yeo

Tomando en cuenta el auge de hoy en día en cuanto al uso de Redes Neuronales se conoce que estas herramientas ingresan a las diferentes industrias con el fin de aplicar mejoras. Para la ocasión presentada a inicios de la materia, la industria médica suele requerir el volumen del ventrículo izquierdo del corazón, donde por cuestiones de salud se toma en base el ecocardiograma en el momento sístole y diástole, he aquí donde con ayuda de estas imágenes decidimos utilizar una arquitectura *U-Net*.

Como primer acercamiento se manejó el uso de máscaras en base a los *frames* entregados, aquí dentro de la arquitectura manejada se observó que el uso forzoso de *flatten* no es necesario dentro de nuestras convoluciones, ya que, las dimensiones tanto de *input* como de *output* se mantienen por igual en 112x112x1. En base a nuestra red *CNN* diseñada obtuvimos resultados altos hablando desde un perspectiva subjetiva tanto en *Accuracy* como en *Dice Score*.

Para la estrategia de *landmarks* decidimos utilizar una técnica de ordenamiento en sentido *clockwise* de los puntos obtenidos por el conjunto de datos proporcionado a inicios del reto, en base a ello se decidió tomar 8 puntos claves

y colocados estratégicamente dentro de las curvaturas principales, cabe mencionar que estos 8 puntos fue a base de recomendación de nuestros docentes ya que 7 también podía ser otra opción, ya que como mínimo se observa que se sigue una tendencia en las posiciones de los *landmarks* en cuanto a donde se colocan al rededor del contorno. También siguiendo una *U-Net* como arquitectura dentro de la red de los *landmarks* y dimensiones de 112x112x8 canales, los resultados a pesar de ser buenos no llegaron a superar las métricas del método de máscaras, esto puede ser a que el criterio de selección de puntos no respeta la curvatura natural de la máscara.

5.3. Francisco Javier Sánchez Panduro

Después de revisar todo lo que hemos hecho en nuestro proyecto de segmentación del ventrículo izquierdo del corazón, creo que hemos aprendido mucho de ambos métodos que probamos: el de las máscaras y el de los *landmarks*.

Con el método de las máscaras, los resultados fueron impresionantes en precisión. Fue capaz de identificar muy bien el ventrículo izquierdo. Pero, hay un pero: hacer estas máscaras a mano es complicado y consume mucho tiempo.

Por otro lado, el método de los *landmarks*, aunque no fue tan preciso, resultó ser más práctico. Es más fácil marcar puntos que crear máscaras desde cero. Si podemos mejorar cómo seleccionamos y usamos estos puntos, este método podría ser realmente útil.

5.4. Isai Ambrocio

La inteligencia artificial en los últimos años ha sido una tecnología que cada vez ha tomado más relevancia, ya que facilita tareas y procesos, debido al gran alcance que tiene en diferentes áreas.

A lo largo del reto pude observar un poco de las aplicaciones que tiene la inteligencia artificial, como lo es el poder analizar datos, hacer segmentaciones, hacer recomendaciones, predicciones, aplicaciones para mejorar la educación y la medicina; la cual, esta última es de suma importancia y es en el rubro el cual nos enfocamos.

Dicho lo anterior, nosotros creamos dos modelos, los cuales puedan aportar un apoyo significativo a los especialistas en cuanto a evaluación respecta. Los modelos se centran en la segmentación de imágenes identificando el ventrículo izquierdo del corazón. Ambos modelos utilizan redes neuronales *U-Net*; sin embargo, cada modelo tiene un acercamiento diferente. Uno de los modelos está enfocado en máscaras, mientras que el otro utiliza *landmarks*.

Ambos modelos tuvieron un desempeño que superó nuestras expectativas; no obstante, cada uno tuvo sus ventajas, pero también sus desventajas. En el caso del modelo de máscaras dio mejores resultados. Por otra parte, el tiempo de entrenamiento fue elevado y a su vez, tuvo mayor demanda de recursos computacionales. En el caso de los *landmarks*, fue lo opuesto al modelo de las máscaras.

Con lo antes mencionado podemos concluir que ambos modelos son confiables y en posibles actualizaciones podrían ser aún mejor, tratando de mejorar las limitaciones actuales y con ello poder ser una herramienta fundamental para el apoyo de diagnósticos y evaluaciones de los especialistas en dicha rama de la salud.

5.5. Vanessa Méndez Palacios

En este trabajo de segmentación del ventrículo izquierdo del corazón, el cual fue realizado con dos enfoques diferentes, se obtuvieron resultados destacados, cada uno con sus propias ventajas y limitaciones.

El enfoque de segmentación de máscaras obtuvo muy buenos resultados al identificar con precisión el ventrículo izquierdo en los ecocardiogramas. Esta metodología se caracterizó por tener una muy buena capacidad para capturar detalles y generar máscaras con formas muy similares a las originales. Sin embargo, se observaron ciertos casos de ruido adicional en algunas máscaras, lo que podría ser objeto de ajustes futuros.

En contraste, el enfoque de segmentación basado en *landmarks*, aunque obtuvo buenos resultados, presentó limitaciones evidentes en la capacidad para identificar y generar curvas de manera precisa, lo cual indica la necesidad de mejoras en la comprensión de la forma de las máscaras, como lo puede ser la incorporación de criterios más estrictos en cuanto a la selección del número de *landmarks* para conseguir así la mejora de precisión.

Se puede concluir que este trabajo nos ha proporcionado una valiosa comparación entre dos enfoques para la segmentación, con ayuda de modelos de redes neuronales. De alguna manera, los resultados que se obtuvieron contribuyen al avance en la comprensión de las capacidades y limitaciones de cada metodología, lo cual puede funcionar como base para futuras investigaciones destinadas a perfeccionar la precisión y eficiencia en la evaluación de la función cardíaca a través de técnicas de procesamiento de imágenes y *Deep Learning*.

Bibliografía

[1] Lever, J.: Classification evaluation: it is important to understand both what a classification metric expresses and what it hides. Nature Methods **13**(8), 603 (2016)

[2] Stack Exchange, https://stats.stackexchange.com/questions/195006/is-the-dice-coefficient-the-same-as-Accuracy. Last accessed November 27, 2023