



**UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



**COMPARACIÓN DE ALGORITMOS DETECTORES DE LATIDOS EN SEÑALES BCG CON  
Y SIN INCORPORACIÓN DE RED NEURONAL CONVOLUCIONAL**

POR

**Miguel Angel Sepúlveda González**

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción  
para optar al título profesional de Ingeniero Civil Biomédico

Profesor Guía  
Esteban Pino Quiroga

Agosto 2023  
Concepción (Chile)

© 2023 Miguel Sepúlveda González

© 2023 Miguel Ángel Sepúlveda González

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

## Agradecimientos

En el camino hacia la culminación de esta memoria de título, me siento profundamente agradecido por el apoyo inquebrantable y las contribuciones significativas de diversas personas e instituciones que han hecho posible este logro.

En primer lugar, quiero expresar mi sincero agradecimiento a mi familia específicamente a mi madre, padre, hermano y abuela por su amor incondicional, constante aliento y apoyo emocional. Su respaldo ha sido mi fuente de fortaleza en cada paso de este viaje académico.

Agradezco a mis profesores en especial al tutor por su dedicación y guía experta. Sus conocimientos compartidos y valiosos comentarios han sido fundamentales para mi crecimiento académico y el desarrollo de esta memoria.

Mis compañeros de estudio merecen un reconocimiento especial por su amistad, intercambio de ideas y experiencias compartidas. Su camaradería ha enriquecido mi tiempo en la universidad y ha sido una fuente de inspiración en mi progreso. Agradezco a la institución por proporcionar un ambiente propicio para el aprendizaje y la investigación, generando oportunidades para enriquecer mi educación.

Mi gratitud también se extiende a mis amigos por su apoyo constante y por ser un pilar de apoyo en momentos de desafíos en los cuales no faltaron las risas y consejos.

Finalmente, deseo expresar mi profundo agradecimiento a todos los que de una forma u otra han contribuido a la realización de esta memoria de título. Este logro es el resultado de un esfuerzo colectivo que se ha desarrollado a lo largo de varios años y me siento honrado y agradecido por cada persona que ha sido parte de mi viaje académico.

Se agradece también al Centro Nacional de Inteligencia Artificial CENIA FB210017, Financiamiento Basal ANID. Por la invitación a la noche CENIA, esencial por el intercambio de ideas con los miembros del equipo donde se adquirió información valiosa.

## Resumen

La intención de este trabajo es avanzar en la adaptación del BCG para su uso en hogares y sistemas Home-Care. El Balistocardiograma (BCG) es el registro de las fuerzas balísticas del cuerpo producidas por la contracción ventricular. La base de datos con los registros BCG a analizar corresponde a la misma utilizada en la competencia de detección de latidos en la Conferencia Internacional en Informática Biomédica y Salud (International Conference on Biomedical and Health, ICBHI). En dicho evento, se presentó el desafío científico de detección de peaks en BCG, evaluando los algoritmos de los participantes mediante un Score que es el mismo método de evaluación para este proyecto.

Se implementan tres algoritmos detectores de peaks, el primero es una recreación de estudios anteriores en la Universidad de Concepción y los otros son modificaciones del primer algoritmo. Un típico problema en esta área son los Artefactos por Movimiento (MAs) los cuales son difíciles de filtrar y entorpecen la detección de latidos. Además, la detección de MAs es un proceso complejo si se trabaja con algoritmos basados en lógica y reglas matemáticas convencionales. Por esta razón se crea una Red Neuronal Convolutiva (Convolutional Neural Network, CNN) especializada para clasificar el ruido y los segmentos limpios, lo que permite cancelar la detección de latidos en zonas de ruido.

Entre los algoritmos detectores de peaks, se destaca el que utiliza técnicas de reconocimiento de patrones como template matching, que implica el uso de una plantilla correlacionada con los complejos IJK de la señal, con el cual se obtuvo un Score igual a 147. Al combinar la detección de peaks y la CNN, el Score mejora a 108. Como referencia, el ganador de la competencia obtuvo un puntaje de 133 puntos. Se manifiesta la utilidad de la red neuronal que mejora notablemente la detección de latidos lo que permite desarrollar un sistema de medición más confiable.

## Abstract

The intention of this work is to advance in the adaptation of the Ballistocardiogram (BCG) for its use in home and Home-Care systems. The Ballistocardiogram is the recording of the ballistic forces of the body produced by ventricular contraction. The database containing the BCG records to be analyzed corresponds to the same one used in the heartbeat detection competition at the International Conference on Biomedical and Health Informatics (ICBHI). In this event, the scientific challenge of detecting peaks in BCG was presented, evaluating participants' algorithms using a Score, which is the same evaluation method for this project.

Three peak detection algorithms are implemented. The first is a recreation of previous studies at the University of Concepción, and the others are modifications of the first algorithm. A common problem in this field is Motion Artifacts (MAs), which are difficult to filter out and hinder heartbeat detection. Additionally, detecting MAs is a complex process when working with logic-based and conventional mathematical rule algorithms. For this reason, a Convolutional Neural Network (CNN) is created to classify noise and clean segments, allowing the cancellation of heartbeat detection in noisy areas.

Among the peak detection algorithms, one that uses pattern recognition techniques like template matching stands out. This involves using a template correlated with the IJK complexes of the signal, resulting in a Score of 147. By combining peak detection with the CNN, the Score improves to 108. For reference, the competition winner obtained a score of 133 points. The utility of the neural network is evident, significantly enhancing heartbeat detection and enabling the development of a more reliable measurement system.

# Tabla de contenidos

|  |           |
|--|-----------|
| ÍNDICE DE TABLAS.....  | VIII      |
| ÍNDICE DE FIGURAS .....  | IX        |
| <b>CAPÍTULO 1. INTRODUCCIÓN.....</b>                               | <b>1</b>  |
| 1.1 OBJETIVOS .....  | 2         |
| 1.1.1 <i>Objetivo General</i> .....                                | 2         |
| 1.1.2 <i>Objetivos Específicos</i> .....                           | 2         |
| 1.2 ALCANCES Y LIMITACIONES.....                                   | 2         |
| 1.3 METODOLOGÍA .....  | 2         |
| 1.4 TEMARIO .....  | 3         |
| <b>CAPÍTULO 2. MARCO TEÓRICO .....</b>                             | <b>4</b>  |
| 2.1 INTRODUCCIÓN.....  | 4         |
| 2.2 BALISTOCARDIOGRAMA .....                                       | 4         |
| 2.2.1 <i>Orígenes del BCG</i> .....                                | 4         |
| 2.2.2 <i>Interpretación del BCG</i> .....                          | 4         |
| 2.2.3 <i>Medidas no invasivas con BCG</i> .....                    | 6         |
| 2.3 SOBRE REDES NEURONALES ARTIFICIALES .....                      | 7         |
| 2.3.1 <i>Generalidades</i> .....                                   | 7         |
| 2.3.2 <i>Redes neuronales convolucionales</i> .....                | 8         |
| 2.4 TRABAJOS PREVIOS.....  | 10        |
| 2.4.1 <i>Sistema de adquisición</i> .....                          | 10        |
| 2.4.2 <i>Trabajos previos en algoritmos de BCG</i> .....           | 12        |
| 2.5 DISCUSIÓN .....  | 24        |
| <b>CAPÍTULO 3. ALGORITMOS Y PROCESAMIENTO DE LAS SEÑALES .....</b> | <b>28</b> |
| 3.1 INTRODUCCIÓN.....  | 28        |
| 3.2 BASE DE DATOS REGISTRADA CON SENSOR EMFI .....                 | 28        |
| 3.3 ALGORITMOS DETECTORES DE PEAKS .....                           | 28        |
| 3.3.1 <i>Jpeaks Cub</i> .....                                      | 31        |

|   |   |           |
|---|---|-----------|
| 3.3.2   | <i>Template matching</i> .....                          | 33        |
| 3.3.3   | <i>Jpeaks Corr</i> .....                                | 34        |
| 3.4   | CNN PARA CLASIFICACIÓN DE SEGMENTOS LIBRES DE MAS ..... | 36        |
| 3.5   | SCORE PARA EVALUAR DETECCIÓN DE PEAKS .....             | 41        |
| 3.6   | DISCUSIÓN .....   | 42        |
| <b>CAPÍTULO 4. RESULTADOS Y DISCUSIÓN</b> ..... |   | <b>45</b> |
| 4.1   | RESULTADOS .....  | 45        |
| 4.2   | DISCUSIÓN .....   | 46        |
| <b>CAPÍTULO 5. CONCLUSIONES</b> .....           |   | <b>47</b> |
| 5.1   | DISCUSIÓN .....   | 47        |
| 5.2   | CONCLUSIONES .....                                      | 47        |
| 5.3   | TRABAJO FUTURO .....                                    | 48        |
| <b>GLOSARIO</b> .....                           |   | <b>50</b> |
| <b>REFERENCIAS</b> .....                        |   | <b>51</b> |

# Índice de Tablas

|     |   |    |
|-----|---|----|
| 2.1 | Hiperparámetros usados por J.P. Moreno . . . . .      | 23 |
| 3.2 | Descripción de la base de datos de Hospital . . . . . | 29 |
| 3.3 | Descripción de la base de datos de Hospital . . . . . | 30 |
| 3.4 | Hiperparámetros del entrenamiento . . . . .           | 39 |
| 4.5 | Scores de los algoritmos . . . . .                    | 45 |
| 4.6 | Promedio de peaks detectados . . . . .                | 46 |

# Índice de Figuras

|      |   |    |
|------|---|----|
| 2.1  | Medición de ICBHI donde se muestran los respectivos ECG-BCG . . . . .           | 5  |
| 2.2  | Características en balistocardiografía . . . . .                                | 6  |
| 2.3  | Diagrama de red neuronal . . . . .  | 7  |
| 2.4  | Perceptrón . . . . .  | 9  |
| 2.5  | Efecto del borde . . . . .  | 10 |
| 2.6  | Convolución 2D . . . . .  | 11 |
| 2.7  | Sensor EMFi . . . . .   | 12 |
| 2.8  | BCG implementado por Javier Chavez . . . . .                                    | 13 |
| 2.9  | Señal no procesada se visualizan los ciclos respiratorios y cardiacos . . . . . | 14 |
| 2.10 | Esquema de descomposición DWT . . . . .   | 15 |
| 2.11 | Señal ideal con BCG . . . . .   | 15 |
| 2.12 | Diagrama de flujo de algoritmo Jpeaks . . . . .                                 | 16 |
| 2.13 | Signal Analyzer . . . . .   | 16 |
| 2.14 | Wavelet toolbox . . . . .   | 17 |
| 2.15 | Ondícula db-6 . . . . .   | 18 |
| 2.16 | Ejemplo transformada del largo con peaks erróneos . . . . .                     | 18 |
| 2.17 | Transformada del largo . . . . .  | 19 |
| 2.18 | Template matching . . . . .   | 19 |
| 2.19 | Transformada de Hilbert . . . . .   | 21 |
| 2.20 | BCG sintético . . . . .   | 22 |
| 2.21 | Proceso de contaminación ECG . . . . .  | 23 |
| 2.22 | Tpología de la CNN . . . . .  | 24 |
| 2.23 | Estructura de la CNN . . . . .  | 24 |
| 2.24 | Autoencoder . . . . .   | 25 |
| 2.25 | Reconstrucción con autoencoders . . . . .                                       | 26 |
| 2.26 | Comparación de modelos de red . . . . .   | 27 |
| 3.27 | Diagrama de flujo de algoritmo Jpeaks Cub . . . . .                             | 31 |

|  |    |
|--|----|
| 3.28 BCG normalizado y amplificado en 3 . . . . .                                  | 32 |
| 3.29 BCG normalizado, amplificado en 3 y elevado al cubo . . . . .                 | 33 |
| 3.30 Template matching . . . . .   | 35 |
| 3.31 Diagrama del algoritmo detector de peaks con correlación de Pearson . . . . . | 36 |
| 3.32 Diagrama para clasificar segmentos . . . . .                                  | 37 |
| 3.33 Diagrama de la construcción de los set de datos . . . . .                     | 38 |
| 3.34 Accuracy y pérdida durante el entrenamiento de la CNN . . . . .               | 40 |
| 3.35 Clasificación con CNN en señal A0146 . . . . .                                | 44 |
| 4.36 Organizacion de resultados . . . . .  | 45 |

## Capítulo 1. Introducción

En la actualidad, el monitoreo y registro de variables cardiológicas son fundamentales para el cuidado de la salud. Desde que en 1901 William Einthoven midió por primera vez la actividad eléctrica del corazón, la medicina ha experimentado avances revolucionarios que han permitido un mayor entendimiento del funcionamiento cardiaco. Junto con el desarrollo de dispositivos que monitorean diversas variables, como temperatura, saturación de oxígeno y presión arterial, también ha surgido la necesidad de mejorar la experiencia del paciente en el proceso de monitoreo.

En busca de una medicina más amigable y cómoda, la tecnología se ha dirigido hacia la medicina no invasiva, donde el uso de dispositivos que no requieren cortes ni introducción de instrumentos es prioritario. El Electrocardiograma (ECG) es un claro ejemplo de esta tendencia, siendo una herramienta no invasiva que ha permitido medir la actividad eléctrica del corazón posando electrodos sobre la piel.

En la misma línea, el Balistocardiograma (BCG) se ha posicionado como una opción mínimamente invasiva. Este dispositivo se basa en sensores de fuerza o presión ubicados en una silla, lo que permite registrar las fuerzas balísticas generadas por la contracción ventricular sin la necesidad de posar electrodos sobre la piel. En este trabajo, se analizan señales BCG registradas con un sensor Película Electromecánica (Electromechanical Film, EMFi) con 200 Hz de muestreo, con el propósito de avanzar hacia un enfoque más domiciliario y amigable, llevando el BCG a las casas y sistemas Home-Care.

Para lograr esta meta, se implementaron algoritmos detectores de peaks, destacando técnicas de reconocimiento de patrones, como el template matching, que correlaciona los complejos IJK de la señal. Sin embargo, el ruido en la señal, proveniente de diversas fuentes como respiración, interferencia de la red y Artefactos por Movimiento (MAs), dificulta la detección de peaks, especialmente en zonas afectadas por MAs del paciente. Con el fin de abordar este desafío, se decide cancelar la detección de peaks en las áreas afectadas por los MAs. Se desarrolló una Red Neuronal Convolutiva (Convolutional Neural Network, CNN) que clasifica el ruido y los segmentos limpios, permitiendo realizar mediciones confiables del BCG en zonas adecuadas.

El presente estudio representa un paso significativo hacia una medicina más accesible y confortable, promoviendo la adopción de tecnologías no invasivas que mejoren la experiencia del paciente en el monitoreo cardíaco. A través de la utilización de técnicas de procesamiento de señales y el uso de redes neuronales convolucionales, se avanza hacia el objetivo de llevar el BCG a entornos de cuidado

domiciliario, brindando una mayor comodidad y confiabilidad en la detección de peaks y mediciones cardiológicas.

## **1.1 Objetivos**

### **1.1.1 Objetivo General**

Mejorar la detección de latidos del corazón en registros BCG utilizando un algoritmo detector de peaks en conjunto con una CNN que clasifique zonas limpias y ruidosas.

### **1.1.2 Objetivos Específicos**

- Implementar algoritmos de detección de peaks en el BCG.
- Implementar una CNN para la detección de ruido.
- Evaluar el algoritmo detector de peaks en conjunto con la red neuronal.

## **1.2 Alcances y Limitaciones**

un algoritmo para detectar peaks en el balistocardiograma y otro para suprimir el ruido utilizando una CNN. Se cuenta con una base de datos que incluye una señal BCG y otra señal ECG tomadas simultáneamente, lo que facilitará la comparación con los peaks detectados por el algoritmo.

Una limitación a tener en cuenta es el tamaño de la base de datos, ya que se considera pequeña para el proceso de creación de la CNN. Esto impide que sea capaz de generalizar y clasificar correctamente.

## **1.3 Metodología**

Se realiza un estudio general sobre la balistocardiografía e investigación detallada sobre papers y memorias hechas por alumnos anteriores.

Integrando lo anterior, se creó un algoritmo detector de peaks en MATLAB debido a su buen rendimiento en el procesamiento de señales. Se escogió Python 3.10 para crear la CNN debido a que es muy

intuitivo y popular en el ámbito de redes neuronales con bibliotecas como Keras v. 2.12.0. y Tensorflow v. 2.12.0. La evaluación en conjunto se realizó exportando archivos desde Python e importándolos en MATLAB para luego calcular el puntaje.

El hardware del notebook con el que se realizó el trabajo cuenta con un procesador AMD Ryzen 5 4600H y 16 GB de memoria RAM, no cuenta con tarjeta de vídeo dedicada.

## **1.4 Temario**

Actualmente, este trabajo se divide en:

- Capítulo 1: Introducción, se presenta el propósito del desarrollo del proyecto, los objetivos generales y específicos.
- Capítulo 2: Marco teórico, se dan a conocer definiciones y características del BCG, los descubrimientos, avances y trabajos previos de diferentes autores respecto al BCG.
- Capítulo 3: Algoritmos y procesamiento de las señales, se explica el funcionamiento y la lógica de los algoritmos.
- Capítulo 4: Discusión General y conclusiones, se sintetiza la información entregada y se dan a conocer algunos puntos de vista del autor.

## **Capítulo 2. Marco Teórico**

### **2.1 Introducción**

A continuación se muestra una breve reseña del BCG para dimensionar y comparar los sistemas antiguos, además se explican conceptos relacionados al BCG para finalmente dar paso a las redes neuronales y otros métodos en el procesamiento de la señal.

### **2.2 Balistocardiograma**

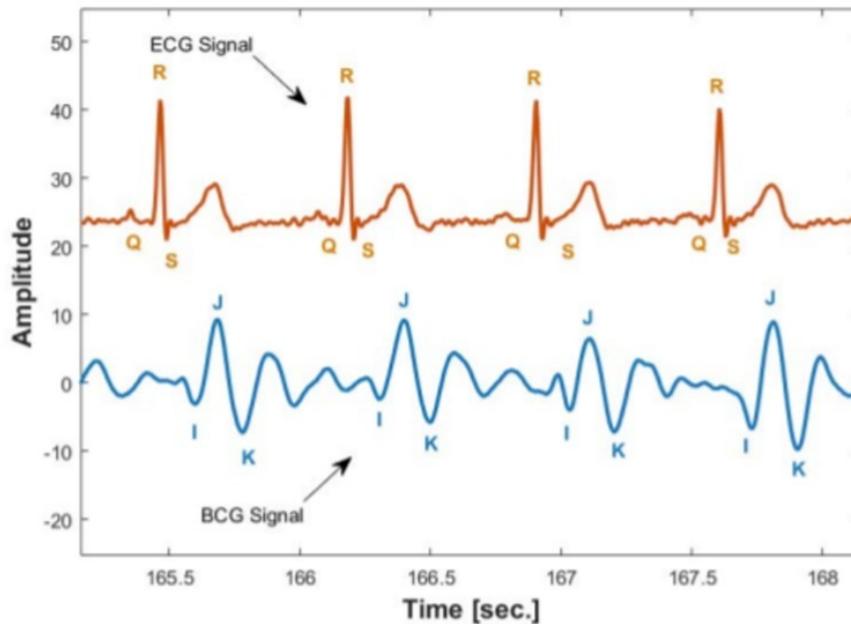
#### **2.2.1 Orígenes del BCG**

El iniciador de los estudios en BCG fue Gordon en 1877, 28 años después, en 1905, Yeadell Henderson utilizaron una mesa oscilante como BCG. Luego en 1928 Angengeister y Lane utilizaron un sismógrafo para registrar las vibraciones corporales. Algunos años más tarde el Dr Starr creó un BCG similar al de Henderson con una mesa suspendida en el techo que impide todos los movimientos excepto los longitudinales, este sistema no está amortiguado. En 1942 Brown describió un BCG portátil que elimina vibraciones extrañas, pocos años después Nickerson y Curtis crearon un sistema con amortiguación y de baja frecuencia. Luego Brown y Pearson introdujeron un sistema electrónico con la finalidad de mejorar el registro. En 1950 Braunstein, Oelker y Gowdy crearon un BCG bidimensional detectando movimientos en sentido longitudinal y transversal [1].

#### **2.2.2 Interpretación del BCG**

Fue en 1902 cuando Einthoven publicó el primer ECG recogido con un galvanómetro de cuerda[2]. Las curvas características del ECG se etiquetan con las letras P,Q,R,S,T, donde el complejo Q-R-S representa la contracción ventricular y la letra R es el punto más alto en cada ciclo del ECG, debido a la gran cantidad de energía utilizada durante la contracción ventricular. Análogo al complejo Q-R-S es el complejo I-J-K que caracteriza la contracción ventricular del BCG. Cabe mencionar que las diferencias entre ambos son la naturaleza de la señal ya que una es eléctrica y la otra mecánica respectivamente. Como se sabe las señales electromagnéticas son más rápidas que las señales mecánicas, pues en el vacío

las ondas electromagnéticas viajan a la velocidad de la luz, debido a esto es que primero se registra la contracción en el ECG y luego en el BCG, ver figura 2.1.



**Figura 2.1: Medición enseña los complejos Q-R-S e I-J-K con su desfase temporal conocido como intervalo R-J [3].**

El BCG utilizado registra la actividad cardíaca a través de sensores EMFi posicionados en la superficie del cuerpo (espalda y gluteos ) detectando el movimiento corporal como reacción a la fuerza de la sangre ejercida sobre los vasos. La señal es originada en los nodos marcapasos del corazón como actividad eléctrica provocando la contracción del miocardio seguida de la eyección de sangre la cual sale desde el ventrículo izquierdo hacia la arteria aorta ramificándose hacia el resto del cuerpo, la sangre al viajar por los vasos sanguíneos choca contra sus paredes, sobre todo cuando se presentan curvas pronunciadas como el cayado de la aorta, pues en esta zona la sangre viaja a gran velocidad por la aorta ascendente y luego de choca con el cayado cambiando de dirección hacia la aorta descendente, se cree que la reacción a este impacto aporta en gran porcentaje a la forma de onda del BCG [4]. Este movimiento hace que las células cercanas y cualquier material alrededor del corazón y arterias también se mueva transmitiendo la energía hasta llegar a la superficie cutánea (de manera similar a la sensación cuando uno se toma el pulso con los dedos sobre la muñeca), donde está posicionado el sensor paralelo a algún plano anatómico para detectar el movimiento sutil del corazón. La señal registrada oscila debido a la resistencia del cuerpo al movimiento o inercia.

### 2.2.3 Medidas no invasivas con BCG

Ya que el BCG registra el complejo IJK el cual es análogo al QRS en el ECG, puede estimar la frecuencia cardiaca y si la medición es muy confiable y segura se pueden detectar arritmias con buena confiabilidad.

En uno de los estudios en BCG hecho por el doctor Isaac Star se observan distintas señales BCG y se corrobora que con el paso de los años de sus pacientes la amplitud de la onda I disminuye, pues a los 60 años ya es aproximadamente la mitad que a los 20 años [5].

Estudios más recientes han encontrado que la amplitud IJ (ver Figura 2.2), es un parámetro importante en la balistocardiografía ya que se correlaciona positivamente con el gasto cardiaco y la disminución de su amplitud a lo largo del tiempo es un indicador de degradación en la salud cardiaca [6].

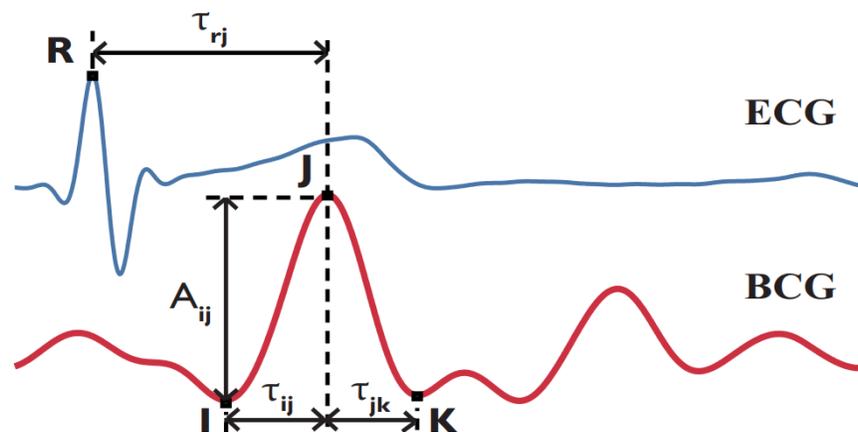


Figura 2.2: En azul se ve el ECG, en rojo el BCG, se destaca el intervalo RJ y la amplitud IJ [6].

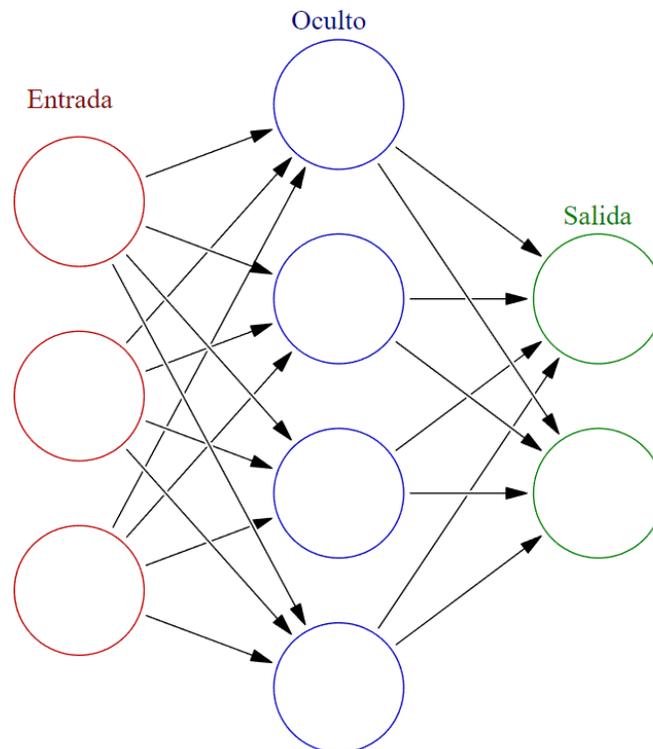
En [7] se utiliza una báscula de baño para sensar BCG, ECG y Pletismografía de Impedancia (IPG), a estas señales se le extraen características mediante un modelo de machine learning y se estima el gasto cardiaco, el cual se compara con el gasto cardiaco medido con el método de Fick antes y después del ejercicio, obteniendo una correlación de Pearson igual a 0.81 y 0.85 respectivamente.

## 2.3 Sobre redes neuronales artificiales

### 2.3.1 Generalidades

La figura 2.3 muestra un diagrama de red neuronal.

Las redes neuronales artificiales permiten modelar y resolver problemas complejos de manera eficiente a partir de datos existentes y métodos de aprendizaje de maquina o reconocimiento de patrones como el aprendizaje supervisado [8]. En el aprendizaje supervisado se induce la capacidad de aprender a través de datos que se les asigna una etiqueta. Es decir que si se quiere enseñar a un computador a clasificar imágenes de perros y gatos, se le entregan o cargan muchas imágenes de perros y gatos como entrada a la red neuronal, especificando cuáles son perros y gatos con la etiqueta 1 y 0 respectivamente, la etiqueta corresponde a la salida de la red neuronal. Luego la red es entrenada y adquiere la capacidad de clasificar este tipo de imágenes.



**Figura 2.3: Diagrama una red neuronal de 3 capas, esta red posee una capa de entrada, una capa oculta y una capa de salida, cada uno de los nodos corresponde a una neurona que puede estar conectada a todas o algunas de las neuronas de la capa siguiente [9].**

La base de las redes neuronales es el perceptrón [10], este es un algoritmo que simula a una neurona

y se puede ver en 2.4, donde  $x_m$  son las  $m$  neuronas de entrada,  $\omega_{km}$  los pesos asignados a cada entrada y  $b_k$  es el bias (similar a un offset), luego se integran todas las neuronas de entrada generando un vector de salida  $v$ , que entran a una función de activación para generar la salida  $y_k$ , la figura 2.4 se traduce matemáticamente a las siguientes ecuaciones:

$$v_k = [x_1 * \omega_{k1} + x_2 * \omega_{k2} + \dots + x_m * \omega_{km}] + b_k \quad (2.3.1)$$

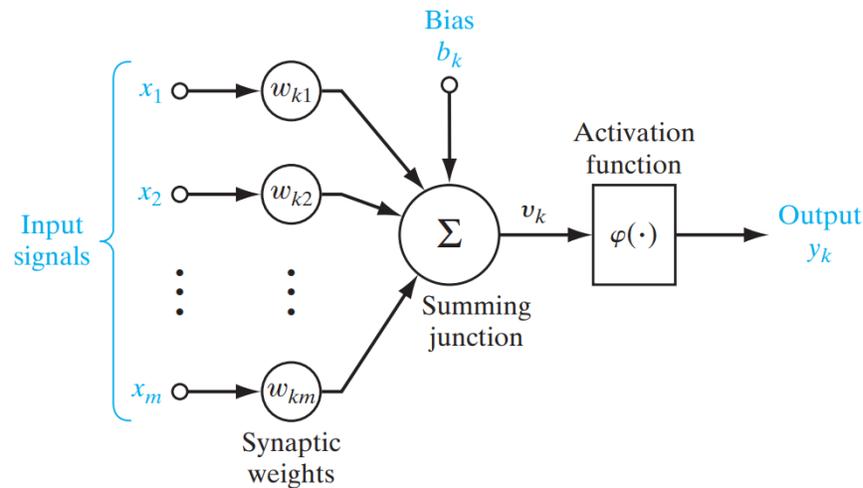
$$y_k = \Phi(v_k) \quad (2.3.2)$$

Si vemos con detalle la ecuación 2.3.1 es una suma de los parámetros de entrada que son lineales, generando a su vez una función lineal, por ende, al agregar más capas seguirá siendo lineal y no podrá resolver problemas no lineales, para eso se utiliza la función de activación lo cual transforma el vector de salida  $v$  a una función no lineal.

Para crear una red neuronal con aprendizaje supervisado necesitamos esencialmente 3 cosas, los datos de entrada, los datos de salida o etiquetas, y una función de pérdida que cuantifique la discrepancia entre la predicción y el valor real. Con los datos de entrada y salida se entrena una red neuronal que ajusta los pesos del modelo en cada época, una época es una pasada de entrenamiento en los datos de la red, los pesos de la red son inicializados aleatoriamente por lo que en un comienzo las predicciones no son buenas, sin embargo a medida que pasan las épocas la red comienza a ajustarse prediciendo con mayor exactitud, este ajuste se realiza mediante un algoritmo llamado backpropagation que busca minimizar la función de pérdida a través del descenso del gradiente. El descenso del gradiente busca el mínimo de la función de pérdida con derivadas parciales de la función de pérdida respecto a los pesos de la neurona multiplicado por un escalar llamado learning rate, siguiendo la regla de la cadena desde la capa de salida hasta la capa de entrada se actualizan los pesos de la red para mejorar las predicciones. La tasa de aprendizaje o learning rate es importante ya que si es muy pequeño, el entrenamiento puede quedar atascado en un mínimo local y nunca encontrará el mínimo global, si es muy grande las predicciones oscilarán sin converger a un valor correcto [11].

## 2.3.2 Redes neuronales convolucionales

Las siguientes explicaciones se ejemplifican con imágenes debido a que existe mucha información sobre la aplicación de CNN's en imágenes. Las imágenes también son consideradas señales solo que están en dos dimensiones como un plano, a diferencia de señales ECG y BCG entre otras que son unidimensionales como una línea.



**Figura 2.4: Diagrama del perceptrón, esta red posee una capa de entrada, una capa oculta y una capa de salida. [12].**

Las CNN o convnet son altamente usadas en visión por computadora y tienen un desempeño mejor que las redes neuronales totalmente conectadas debido a que las CNN extraen características mediante dos tipos de capas, capas con filtros convolucionales y capas de maxpooling, sobre una pequeña parte de la imagen lo que extrae características locales detalladas (ver figura 2.6), en cambio las redes neuronales totalmente conectadas extraen características a partir de la imagen completa.

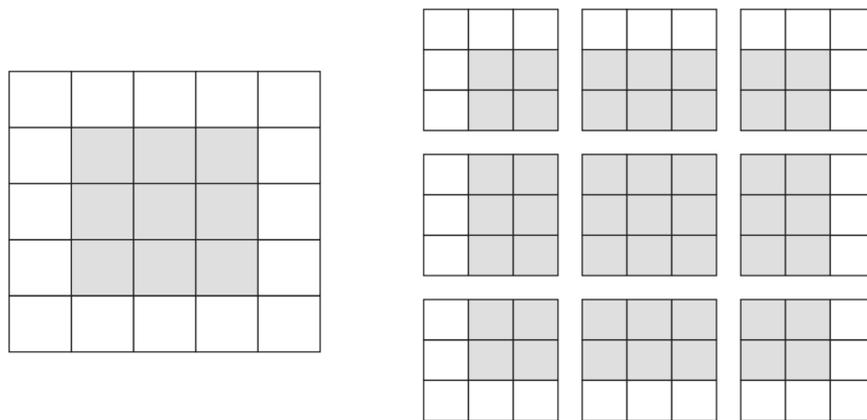
Esta forma de extraer características es invariante al desplazamiento, es decir que se pueden reconocer características similares en diferentes lugares. También se aprenden patrones en orden jerárquico, es decir que la primera capa convolucional extrae características igual al número de filtros, las cuales son la entrada a la siguiente capa convolucional, hasta obtener una salida.

La convolución tiene ciertos problemas como el efecto de borde descrito en el siguiente ejemplo, primero hay que saber que las características se extraen realizando el producto punto del filtro sobre la imagen. Si se considera una imagen de 5x5 pixels con un kernel o filtro de 3x3 pixels, el centro del filtro solo puede posicionarse sobre los 9 pixels centrales por lo que se realizan 9 productos puntos, reduciendo la dimensión de la imagen filtrada a 3x3 pixels, ver figura 2.5. Para obtener una salida de la misma dimensión que la entrada se puede utilizar padding que consiste en añadir un contorno sobre la imagen de 5x5 agrandándola a 7x7, lo que permite posicionar el kernel sobre todos los pixels de la imagen de 5x5.

Otro factor que afecta al tamaño de salida es el stride, el cual es el paso que existe entre las convolu-

ciones y por defecto es igual a uno, si el stride es igual a 2, la siguiente convolución se realiza con una distancia de 2 pixels lo que reduce el tamaño de características a la mitad, este método de submuestreo no es comúnmente utilizado.

Para hacer downsampling se utiliza MaxPooling, este proceso es similar a la convolución solo que en vez de realizar el producto punto con un kernel, se extrae el valor máximo en cada ventana, como los parámetros de la red aumentan con cada capa añadida se obtiene una gran cantidad de parámetros entrenables que puede resultar en sobre entrenamiento, además reduce el esfuerzo computacional y mejora la invarianza de traslación permitiendo comparar características trasladadas fácilmente.



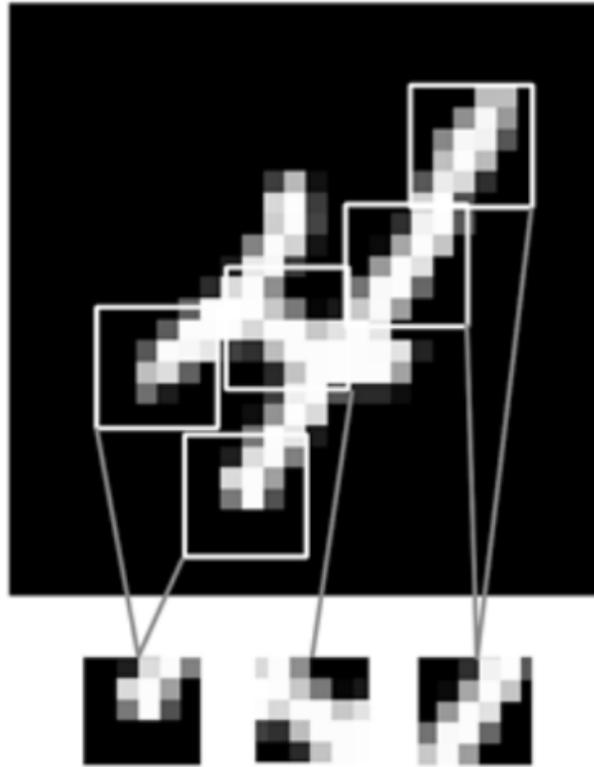
**Figura 2.5:** A la izquierda esta la imagen de 5x5 pixels, el área sombreada corresponde a las posiciones donde el kernel de 3x3 puede posicionarse, en las zonas blancas no puede ya que parte del kernel estaría fuera de la imagen. A la derecha están los cuadros donde el kernel puede realizar el producto punto. [13].

## 2.4 Trabajos previos

A continuación, se presentan los trabajos realizados por estudiantes de Ingeniería Civil Biomédica en años anteriores y además estudios publicados en repositorios.

### 2.4.1 Sistema de adquisición

Como se describe anteriormente, el BCG fue un método de análisis de la actividad cardiaca con un sistema difícil de instalar, lo que perjudicó bastante su utilidad durante varios años debido a la simpleza y exactitud del ECG que fue lanzado en 1902 y hasta la actualidad es frecuentemente utilizado en todo el



**Figura 2.6: Las características para realizar convolución se extraen mediante kernels que corresponden a las pequeñas mallas de pixels de igual tamaño. [14].**

mundo. Actualmente el sistema del BCG se ha simplificado al punto de que puede ser implementado con un paciente sentado sobre una silla con sensores EMFi en el respaldo y asiento, logrando desarrollar un sistema más cómodo, en la figura 2.7 se muestra un esquema de sensor EMFi.

En [16], un estudiante de Ingeniería Civil Biomédica en la Universidad de Concepción realizó un estudio sobre BCG, aquí comparó un Sensor Piezoeléctrico (Piezoelectric Sensor, PVDF) con un EMFi, comprobando que este último tiene un mejor registro ya que perciben mejor los esfuerzos normales o perpendiculares a la superficie del sensor, estos sensores son de forma plana similares a un capacitor ya que en la parte superior tienen carga positiva y en la inferior cargas negativas que están separadas por burbujas de gas, cuando se presiona el sensor la distancia entre cargas se modifica aumentando y disminuyendo la magnitud de la lectura. Una vez registrada la señal biológica cruda, que posee información de frecuencia respiratoria entre (0.1-0.5) Hz, frecuencia cardiaca entre (1-20) Hz y MAs que ocupan una banda de frecuencias más ancha como se observa en la figura 2.9. Este sistema fue utilizado para obtener las señales del estudio.

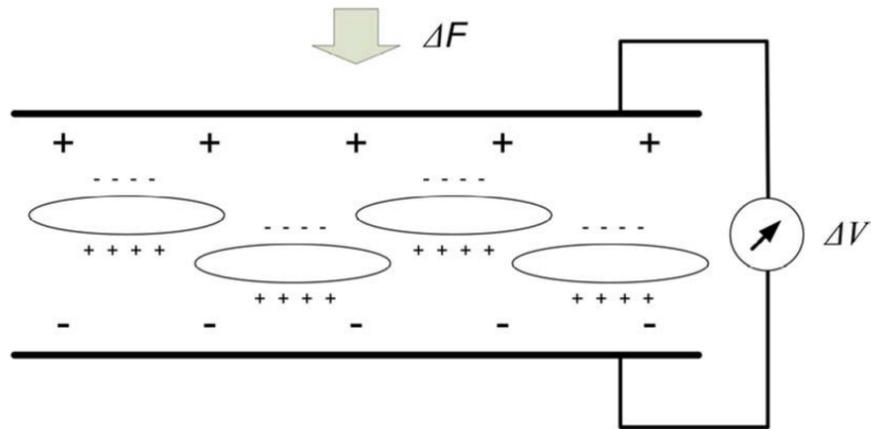


Figura 2.7: Ilustración de sensor EMFi [15].

## 2.4.2 Trabajos previos en algoritmos de BCG

### 1. Generalidades.

Las señales analizadas poseen al menos tres fuentes de información, respiración, frecuencia cardíaca y MAs. Una herramienta para separar la frecuencia respiratoria de la cardíaca son los filtros, sabiendo que el BCG trabaja entre 1 y 20 hz [17], se aplica un filtro pasa banda para limpiar la señal. Otro método es la Transformada Wavelet Discreta (Discrete Wavelet Transform, DWT), esta es una ondícula de energía finita lo que resulta esencial para extraer información de señales de frecuencia variable en el tiempo, al tener energía finita permite extraer información tanto en el dominio del tiempo como en el dominio de la frecuencia. La DWT indica que tanto se parece la ondícula con los segmentos de la señal, por lo que debe utilizarse una wavelet madre similar al segmento de interés que serían los puntos H-I-J-K-L-M-N del BCG. La similitud se calcula realizando el producto interior de la ondícula con la señal, intuitivamente los resultados más positivos indican similitud de la ondícula con el complejo IJK de la señal. La descomposición consiste en aplicar la DWT en varios niveles de descomposición. Básicamente la DWT realiza dos procesos: filtrado de la señal utilizando un filtro pasa bajos (aproximaciones) y un filtro pasa altos (detalles) de descomposición, y el submuestreo de las subseñales de salida, Los dos procesos se repiten varias veces a partir de la señal de salida del filtro pasa bajos[18]. El proceso está en la figura 2.10.

Las aproximaciones son descomposiciones de baja frecuencia y los detalles de alta frecuencia. Como la frecuencia respiratoria es menor que la cardíaca se recompone la señal utilizando los detalles. Luego de separar las señales, surge el problema de cómo detectar los peaks ya que el BCG tiene una forma suave y posee ondas con peaks L-M-N que pueden ser confundidos como peaks



**Figura 2.8: Balistocardiograma implementado en [16].**

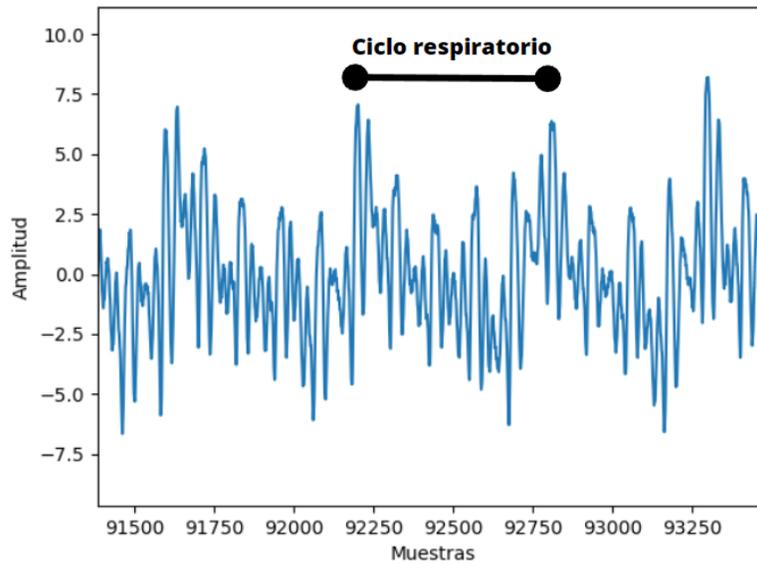
del complejo IJK, ver figura 2.11. Esto hace que detectar un posible punto de máxima energía durante la contracción ventricular registrado en el BCG (Peak J) se presente más complejo que en un ECG. Cabe destacar que en las zonas con mucho ruido el algoritmo comienza a detectar peaks que no corresponden a complejos IJK, si no a peaks de alta energía en la misma banda de frecuencia generados por MAs.

## 2. Recreación del algoritmo detector de peaks [16].

El diagrama del algoritmo está en la figura 2.12, este algoritmo se nombró como Jpeaks y es una de las referencias para la creación de dos variaciones de detectores de peaks que son explicados en el capítulo 3, los pasos se describen a continuación:

- Transformada wavelet.

Para el análisis con DWT se utilizó la ondícula madre “db 6” 2.15 que se encuentra disponible en las ondículas standard ofrecidas por wavelet-toolbox en MATLAB, la elección de la ondícula se hizo visualmente considerando la similitud entre la ondícula y el segmento



**Figura 2.9:** Señales obtenidas por BCG con sensores EMFi, se observa ciclo respiratorio y cardiaco[15].

H-I-J-K facilitando la limpieza de la señal que ahora está lista para ser utilizada por el algoritmo que detecta las contracciones ventriculares, la comparación puede visualizarse en la figura 2.13 donde se muestra el espectro de poder de la señal antes y después de ser filtrado por DWT. La señal se descompone en 7 niveles de aproximación y detalle cómo se observa en la figura 2.14, luego se reconstruye con los detalles del 4 al 7 [16].

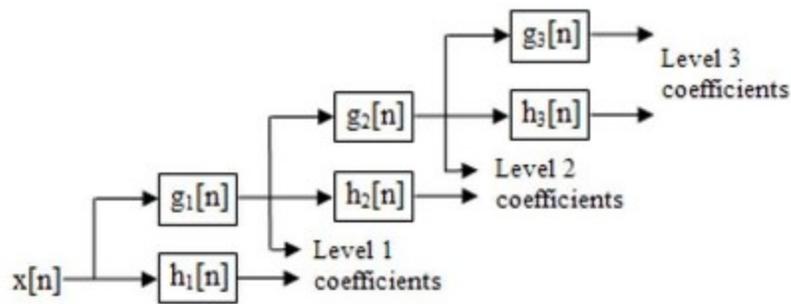
- Transformada del largo.

Luego se aplica la transformada del largo que se define como la distancia euclidiana entre dos puntos consecutivos a lo largo de toda la señal, esta se calcula a partir del teorema de Pitágoras, al aplicar transformada del largo a la señal procesada se crea una forma similar a una envoltura de la señal que ignora en gran cantidad los rebotes u ondas L-M-N que en muchos casos son confundidos con Peak J, por ende los máximos de esta transformada son una buena referencia de Peak J, aun así en esta transformada se observan imperfecciones que dificultan la selección, ante esto se procede a suavizar la señal con una ventana deslizante de 60 muestras que promedia dichos valores.

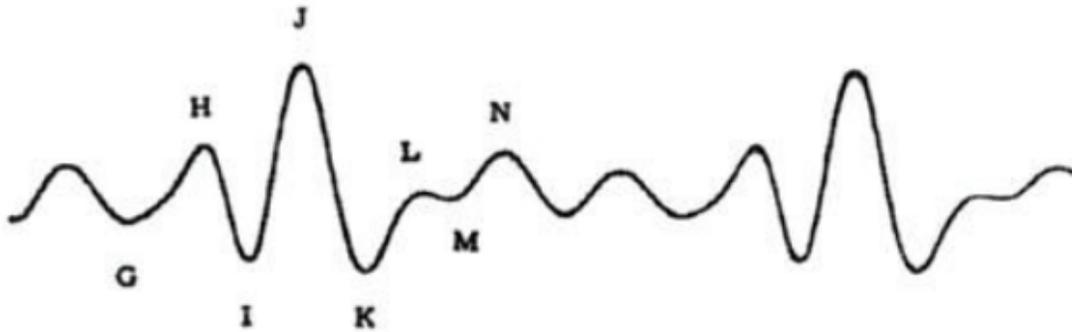
$$LT(i) = \sum_{i=1}^n \sqrt{[F(i) - F(i-1)]^2 + 1} \quad (2.4.1)$$

$$SLT(i) = \frac{1}{60} \sum_{i=n+59}^n LT(i) \quad (2.4.2)$$

- Detección de peaks.



**Figura 2.10: Diagrama del proceso de descomposición por DWT. X es la señal de entrada, g las aproximaciones y h los detalles.[19].**



**Figura 2.11: Señal ideal de BCG con sus peaks correspondientes, se observa la similitud de sus ondas que sumado al ruido de un BCG real dificulta la apreciación de peaks J[20].**

Se detectan los peaks en la transformada del largo suavizada aplicando la función findpeaks con el parámetro distancia mínima entre muestras igual a 300 ms, este valor es considerado como el tiempo mínimo entre dos contracciones ventriculares.

- Primera eliminación de peaks.

Se eliminan los peaks que no son máximos dentro de una vecindad de 300 ms, ya que pese a utilizar la transformada del largo suavizada existen máximos locales en esta función que son confundidos con posibles Peak J, ver máximos locales en la figura 2.16, Para solucionar este problema se eliminan todos aquellos peaks que no son máximos dentro de una vecindad de 300 ms.

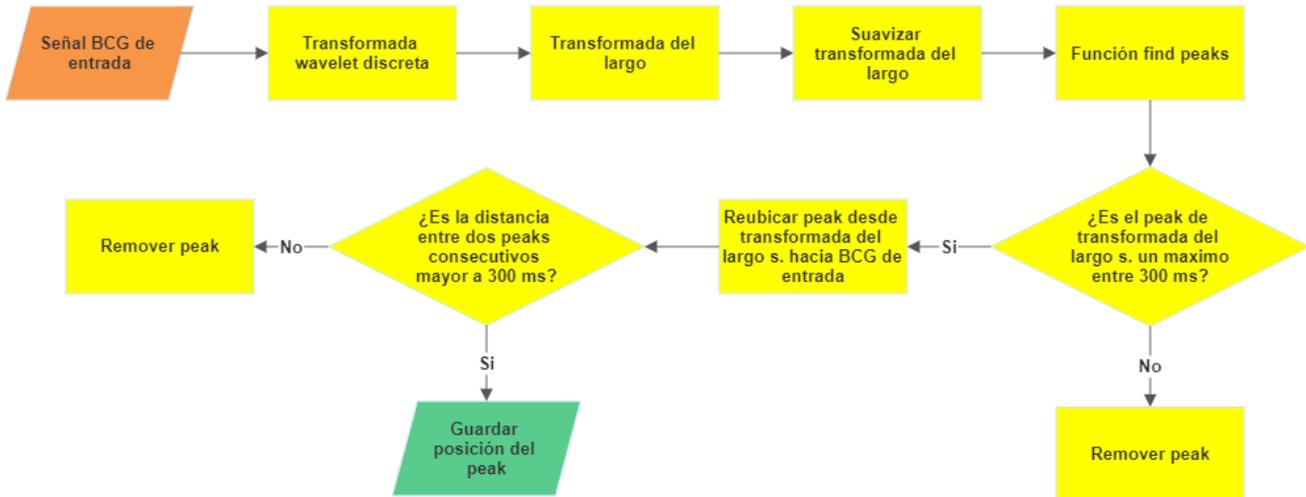


Figura 2.12: Diagrama de flujo del algoritmo Jpeaks.

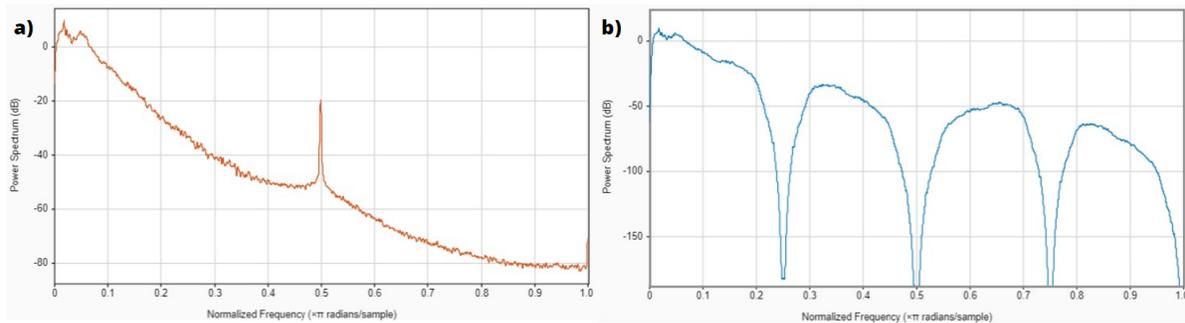


Figura 2.13: En a) se ve el espectro de poder del BCG original y en b) el BCG recompuesto por los detalles. Notar la componente de 50 hz correspondiente a la línea de poder que luego de la DWT desaparece.

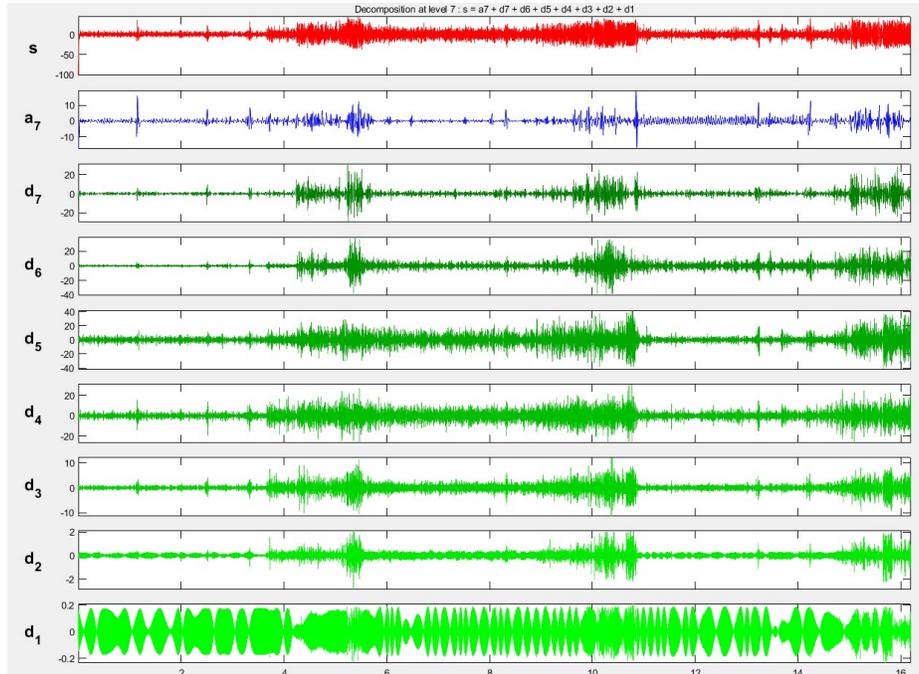
Elaborado en el Toolbox de MATLAB llamado Signal Analyzer

- Reubicación.

Al aplicar la transformada del largo se crea un desfase temporal haciendo que la transformada del largo se adelante en el tiempo como se muestra en la figura 2.17. Por esta razón al trasladar los peaks de la transformada del largo a nuestra señal BCG deben reubicarse, para esto se utilizan los peaks de la transformada del largo como referencia para encontrar el máximo de una ventana que comienza en el punto de referencia y termina en el punto anterior al siguiente peak del punto de referencia. Esto asegura que se reubico en el punto máximo del BCG utilizando el intervalo  $[J_i - J_{i+1}[$  de la transformada del largo.

- Segunda eliminación.

Al reubicar los peaks, se vulnera en algunos puntos el parámetro distancia mínima de 300 ms usado en la función findpeaks. El último paso de este algoritmo consiste en eliminar



**Figura 2.14:** En esta figura se observan las descomposiciones, donde el detalle 1 es el de menor frecuencia y el 7 de mayor frecuencia. Elaborado en el Toolbox de MATLAB llamado Wavelet Analyzer

los peaks que se encuentran a menos de 60 muestras de distancia que corresponde a 300 ms muestreados a 200Hz.

### 3. Métodos de detección de peaks propuestos por otros investigadores.

Otro método utilizado comúnmente en el procesamiento de señales es template matching o plantillas coincidentes, su función es detectar patrones en las señales a base de correlación para cuantificar la similitud de la plantilla con el segmento de señal, debido a esto la plantilla debe representar la forma del ciclo cardiaco similar a la figura 2.11. Se propone en [22] generar un BCG promedio superponiendo segmentos libres de MAs, de igual largo y con todos los Peak J alineados en el mismo tiempo, visualizar en la figura 2.18.

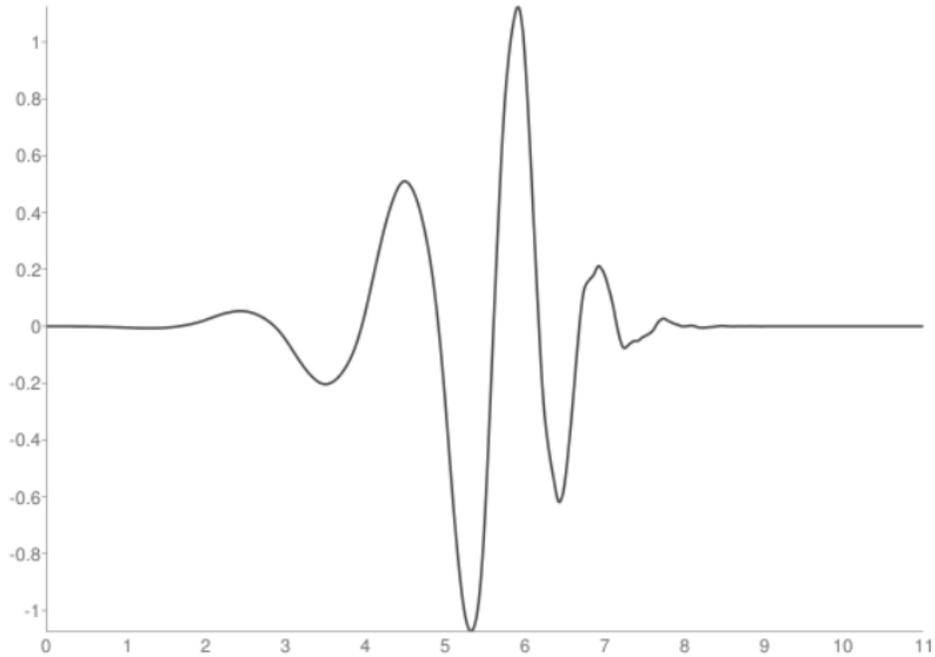
Los tipos de correlación utilizados para detectar patrones de posibles peaks son:

- Correlación cruzada.

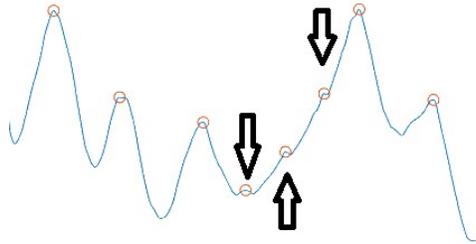
$X$  es el segmento de señal,  $Y$  es la plantilla,  $n$  es el número de muestras

$$X = (x_1, x_2, \dots, x_n) \quad (2.4.3)$$

$$Y = (y_1, y_2, \dots, y_n) \quad (2.4.4)$$



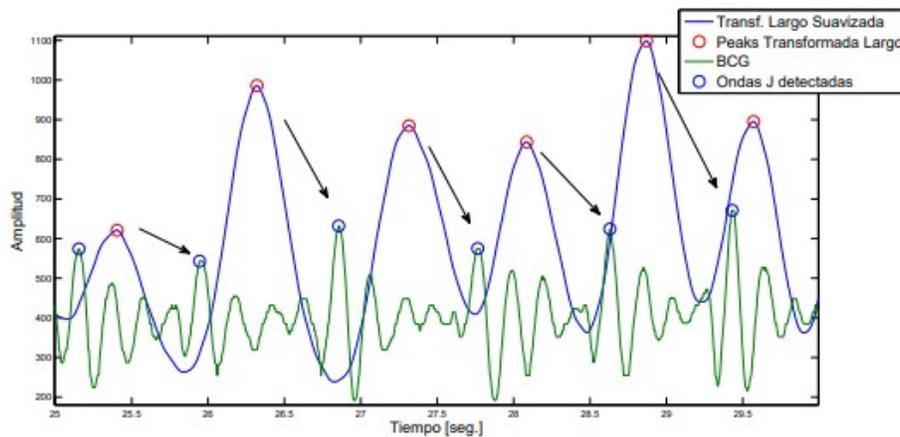
**Figura 2.15:** Wavelet madre de familia Debauchie conocida como 'db-6'[21].



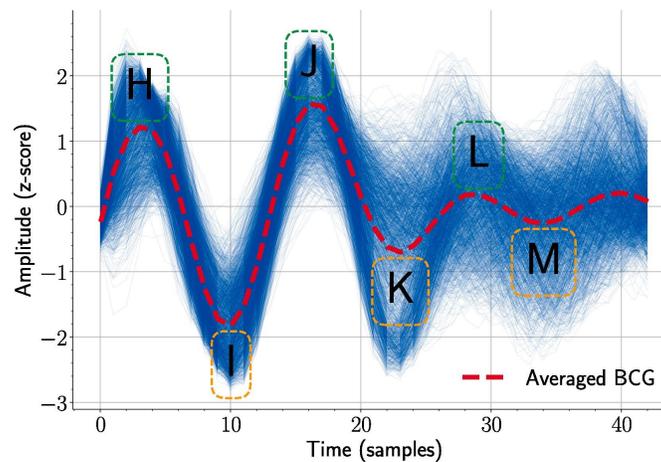
**Figura 2.16:** Se muestran los peaks detectados en el paso anterior y los máximos locales indicados con flechas.

Como consecuencia del deslizamiento de la plantilla sobre la señal se obtiene  $R[k]$  correspondiente al coeficiente  $k$ 'esimo de la correlación cruzada y  $k \in [1, 2n - 1]$

$$\begin{aligned}
 R[1] &= [x_1 \cdot y_n] \\
 R[2] &= [x_1 \cdot y_{n-1} + x_2 \cdot y_n] \\
 &\dots \\
 R[k-1] &= [x_{n-1} \cdot y_1 + x_n \cdot y_2] \\
 R[k] &= [x_n \cdot y_1]
 \end{aligned}
 \tag{2.4.5}$$



**Figura 2.17:** Transformada del largo y señal BCG, se observa como la transformada es una envoltura del BCG.[15].



**Figura 2.18:** Construcción de la plantilla[22].

$R_{X,Y}$  es la correlación cruzada entre  $X$  e  $Y$  equivalente a la suma de los coeficientes.

$$R_{X,Y} = \sum_{i=1}^k R[i] \quad (2.4.6)$$

- Correlación de Pearson.

Para entender la correlación de Pearson, hay que saber que la covarianza mide la similitud de dos conjuntos de variables respecto a sus medias 2.4.7 y la desviación estándar, estima la dispersión de un tipo de variable respecto a su media 2.4.8 (en este caso se muestra la desviación estándar de  $X$  2.4.3).

$$\text{cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (2.4.7)$$

$$\sigma_X = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2.4.8)$$

La correlación de Pearson es el cociente de la covarianza con el producto de las desviaciones estándar. Donde  $r_{X,Y}$  representa al coeficiente de correlación de Pearson 2.4.9.

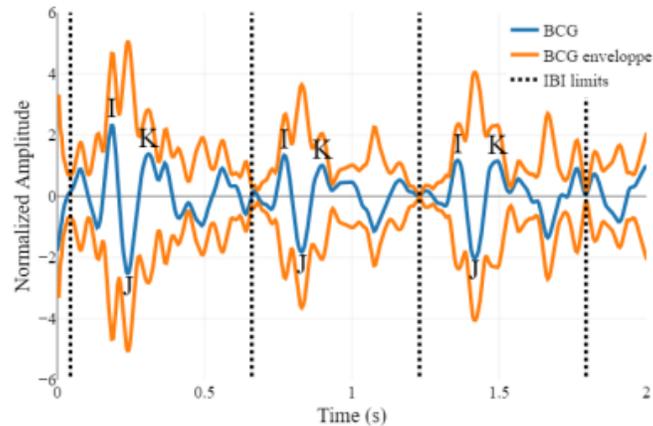
$$r_{XY} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \quad (2.4.9)$$

$$r_{XY} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

Un método similar se presenta en [23], aquí se crea un template matching que es comparado con la señal BCG utilizando la función de distorsión dinámica temporal más conocido como Dynamic Time Warping (DTW), esta función cuantifica la similitud de dos segmentos que no necesariamente deben estar alineados temporalmente ni ser de igual largo como en el método anterior. Los segmentos de la plantilla y el BCG no están alineados ya que esta va cambiando a lo largo del tiempo seleccionando algunos segmentos de la señal a medida que es procesada, los segmentos son seleccionados según mínimos locales en el valor absoluto de transformada de Hilbert de la señal como se observa en la figura 2.19. Los mínimos locales no son espaciados de igual forma pues esto depende de la frecuencia cardiaca del paciente, sin embargo, cada segmento corresponde a un ciclo del BCG, la ventaja de este algoritmo es que facilita la adaptación de la plantilla a diversas morfologías de señales. Otro factor que influye en la participación del segmento en la creación de template matching es la Razón Señal Ruido (Signal Noise Reason, SNR), para esto se calculó el promedio en la SNR en diversas señales, esta referencia sirve de umbral para descartar segmentos con MAs y guardar así los últimos segmentos para actualizar la plantilla deslizante.

#### 4. CNN para supresión de ruido[10]

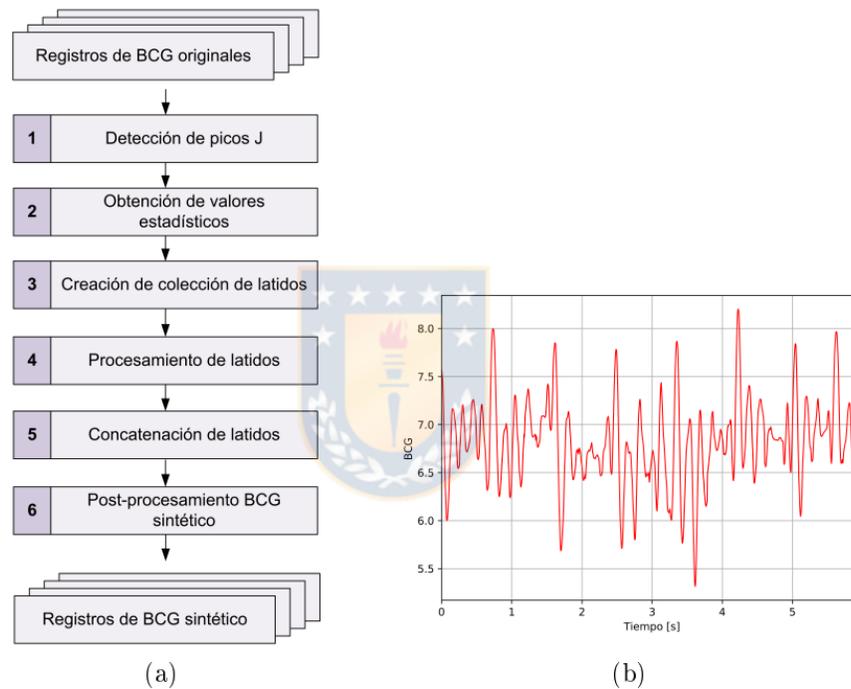
El algoritmo detector de peaks no es confiable en las zonas de ruido. Por esta razón se creó un algoritmo de detección de MAs con una Red Neuronal Profunda (Deep Neural Network, DNN). Las redes neuronales exigen una gran base de datos para ser entrenadas y tener un desempeño confiable, la red se basa en aprendizaje supervisado asignándole la etiqueta (0) a los segmentos limpios y (1) a los segmentos con ruido. Como referencia este algoritmo primero es implementado para ECG pues existen muchas bases de datos disponible como la de MIT-BIH de donde se obtuvieron 116.137 complejos Q-R-S, vale decir que en este proyecto se explica el algoritmo implementado para BCG



**Figura 2.19:** En naranja se observa la transformada de Hilbert, en azul la señal BCG y las líneas punteadas marcan mínimos locales segmentando los intervalos entre latidos, es evidente que los segmentos no siempre serán de igual largo y los peaks no necesariamente están alineados[23].

el cual lamentablemente cuenta con escasas bases de datos, ante la situación se volvió a utilizar la base de datos de Javier Chávez. La base de datos BCG fue adquirida en la sala de espera del hospital regional y cuenta con 35 registros de dos minutos muestreados a 200 Hz, estos registros además incluyen el ECG medido al mismo tiempo que el BCG para hacer pruebas, comparaciones y evaluaciones. Esta base de datos ofrece 4490 complejos Q-R-S lo cual es poco para construir una red neuronal. Para afrontar este problema se creó una base de datos BCG semi sintético, el proceso se observa en la figura 2.20.

Los datos creados corresponden a la clase 0, respecto a la clase 1 correspondiente a MAs se encontró un total de aproximadamente 4 minutos lo que es muy poca información cuando se trata de entrenar una red neuronal, para afrontar este problema se contaminaron las señales con ruido obtenido de la base de datos MIT-BIH NST que contiene varios tipos de ruido como variación de la línea base, artefactos musculares y artefacto por Movimiento de Electrodo (EM), este último es de interés. El ruido se añade de forma controlada para mantener una buena SNR. El esquema de contaminación se observa en la figura 2.21. Cabe mencionar que el esquema corresponde al proceso de contaminación ECG que es idéntico al de contaminación BCG. Así obtuvo 162 registros que son considerados como la nueva base de datos BCG para el entrenamiento de la CNN. El diseño de la CNN es de topología básica y se muestra en la figura 2.22. Para el modelo se presenta un conjunto de hiperparámetros para los cuales se realiza una búsqueda en iterativa, es decir escogiendo algunos hiperparámetros iniciales y luego utilizar la búsqueda en grilla con la finalidad de encontrar el mejor modelo en el menor tiempo posible, esto debido a que si solo utilizamos la búsqueda en grilla el tiempo de procesamiento es extremadamente largo, donde Juan Pablo Moreno sostuvo que era



**Figura 2.20:** En a) se observa el proceso para crear BCG semisintético y en b) se visualiza un extracto.[10].

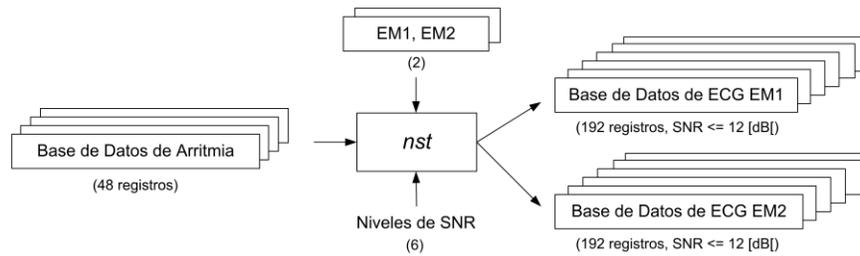
aproximadamente 1 año[10], los hiperparámetros utilizados presentan en 2.1 y la estructura de la red en la figura 2.23.

Luego la red es entrenada con el método K-fold cross validation de 9 particiones, este método también llamado validación cruzada consiste en dividir la base de datos en K particiones y entrenar cada parte por separado, cada fold o partición es una vez conjunto de validación y test.

#### 5. Filtrado de señales mediante redes neuronales.

Las CNN's extraen características de las señales mediante filtros convolucionales, (estos filtros poseen una forma interna similar al template matching para detectar complejos IJK mediante correlación, solo que su forma está diseñada para extraer otras características) esto en parte permite que la red aprenda de los datos de entrada, ya se vio antes que la red puede entrenarse para clasificar, ahora veremos cómo se utiliza para reconstruir datos.

Un autoencoder o codificador automático es un modelo de red neuronal que consta de dos partes, un codificador y un decodificador, el codificador comprime las características de entrada, lo que reduce la dimensionalidad para aprender lo esencial de los datos de entrada, luego el decodificador se encarga de expandir los datos comprimidos de manera que se obtenga una salida similar a la entrada, la diferencia es que la salida solo contiene las características esenciales lo que actúa



**Figura 2.21:** Aquí se muestra como la herramienta *nst* recibe como entrada la señal ECG original, los EM para contaminar dicha señal y la razón señal ruido. En la salida se aprecia la señal contaminada.[10].

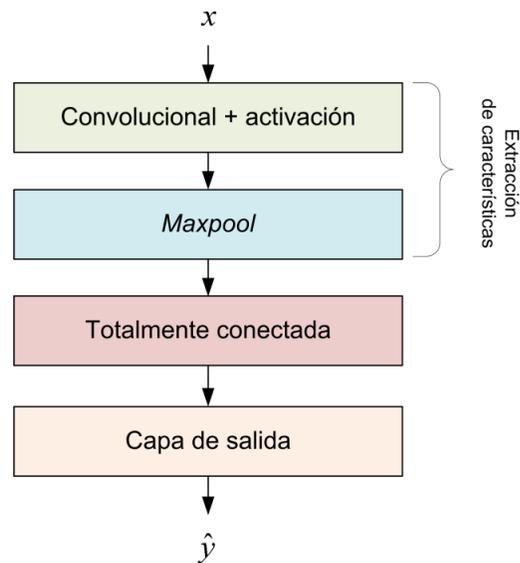
**Tabla 2.1:** Hiperparámetros usados en la búsqueda iterativa [10].

| Hiperparámetro               | Valor inicial | Valor final |
|------------------------------|---------------|-------------|
| Tamaño del vector de entrada | 1 [s]         | 6 [s]       |
| Función de activación        | Leaky Relu    | Leaky Relu  |
| Número de capas FE           | 6             | 8           |
| Tamaño de capa t. conectada  | 512           | 512         |
| Campo receptivo del kernel   | 9             | 9           |
| Número de filtros            | 32            | 32          |

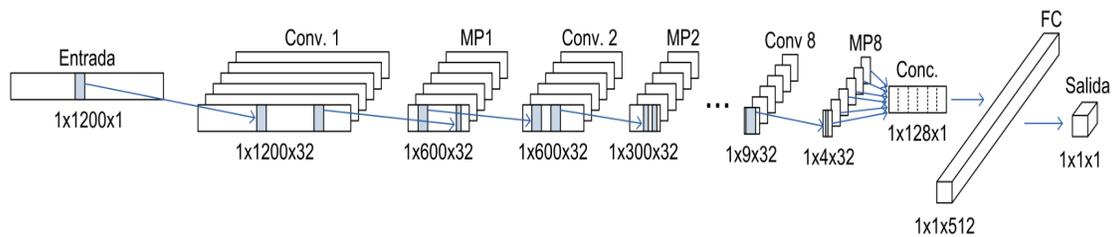
como un tipo de filtro inteligente aplicado en el espacio de características. Ver 2.24

Un grupo de investigadores de la Universidad nacional de Taiwán proponen un tipo de autoencoder denominado FCN o fully convolutional network [24], en este estudio los investigadores contaminaron con ruido algunas señales ECG para luego probar el filtro con tres diferentes tipos de autoencoders, el primero es un autoencoder de capas densas totalmente conectada, la cual es más sencilla y ahorra tiempo de entrenamiento, el segundo es un autoencoder convolucional el cual cuenta con capas de maxpooling que reducen la cantidad de datos. El tercero es el autoencoder FCN que solo tiene capas convolucionales reduciendo la cantidad de filtros no así la cantidad de datos como en el caso anterior. Sus resultados aseguran que el autoencoder totalmente conectado no tiene un buen desempeño en la reconstrucción de la señal, siendo los convolucionales los que más se asemejan a la señal original. Ver figura 2.25

Como el desempeño de la red totalmente conectada no es bueno los investigadores ofrecen la siguiente imagen 2.26 comparando las arquitecturas en los modelos totalmente conectado y convolucional



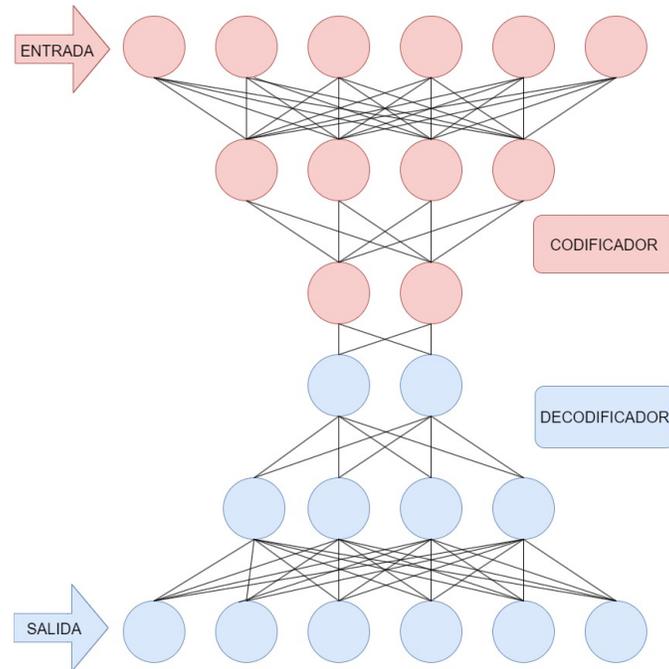
**Figura 2.22: Topología de la CNN [10].**



**Figura 2.23: Estructura de la CNN[10].**

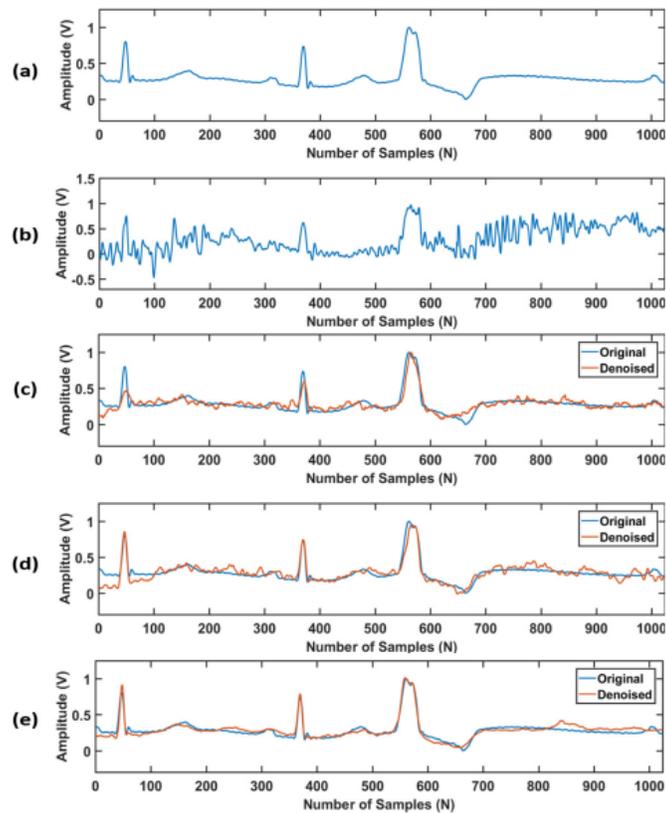
## 2.5 Discusión

El este capítulo se ofrece una breve reseña del BCG, lo cual es esencial para dimensionar y comparar los sistemas antiguos con los más modernos, esta revisión histórica permite apreciar cómo ha evolucionado la tecnología, sobre todo en términos de espacio. Al abordar conceptos relacionados con el BCG, se proporciona una base para aquellos que no estén familiarizados con esta señal biológica, explicar los fundamentos del BCG es importante para que el lector comprenda mejor su aplicabilidad y utilidad, especialmente en el campo de la salud cardiovascular. Además, se menciona el uso de redes neuronales y otros métodos en el procesamiento de señales BCG proporcionando una perspectiva del panorama actual en esta área. Este es un punto clave, ya que las redes neuronales han demostrado ser herramientas poderosas para resolver problemas complejos en diversas disciplinas, permitiendo realizar medidas más

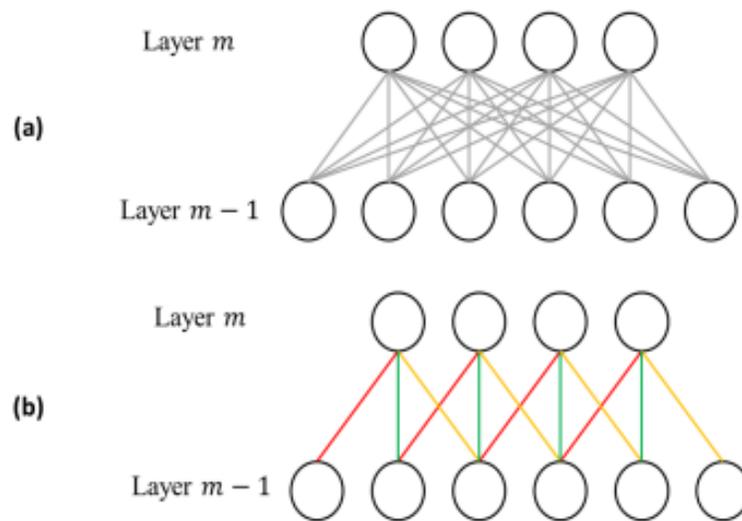


**Figura 2.24: Arquitectura de un autoencoder totalmente conectado. Se observa como la entrada tiene 5 neuronas que equivale a 5 muestras de una señal y se comprime hasta llegar a 2, luego el decodificador es el reflejo del codificador expandiendo las características comprimidas.**

confiables y versátiles.



**Figura 2.25: Reconstrucción o filtrado de la señal. En a) se ve la señal original, en b) la señal contaminada con ruido, en c) la reconstrucción con autoencoder totalmente conectado, en d) la reconstrucción con autoencoder convolucional + maxpooling y e) corresponde al autoencoder totalmente convolucional [24].**



**Figura 2.26:** La imagen a) corresponde al modelo totalmente conectado donde las neuronas de la capa anterior se unen a todas las neuronas de la capa siguiente. En la imagen b) las neuronas de la capa anterior se unen solo a algunas neuronas de la capa sucesora, el número de conexiones se llama campo receptivo que para este caso tiene un tamaño de 3 [24].

## **Capítulo 3. Algoritmos y procesamiento de las señales**

### **3.1 Introducción**

En este capítulo primero se muestra la organización de la base de datos obtenida con el BCG de sensor EMFi, luego se explica el procesamiento de las señales comenzando por los dos algoritmos detectores de peaks probados y finalmente la CNN para clasificar el ruido.

### **3.2 Base de datos registrada con sensor EMFi**

Esta base de datos fue la misma que se utilizó en ICBHI2022 challenge, la cual consiste en cuatro subcarpetas desde A01 hasta A04 con un total de 145 registros BCG y ECG simultáneos. Cada subcarpeta registra diferentes experimentos, vea la tabla 3.2 para más detalles. Luego se crean tres secciones donde se distribuyó equitativamente las señales de los distintos experimentos, las dos primeras se llamaron phase 0 y phase 1 (ambas carpetas con 47 registros cada una), la última corresponde a testing con 51 registros. Todos los algoritmos son creados con los datos de la sección phase 0 y phase 1 para luego ser probados por la carpeta de testing. De esta forma se asegura de que el algoritmo es capaz de generalizar y ser utilizado en nuevas señales capturadas, ver la distribución en la tabla 3.3

### **3.3 Algoritmos detectores de peaks**

Se crean dos algoritmos detectores de peaks para comparar el desempeño, el primero se llama Jpeaks Cub, y el segundo Jpeaks Corr, ambos surgen del algoritmo detector de peak que se llamó Jpeaks descrito en el marco teórico. Los scripts comparten algunos algoritmos o procesos como la DWT, la función findpeaks entre otros, por ello solo se explican las modificaciones y funciones adicionales, el proceso completo de cada script se muestra en el diagrama de flujo.

**Tabla 3.2: Descripción de registros en subcarpetas A01, A02, A03 Y A04, todos los registros contienen ECG y BCG simultáneos adquiridos con sensor EMFi de 200 Hz de muestreo.**

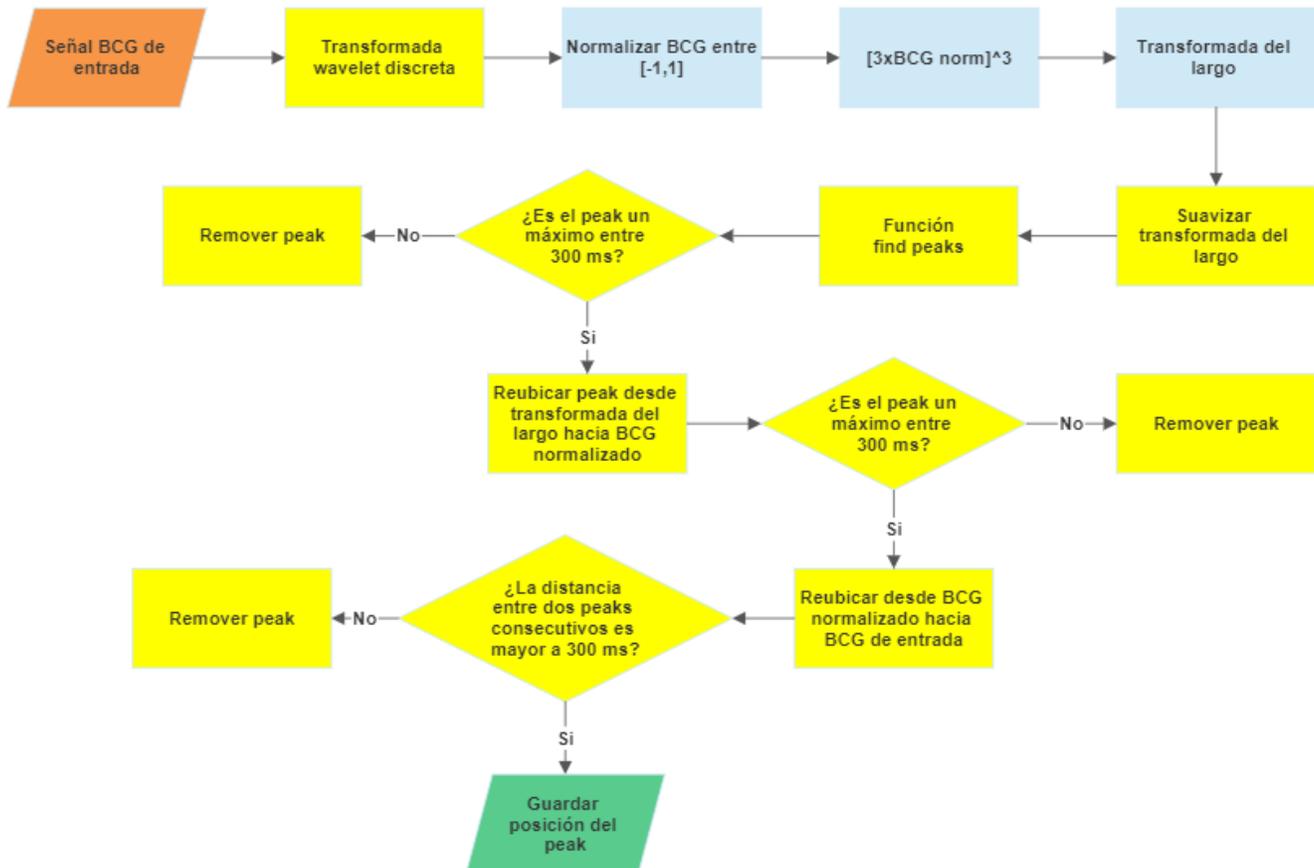
| <b>Carpeta</b> | <b>Contenido</b>                                      | <b>Experimento</b>   |
|----------------|---|--|
| A01            | 68 registros de 13.5 minutos desde A0101 hasta A0168. | 34 voluntarios fueron medidos en dos situaciones diferentes obteniendo dos señales por cada uno:<br>1) Descanso: Los voluntarios hicieron ejercicio durante 1 minuto y luego descansaron sobre la silla.<br>2)Ejercicio: Los pacientes adoptaron tres posiciones diferentes apoyándose sobre el respaldo de la silla y luego presionando los muslos con el asiento luego descansaron durante tres minutos entre cada postura realizando actividades. |
| A02            | 24 registros de 3 minutos desde A0201 hasta A0224.    | Los pacientes son registrados clínicamente con fibrilación auricular.  |
| A03            | 35 registros de 1 minuto desde A0301 hasta A0335.     | Se utilizo una silla con sensor EMFi en el asiento y otro con el sensor EMFi en el respaldo.   |
| A04            | 18 registros de dos minutos desde A0401 hasta A0418.  | Se utilizo una silla con sensor EMFi en el asiento y otro con el sensor EMFi en el respaldo.   |

**Tabla 3.3: Distribución de los registros de las subcarpetas.**

| <b>Phase 0</b>  | <b>Phase 1</b>   | <b>Testing</b>  |
|---|--|---|
| A0107-A0108-A0109-<br>A0110-A0115-A0116-<br>A0119-A0120-A0121-<br>A0122-A0125-A0126-<br>A0133-A0134-A0139-<br>A0140-A0143-A0144-<br>A0155-A0156-A0167-<br>A0168-A0201-A0206-<br>A0209-A0212-A0214-<br>A0217-A0219-A0223-<br>A0302-A0306-A0308-<br>A0310-A0312-A0314-<br>A0317-A0321-A0326-<br>A0330-A0334-A0400-<br>A0402-A0404-A0407-<br>A0410-A0414 | A0103-A0104-A0117-<br>A0118-A0123-A0124-<br>A0127-A0128-A0137-<br>A0138-A0149-A0150-<br>A0153-A0154-A0157-<br>A0158- A0159-A0160-<br>A0161-A0162-A0165-<br>A0166-A0202-A0204-<br>A0207-A0210-A0213-<br>A0218-A0220-A0224-<br>A0303-A0307-A0309-<br>A0311-A0315-A0319-<br>A0323-A0327-A0328-<br>A0331-A0335-A0401-<br>A0405-A0408-A0415-<br>A0416-A0417 | A0101-A0102-A0105-<br>A0106-A0111-A0112-<br>A0113-A0114-A0129-<br>A0130-A0131-A0132-<br>A0135-A0136-A0141-<br>A0142-A0145-A0146-<br>A0147-A0148-A0151-<br>A0152-A0163-A0164-<br>A0203-A0205-A0208-<br>A0211-A0215-A0216-<br>A0221-A0222-A0301-<br>A0304-A0305-A0313-<br>A0316-A0318-A0320-<br>A0322-A0324-A0325-<br>A0329-A0332-A0333-<br>A0403-A0406-A0409-<br>A0411-A0412-A0413 |

### 3.3.1 Jpeaks Cub

El diagrama de flujo de Jpeaks Cub se ve en la figura 3.27. Las modificaciones que están en los cuadros de color azul se describen a continuación.



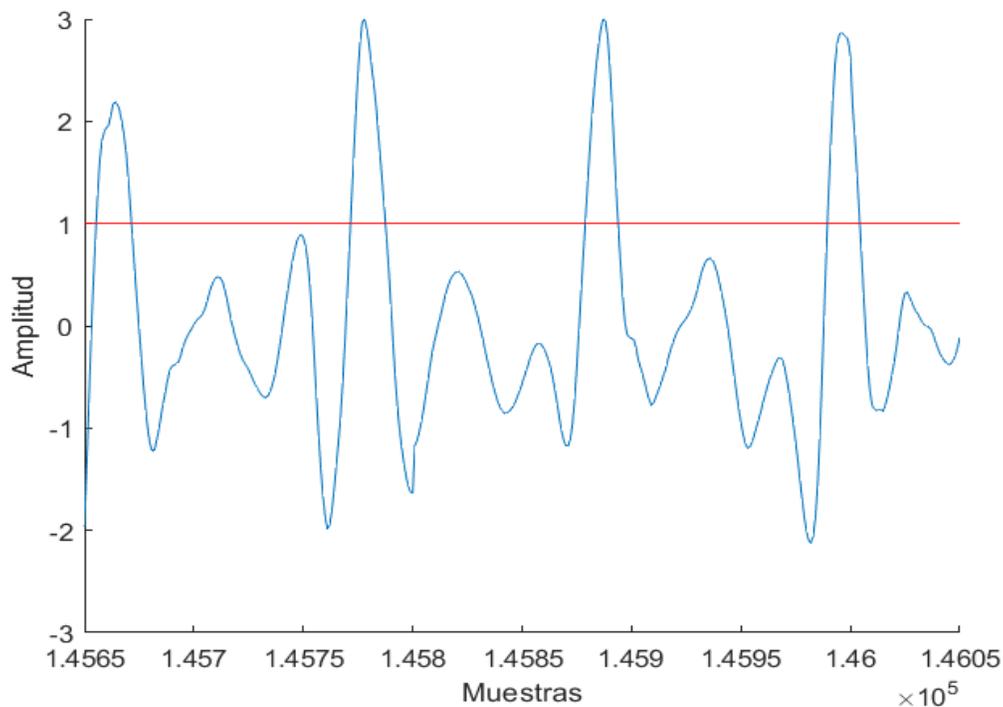
**Figura 3.27:** Pasos para realizar el algoritmo Jpeaks Cub, los bloques amarillos corresponden al algoritmo Jpeak original los cuales pueden estar duplicados y aplicados en diferente orden sin embargo siguen la misma lógica, los bloques azules se agregaron recientemente y se describen en esta sección.

#### 1. Normalizar la señal.

Normalizar la señal entre menos uno y uno es esencial para tener una buena señal cubica ya que cuando no está normalizada en algunas zonas de MAs la razón señal ruido aumenta exponencialmente al elevar al cubo. Además, permite umbralizar las amplitudes fácilmente ya que todos los peaks se encuentran a una escala similar. Para normalizar la señal se aplica una ventana de un segundo con desplazamiento de un segundo en toda la señal, en cada iteración se divide el segmento de señal en el máximo del valor absoluto del segmento.

## 2. Amplificar la señal normalizada y elevarla al cubo.

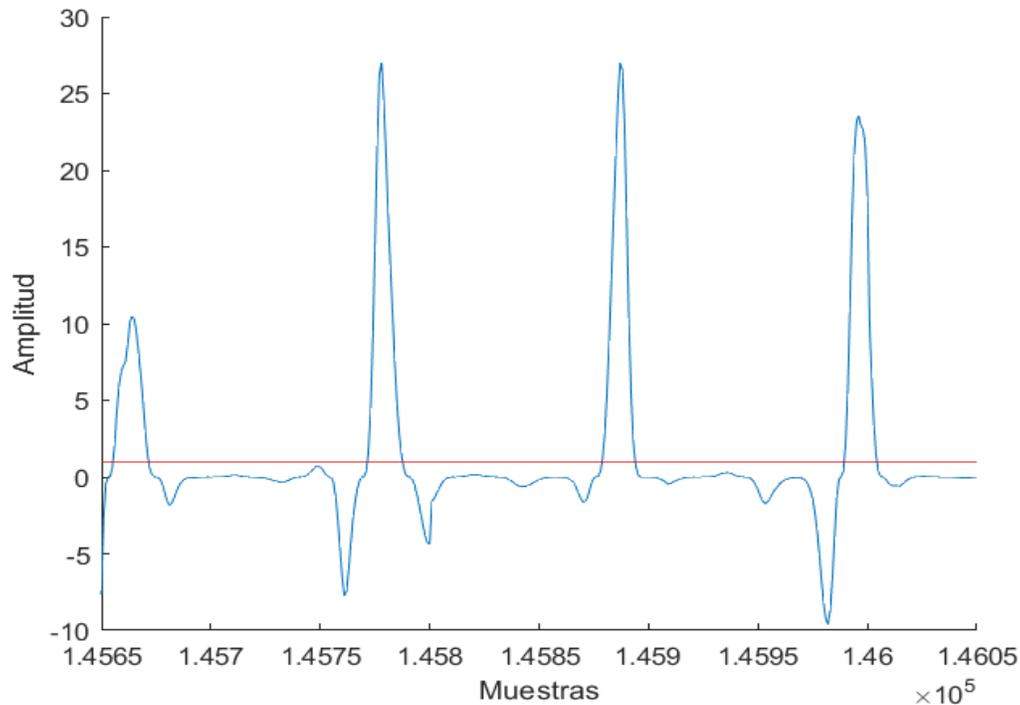
Se amplifica la señal normalizada por un factor de 3. Luego es elevada al cubo ya que al ser una función impar mantiene los puntos dentro de los mismos cuadrantes. Este proceso umbraliza la señal indirectamente en  $1/3$  de su amplitud debido al comportamiento matemático de las funciones exponenciales donde cualquier valor en el intervalo  $]-1, 1[$  disminuye y cualquier valor fuera de este rango crece de forma exponencial. Esto remarca el complejo I-J-K lo que ayuda a realizar una transformada del largo más precisa al disminuir en proporción al resto de la señal las longitudes de los rebotes L-M-N. Al realizar el proceso se sacrifica la posibilidad de encontrar un Peak J en la zona intermedia, aunque esta posibilidad de por sí ya es baja.



**Figura 3.28: BCG normalizado y amplificado en 3.**

## 3. Reajuste de constante en la transformada del largo.

Cambio en la constante 1 dentro de la transformada del largo 2.4.1, la transformada del largo calcula la hipotenusa formada por el triángulo imaginario entre dos muestras consecutivas [16], esto se debe a que el uno representa la distancia muestra a muestra y no el tiempo real entre dos muestras, sabiendo que el sensor tiene una frecuencia de muestreo de 200 Hz lo que equivale a 0.005 s entre muestras. Cabe señalar que este cambio mejora en el puntaje es solo de unas cuantas



**Figura 3.29:** BCG normalizado, amplificado en 3 y elevado al cubo, al comparar con la imagen anterior se observa como las amplitudes bajo la línea umbral disminuyen y las que están sobre la línea aumentan.

unidades.

$$LT(i) = \sum_{i=1}^n \sqrt{[F(i) - F(i-1)]^2 + 0.0025} \quad (3.3.1)$$

### 3.3.2 Template matching

En la segunda variación de este algoritmo se crea una plantilla con el método de Ibrahim Sadek y Bessam Abdulrazak [22], conocido como template matching. La plantilla creada es correlacionada con la señal para detectar posibles peaks.

#### 1. Recopilación de datos.

Para crear esta plantilla se utilizó el algoritmo de variación 1 descrito en el capítulo anterior para guardar las posiciones donde ocurren los Peak J, este proceso se reiteró en Python con la mayoría de las señales (descartando las de ruidos triangulares) en las carpetas phase 0 y phase 1. Se creó un diccionario con los nombres de las señales como claves y sus posiciones correspondientes en listas.

## 2. Limpieza de los datos.

Luego se debe verificar que los tiempos no estén dentro de una zona con MAs para ello se siguen los siguientes pasos finalizando con una limpieza visual de datos, la cual se recomienda debido a que el algoritmo no es muy preciso, sin embargo, es más rápido a si solo fuera una limpieza visual.

- Crear una ventana de 30 segundos con desplazamiento (overlap) de 15 segundos.
- Guardar en un vector la desviación estándar de cada ventana deslizada a lo largo de la señal.
- Calcular desviación media absoluta (DMA) de las desviaciones estándar para servir de umbral.

$$DMA = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}| \quad (3.3.2)$$

- Finalmente las ventanas con desviación estándar mayor al doble de DMA se consideran ruido.

## 3. Creación del template matching.

Después de separar las zonas con MAs se crean ventanas de 130 muestras con los Peak J normalizados en el centro, obteniendo una matriz  $M_{n,130}$  donde n corresponde a la cantidad de plantillas sin MAs. Una vez creada la plantilla se promedia obteniendo un vector de 130 columnas que corresponde al template matching.

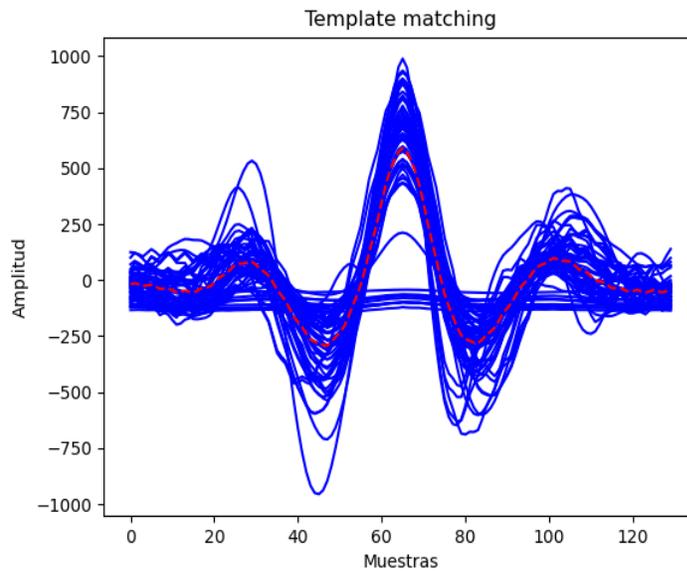
$TM[k]$  es la amplitud de la muestra k'esima del template matching

$$TM[k] = \sum_{i=1}^n M_{i,k} \quad (3.3.3)$$

$$k \in [1 : 130]$$

### 3.3.3 Jpeaks Corr

Este algoritmo funciona en conjunto con el template matching descrito en la sección anterior. El diagrama de este algoritmo está en la figura 3.31, el cual en sencillas palabras funciona detectando los



**Figura 3.30: En color rojo se ve la plantilla para la correlación. Se creó promediando los complejos IJK de todas las señales sin MAS en color azul.**

peaks del BCG deslizando muestra a muestra el template matching sobre la señal BCG mientras se correlacionan en cada paso lo que puede estimar la presencia de un Peak J, los procesos nuevos del algoritmo son descritos a continuación.

1. Correlación de Pearson entre la plantilla y la señal.

Para comparar la señal y la plantilla se utiliza el método de Pearson. Para esto se posiciona la ventana sobre la señal, encontrando los coeficientes a medida que es deslizada muestra a muestra. El resultado es la señal de correlación llamada Corr BCG.

2. Elevar Corr BCG al cuadrado.

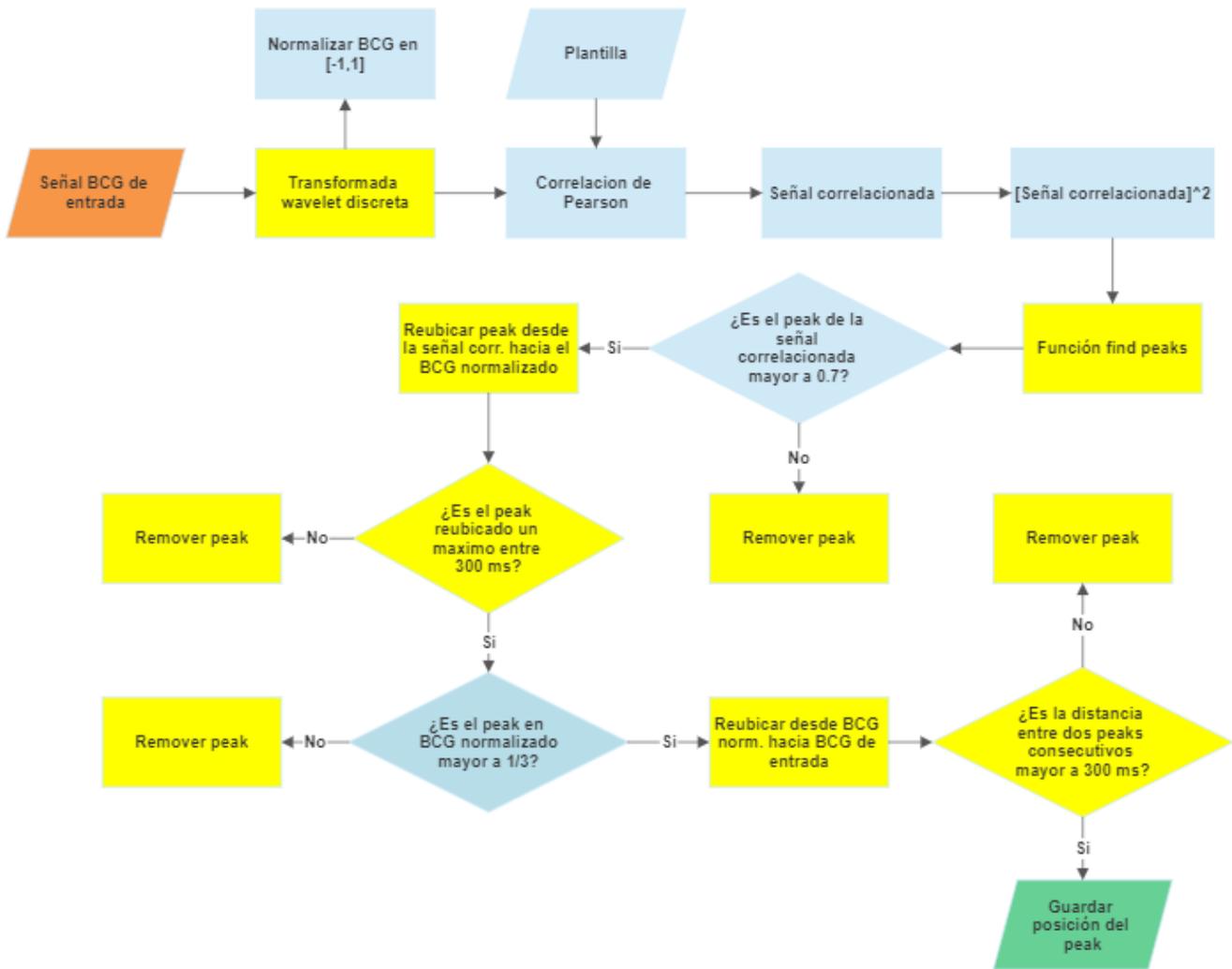
La señal Corr BCG es elevada al cuadrado, esto hace mayor la diferencia entre las amplitudes bajas y altas facilitando la detección de peaks.

3. Eliminación de peaks con baja correlación.

Una vez detectados los peaks de Corr BCG al cuadrado se eliminan aquellos que no superen una correlación de 0.7 (Las posiciones de los peaks al cuadrado son las mismas que en Corr BCG).

4. Eliminación de peaks con baja amplitud.

Luego de reubicar los peaks de Corr BCG en filt BCG y verificar los máximos, se utiliza la señal llamada Normalized Filt BCG que se normaliza en el intervalo  $[-1, 1]$  con ventanas de 3



**Figura 3.31: Diagrama de flujo explica el algoritmo detector de peaks que utiliza correlación de Pearson, los bloques siguen la lógica del algoritmo Jpeaks, son utilizados en diferente orden pero siguen la misma lógica, los bloques azulados se agregaron recientemente y se describen en esta sección.**

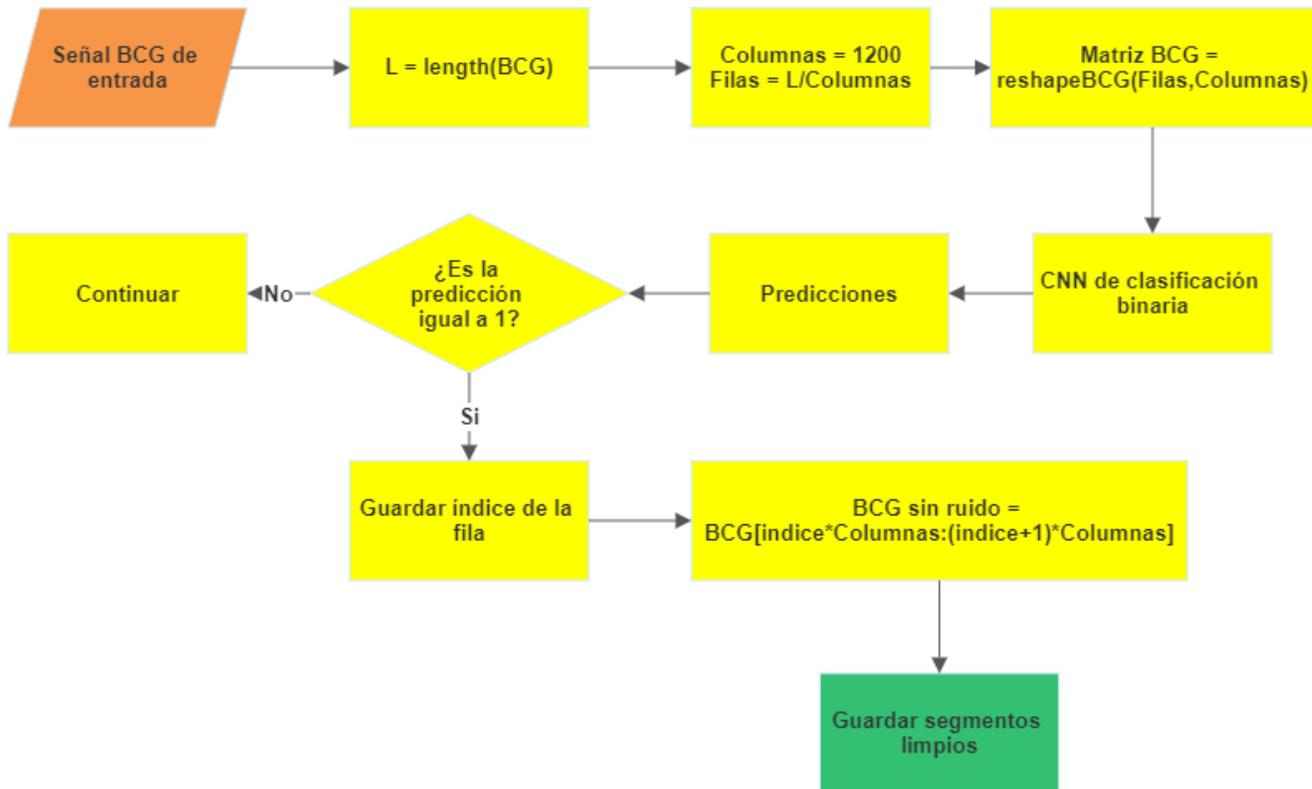
segundos, entonces ahora existe al menos un máximo de amplitud 1 cada 3 segundos, los peaks detectados dentro cada ventana con amplitud menor a  $1/3$  son removidos.

### 3.4 CNN para clasificación de segmentos libres de MAs

#### 1. Descripción del trabajo.

A continuación se explica como crear una red neuronal para clasificar los segmentos limpios como 1 y segmentos con MAs u otro tipo de ruido como 0. Aquí surge la problemática de que

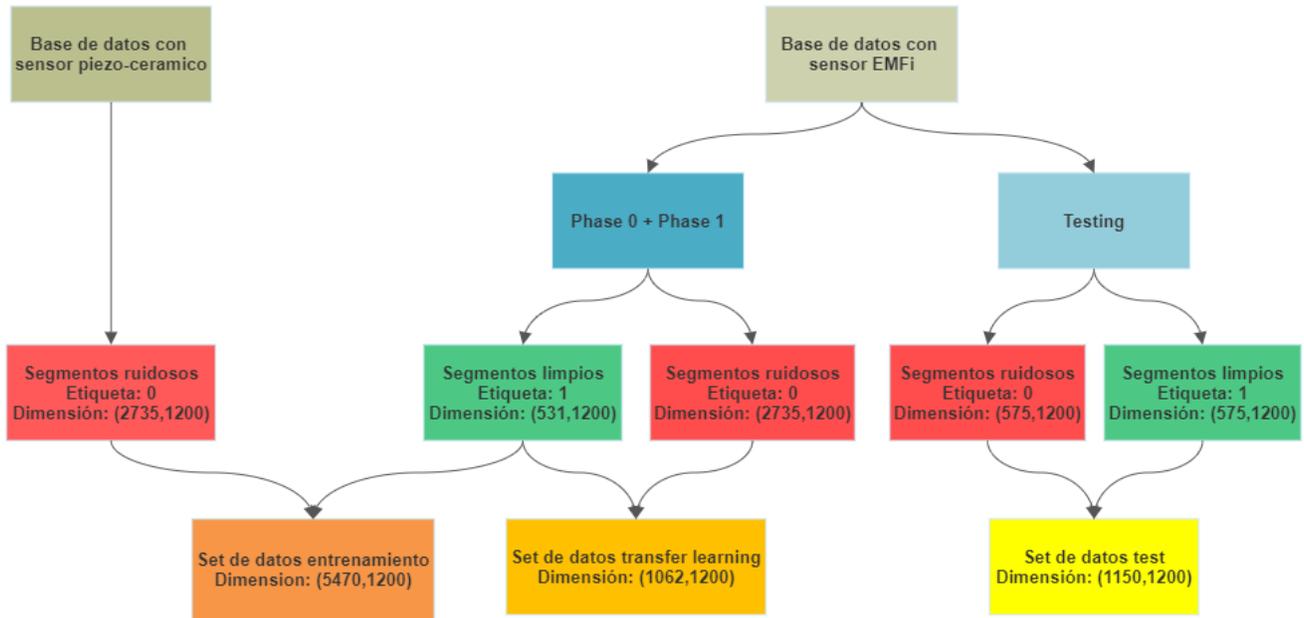
los datos proporcionados por el sensor EMFi son insuficientes para que la red neuronal generalice y clasifique confiablemente, sobre todo por la baja cantidad de MAs, por ello se utiliza una base de datos BCG encontrada en internet que proporciona gran cantidad de información sobre los MAs, cabe mencionar que a diferencia de nuestro BCG que tiene un sensor EMFi con 200 Hz de frecuencia de muestreo, la base de datos de internet es registrada con un sensor piezo-cerámico que muestrea a 330 Hz[25]. Ahora se puede entrenar una red neuronal mixta que aprende las características de la clase 0 y clase 1 con las señales adquiridas por el sensor piezo-cerámico y EMFi respectivamente. Luego se procede a usar el aprendizaje por transferencia lo cual permite que la red aprenda de una base de datos más pequeña, el transfer learning consiste en utilizar el aprendizaje de la red pre-entrenada, transfiriendo el aprendizaje de la base de datos mixta a una base de datos adquirida solamente con el sensor EMFi. Finalmente, la red es incluida en un algoritmo que clasifica segmentos de la señal BCG, el diagrama de flujo se puede ver en la figura 3.32.



**Figura 3.32: Diagrama del algoritmo para clasificar segmentos libres de ruido. Este algoritmo se llama Find Clean Segments.**

## 2. Creación de conjuntos de datos.

El diagrama de la figura 3.33 muestra cómo se crean los sets de datos.



**Figura 3.33: Procedimiento para crear los conjuntos de datos utilizados en el entrenamiento de la CNN, transfer learning y test.**

Para realizar este proceso se cuenta con las carpetas phase 0, phase 1, testing, y una adicional extraída de internet, esta es nombrada db internet[25]. para entrenar se utilizaron las carpetas phase 0 y phase 1, estas proporcionan una gran cantidad de elementos de clase 1, es decir segmentos libres de MAs, por su parte no tiene suficientes datos de MAs lo que genera una base de datos muy desbalanceada, por lo que la clase 0 se extrajo de la base de datos de internet, finalmente se formó una matriz de entrenamiento con forma  $(5470 \times 1200)$ , donde las filas corresponden a la cantidad de segmentos y las columnas a las muestras de cada segmento, cabe destacar que la matriz está distribuida equitativamente para clases positivas y negativas, esto quiere decir que hay 273 minutos y 30 segundos de información para cada clase.

El conjunto de prueba se crea con las señales de la subcarpeta A01 distribuidas en la carpeta testing, donde el experimento 1 aporta la clase 1 y el experimento 2 la clase 0, ver la tabla 3.2 para información más detallada, así se obtiene una matriz de forma  $(1150 \times 1200)$  distribuida equitativamente para cada clase, es decir se cuenta con 57 minutos y 30 segundos para cada clase.

Todos los conjuntos de datos son aleatorizados, así se evita que la red aprenda patrones con relación a agrupaciones dentro del dataset o relaciones causales, lo que podría resultar en una predicción incorrecta.

Para el transfer learning se crea otro conjunto de entrenamiento con las carpetas phase 0 y

**Tabla 3.4: Hiperparámetros escogidos para entrenar la red neuronal.**

| <b>Hiperparametros</b>       | <b>Valor</b>  |
|------------------------------|---------------|
| Tamaño del vector de entrada | 1200 muestras |
| Función de activación        | Leaky Relu    |
| Numero de capas FE           | 8             |
| Tamaño de capa t. conectada  | 512           |
| Campo receptivo del kernel   | 9             |
| Número de filtros            | 32            |
| Learning rate                | 0.001         |

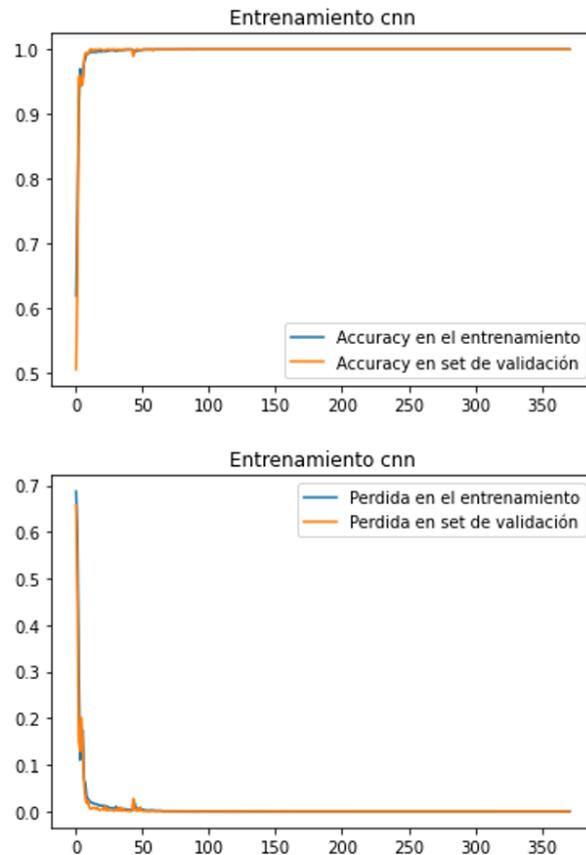
phase 1, la clase positiva corresponde a una sección del entrenamiento inicial de la red neuronal, y la clase negativa ahora corresponde a las carpetas phase 0 y phase 1. Como la cantidad de MAs de esta base de datos es pequeña, se limita la cantidad de datos de la clase positiva al tamaño de los MAs entregados por esta carpeta para crear una matriz de forma (1062x1200), es decir se cuenta con 53.1 minutos para cada clase.

Las muestras de ruido de las carpetas phase 0, phase 1 y testing se extraen visualmente, Las muestras de ruido de la carpeta de internet se extraen con el algoritmo de limpieza de datos explicado en 3.3.2, sin embargo igual se debe recurrir a la limpieza visual ya que algunos segmentos son mal seleccionados. Todas las bases de datos están balanceadas al 50 %.

### 3. Modelo y entrenamiento de la red neuronal.

El modelo de la red neuronal escogido fue el mismo utilizado en 2.23 , en su búsqueda iterativa se menciona cuales son los mejores hiperparámetros para entrenar esta arquitectura de red convolucional y se presentan en la tabla 3.4, el hiperparámetro dropout disminuye el sobreajuste apagando ciertas neuronas en cada paso del entrenamiento[26]. Esta red se entrena con la matriz mixta de forma (5479x1200) la cual es normalizada para evitar sesgos de amplitud, ya que la clase limpia proviene de un sensor EMFi y la la clase de ruido que corresponde de un sensor piezo-cerámico que es más sensible a los cambios de presión, de esta misma matriz se extrajo un 10 % para su validación y realizar early stoping, al early stoping o parada temprana es uno de los Call-backs [27] ofrecido por keras, estos son acciones que se pueden realizar durante el entrenamiento, dentro de dichas acciones se encuentra early stoping [28], que detiene el entrenamiento cuando la métrica escogida no mejora en un cierto número de épocas, la métrica escogida para este caso es el valor de la función de pérdida.

El resultado del entrenamiento se ve en la figura 3.34, donde se obtiene un accuracy de 0.9981 en el conjunto de validación y 0.6372 en el conjunto de test. La pérdida tuvo un valor de 0.0137.



**Figura 3.34: Accuracy y pérdida durante el entrenamiento. Tanto en la pérdida como en la validación, se observa similitud entre el conjunto de entrenamiento y el de validación, lo que indica que el modelo está generalizando correctamente. Además, la precisión es alta, lo que sugiere que las etiquetas están asignadas de manera relativamente precisa.**

#### 4. Transfer learning.

Como esta la red es entrenada con datos de un sensor piezo-cerámico, no tiene buen desempeño en el sistema creado con sensor EMFi, este problema se puede solucionar con transfer learning o transferencia de aprendizaje[29]. Para realizarlo se necesita una red pre-entrenada la cual corresponde a la CNN entrenada con datos mixtos, luego se congelan las capas convolucionales que se encargan de extraer características, esto quiere decir que las capas convolucionales no serán entrenables y se dejan descongeladas solo las capas densas que se encargan de reunir la información para luego clasificar, así se entrenan estas capas descongeladas nuevamente, con menos datos

y sin cambiar los pesos de las capas convolucionales, adaptando las capas densas y salidas a los registros adquiridos con sensor EMFi. Además, como son pocos datos se decide realizar validación cruzada conocida también como K-fold, de 9 iteraciones, lo que consiste en dividir el set de datos en 9 partes de igual tamaño y entrenar por separado estas partes, cada una de estas partes se testeó con el mismo set de prueba. Durante el entrenamiento del transfer learning se utiliza el callback ModelCheckpoint [30], así en cada iteración del K fold, este callback guarda los mejores pesos durante el entrenamiento, para luego escoger el de mayor accuracy en el set de prueba con un valor de 0.9334 en el conjunto de test lo que es un aumento significativo, la clasificación se observa en la figura 3.35. La pérdida tuvo un valor de 0.4152.

### 3.5 Score para evaluar detección de peaks

Los algoritmos son medidos con el Score usado en el desafío científico de ICHBI 2022[3], donde participaron científicos de varios países y el mejor Score presentado fue de 133. Como existe un desfase temporal entre el punto de máxima energía durante la contracción ventricular registrado en el ECG (Peak R) del ECG y el J del BCG, entre cada intervalo temporal de Peaks R o  $\Delta R_i$  debe existir un peak J, sin embargo el algoritmo puede fallar en algunos casos donde se detecta más de un peak o ningún peak. Estos tres casos se detallan a continuación para cuantificar la capacidad de detección de Peak J.

1. Existe un solo un peak  $J_j$  en el intervalo  $[R_i, R_{i+1}[$ .

$$Score_i = abs(\Delta R_i - \Delta J_j)$$

2. No existen peaks  $J_j$  en el intervalo  $[R_i, R_{i+1}[$ .

$$Score_i = abs(\Delta R_i)$$

3. Existen K peaks  $(J_j, J_{j+1}..J_{j+k})$  en el intervalo  $[R_i, R_{i+1}[$ .

$$Score_i = \sum_{k=1}^{k-1} abs(\Delta R_i - \Delta J_{j+k})$$

En resumen, a cada intervalo  $\Delta R_i$  le corresponde un Score. El caso uno es el ideal y aporta poco Score que corresponde al valor absoluto de la diferencia entre  $\Delta R_i$  y  $\Delta J_j$ , el caso dos aporta más Score que el primero pues corresponde solo al valor absoluto del intervalo  $\Delta R_i$ , el tercer caso igual aporta más

score que el primero. Según esta lógica mientras mayor sea el Score el desempeño del algoritmo detector de peaks es más bajo.

El Score total de la señal se calcula como el promedio de todos los Scores calculados para cada intervalo  $\Delta R_i$ .

$$TotalScore = \frac{1}{n-1} \sum_{n=1}^{n-1} Score_i$$

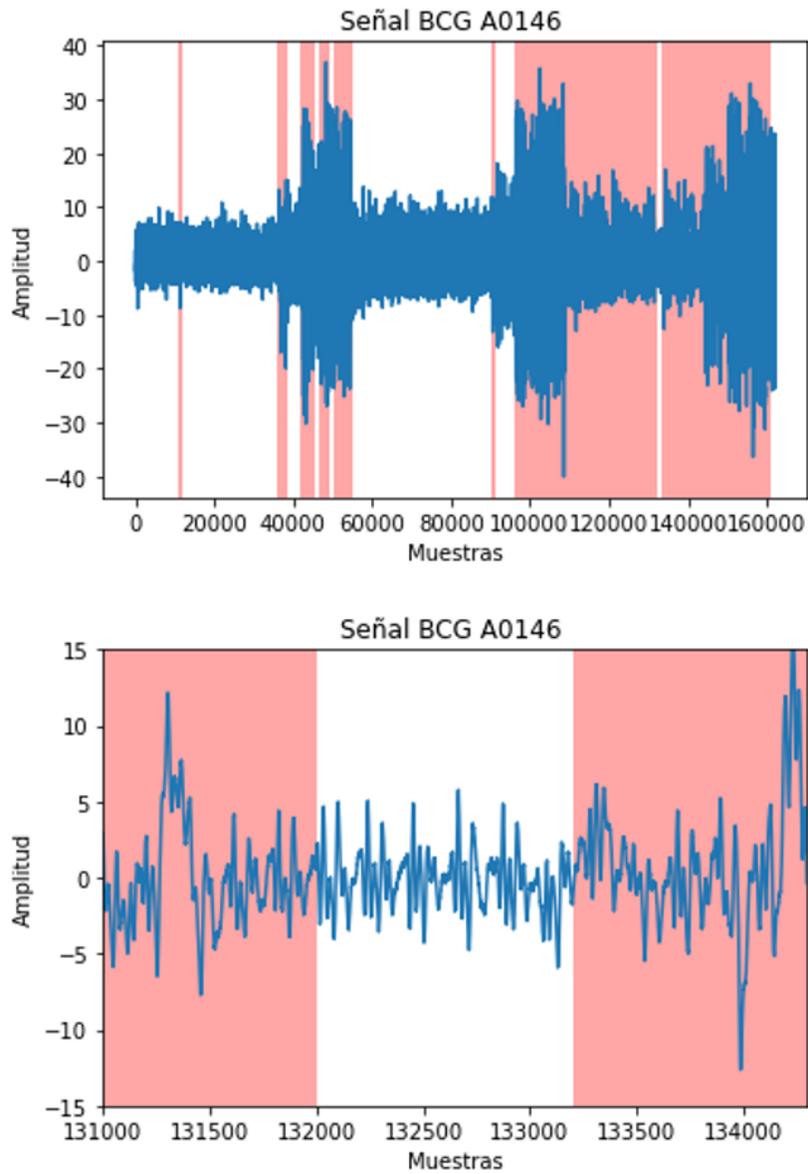
El Score global de la carpeta testing corresponde al promedio de los Total Score calculados para cada señal de la carpeta testing.

Para calcular el Score del detector de peaks en conjunto a la CNN de clasificación de segmentos limpios se exporto un archivo desde el script de la CNN que contiene un vector con los tiempos en ms donde la señal es clasificada como 1 o limpia. Luego en el script del Score se importa el archivo y se verifica que los peaks de los intervalos  $\Delta J_j$  y  $\Delta R_i$  están solamente dentro de los tiempos limpios (para el caso de los Peak R se desplaza los tiempos 30 muestras hacia atrás debido al intervalo R-J), si un peak no se encuentra dentro del vector de tiempo del archivo se procede a eliminar el Score correspondiente a ese peak  $R_i$ . Aquí hay que considerar el siguiente detalle debido a que los tiempos del archivo de salida de la CNN solo contiene tiempos clasificados como 1 mientras los segmentos ruidosos no son contabilizados, por lo que hay saltos que dependen de la longitud temporal del ruido provocando que en los bordes de los segmentos limpios el intervalo  $\Delta R_i$  tenga un valor mayor a la longitud temporal del ruido debido a que el peak  $R_i$  se encuentra en un tiempo anterior al segmento ruidoso eliminado y el peak  $R_{i+1}$  se encuentra en un tiempo que esta después del segmento con ruido, lo que aporta mucho Score, entonces también se deben eliminar estos Scores que se encuentran en los bordes de los segmentos limpios.

### 3.6 Discusión

En este capítulo, el enfoque en la organización de la base de datos obtenida con BCG es crucial para el análisis adecuado y eficiente de las señales. La organización adecuada de los datos es esencial para evitar sesgos y garantizar la precisión de los resultados en cualquier estudio. El procesamiento de señales es una etapa crítica en el análisis del BCG, y el hecho de que el capítulo comience con una descripción detallada de los dos algoritmos detectores de peaks probados es muy apropiado. Los algoritmos de detección de peaks son fundamentales para identificar eventos relevantes en las señales y representan el punto de partida para la extracción de características y la clasificación posterior. La comparación y evaluación de varios algoritmos brinda una visión completa de su rendimiento y potencial para el procesamiento

de las señales BCG. La incorporación de una red neuronal convolucional para clasificar el ruido es una adición interesante al capítulo. Las CNN han demostrado ser poderosas en el procesamiento de señales y en la clasificación de patrones complejos. Al implementar una CNN específicamente para el propósito de clasificar el ruido en las señales BCG, se ofrece una solución prometedora y moderna para mejorar la calidad y precisión del análisis, además sigue un hilo del desarrollo de alumnos anteriores que facilita los hiperparámetros de la red neuronal. La evaluación de los algoritmos es un método confiable pues se ha utilizado en el desafío científico en el año 2022 de International Conference on Biomedical and Health Informatics.



**Figura 3.35:** Los segmentos marcados de la señal A0146 se clasifican como ruido por parte de la CNN, La imagen superior es una vista completa y la imagen inferior un zoom.

## Capítulo 4. Resultados y discusión

### 4.1 Resultados

Para mostrar los resultados se decide evaluar con el Score los algoritmos detectores de peaks Jpeaks, JpeaksCub y JpeaksCorr como se ve en la siguiente figura 4.36. Estos son evaluados en dos casos, el primer caso se evalúa solamente el rendimiento del detector de peaks mediante el Score, el segundo caso evalúa el rendimiento del detector de peaks en conjunto a la CNN que clasifica segmentos limpios. Los resultados están en la tabla 4.5.

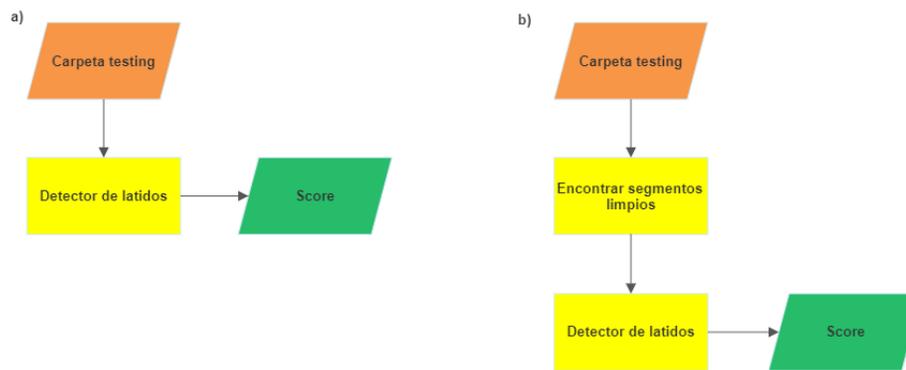


Figura 4.36: a) Primer caso. b) Segundo caso

También se hace un registro medio de todas las señales de la carpeta testing el cual está en la tabla 4.6, en este se calcula el promedio de peaks que ha encontrado el algoritmo detector de peaks, el promedio de los Peak J y el promedio de peaks R correctamente encontrados por los algoritmos.

Tabla 4.5: Scores de los scripts.

| Algoritmo   | Global Score sin CNN | Global Score con CNN |
|-------------|----------------------|----------------------|
| Jpeaks      | 196.57               | 127.72               |
| Jpeaks Cub  | 164.86               | 118.35               |
| Jpeaks Corr | 147.32               | 108.68               |

**Tabla 4.6:** La primera columna muestra el nombre del algoritmo, en la columna Total peaks se encuentra el número de peaks encontrados por el algoritmo (no todos son Peak J), la columna Peaks J muestra el número total de Peak J, la columna Peak R muestra el número de Peak R como referencia. Las últimas tres columnas siguen la misma lógica que las anteriores solamente que se evalúa en los segmentos limpios gracias a la CNN.

| Algoritmo   | Total peaks | Peaks J | Peaks R | Total p. + CNN | Peaks J + CNN | Peaks R + CNN |
|-------------|-------------|---------|---------|----------------|---------------|---------------|
| Jpeaks      | 544         | 516     | 583     | 407            | 393           | 421           |
| Jpeaks Cub  | 585         | 533     | 583     | 422            | 397           | 421           |
| Jpeaks Corr | 641         | 556     | 583     | 455            | 410           | 421           |

## 4.2 Discusión

Según la tabla 4.5 la CNN mejora en gran cantidad los Scores de todos los scripts, y el mejor Score corresponde a Jpeaks Corr. Este método es más sofisticado debido a la correlación de Pearson reflejándose en que para procesar 1061 minutos entre las 51 señales de la carpeta testing demora aproximadamente 80[s], a diferencia de los scripts anteriores donde Jpeaks se demora aproximadamente 15[s] y Jpeaks Cub 20[s].

Al interpretar la tabla 4.6 se observa que existe un total de 583 Peak R por lo tanto debería existir 583 Peak J. Jpeaks Cub detecta un total de 585 peaks de los cuales 516 son verdaderos positivos, Jpeaks Corr detecta un total de 641 peaks de los cuales 556 son verdaderos positivos y Jpeaks detecta un total de 544 peaks donde 516 son verdaderos positivos.

Al comparar ambas tablas se observa que Jpeaks Corr obtuvo el mejor Score y es el que detecta mayor cantidad de verdaderos positivos (556) y falsos positivos (641), también se observa que Jpeaks obtuvo el Score más bajo y es el único que detecta menos peaks (544) que la referencia de 583 Peak R. Jpeaks Cub detecta 585 peaks siendo el más cercano a la referencia sin embargo solo 516 son verdaderos positivos, este comportamiento indica que el Score aumenta más si faltan peaks por detectar que cuando se han detectado peaks adicionales.

Cuando se utiliza la CNN hay mejoras respecto a algunas proporciones. Por ejemplo, en la predictividad positiva que mide la proporción entre Peak J y el total de peaks detectados, se observa que Jpeaks Corr aumenta de 0.867 a 0.901, Jpeaks Cub aumenta de 0.911 a 0.941 y Jpeaks de 0.948 a 0.965. Esto indica una disminución en la cantidad de falsos positivos al utilizar la CNN.

## Capítulo 5. Conclusiones

### 5.1 Discusión

Comprender la historia y evolución del análisis de señales BCG es esencial para apreciar cómo los avances tecnológicos han influido en términos de espacio y funcionalidad en los sistemas de detección. Esta revisión histórica permite dimensionar y comparar sistemas antiguos con enfoques más modernos, sentando las bases para un análisis más profundo y actual.

El análisis de señales BCG se vuelve más completo al abordar los conceptos relacionados con esta señal biológica. Esta base teórica es fundamental para comprender la aplicabilidad y utilidad del BCG en el contexto de la salud cardiovascular. La introducción de redes neuronales y otros métodos modernos en el procesamiento de señales BCG brinda una perspectiva actualizada, demostrando cómo estas herramientas han demostrado ser poderosas para resolver problemas complejos en diversas disciplinas.

En el procesamiento de señales BCG, se destaca la relevancia de una estructuración adecuada de la base de datos para evitar sesgos y garantizar resultados precisos. Los algoritmos de detección de peaks juegan un papel fundamental en la identificación de eventos relevantes en las señales BCG, lo que permite una extracción precisa de características en la contracción ventricular del corazón. La introducción de una CNN para clasificar el ruido en las señales BCG amplía aún más las capacidades de análisis, al abordar patrones complejos y mejorar la calidad de las mediciones.

La evaluación y comparación de algoritmos y la CNN aportan una visión práctica y cuantitativa del rendimiento en la detección de peaks. Se observa cómo la implementación de la CNN mejora los puntajes y reduce la proporción de falsos positivos en todos los algoritmos de detección de peaks, lo que indica un progreso prometedor en el análisis de señales BCG.

### 5.2 Conclusiones

Después de implementar diversos algoritmos detectores de peaks y comparar sus resultados, se puede concluir que las técnicas de reconocimiento de patrones son capaces de detectar una mayor cantidad de Peaks J. Además, se observa que a medida que el rendimiento del algoritmo aumenta, también aumenta el tiempo de ejecución necesario. Sin embargo, es importante destacar que esta tendencia puede variar

según el tipo de algoritmo utilizado.

En relación con la Red Neuronal Convolutiva de clasificación binaria, se llega a la conclusión de que resulta de gran utilidad al combinarla con los detectores de peaks, pues se logra mejorar tanto el Score como la predictividad positiva de todos los algoritmos de detección de peaks. Esto pone de manifiesto que las mediciones se vuelven más confiables al emplear esta red neuronal. No obstante, al igual que muchas otras redes neuronales, esta también presenta un sesgo asociado a los datos de entrenamiento. Dicho sesgo depende de diversos factores, siendo uno importante el tipo de pacientes registrados en la base de datos. Estos pacientes pueden ser categorizados según edad, índice de masa corporal, hábitos alimenticios y diversos estados de salud. Por ende, es crucial considerar estos factores al utilizar la red y al registrar nuevas señales para formar una base de datos lo más equilibrada y representativa posible de una población diversa.

Si bien los avances han demostrado que la CNN mejora la calidad de las mediciones, aún no es lo suficientemente confiable para implementarse en un sistema Home-Care. No obstante, ampliar la base de datos para obtener mayor cantidad de MA's, ya sea mediante datos sintéticos o nuevos registros, tiene el potencial de aumentar la precisión de esta red neuronal.

### **5.3 Trabajo Futuro**

Ha quedado en evidencia la utilidad de las redes neuronales, por esto se propone hacer la detección de peaks mediante estas mismas, lo que podría evitar problemas de algoritmos convencionales donde muchas veces se necesita escoger un umbral para diferenciar dos casos lo cual no necesariamente aplica a conveniencia nuestra debido a la variada morfología del BCG.

Se puede realizar la detección de fibrilación auricular si las etiquetas de clasificación para entrenar la CNN son escogidas de manera más rigurosa, escogiendo al segmento de clase limpia cuando solo tiene un Peak J entre el intervalo R. Este proceso ajusta la red neuronal a la capacidad del algoritmo detector de peaks sin embargo asegura que en el segmento clasificado como limpio la detección de Peak J es intachable, por lo que un aumento repentino de la frecuencia cardíaca indicaría un posible evento de fibrilación auricular con altísima exactitud.

Adicionalmente se puede crear otra CNN similar a los autoencoders que sea capaz de reconstruir algunos segmentos de la señal clasificados como ruido por parte de la CNN. Para realizar esto se necesita subclasificar el ruido en reconstruible e irreconstruible, luego los segmentos reconstruibles deben

ser comparados con los peaks de su ECG respectivo y ser pasados por un algoritmo que modifique el segmento BCG original según la referencia en el ECG creando un segmento sintético reconstruido, esto permite más tiempo de monitoreo.

Finalmente puede estimarse el gasto cardiaco del paciente, para lograrlo se debe realizar un estudio integrando la forma de onda de la señal BCG con datos específicos del paciente como altura, edad, masa, sexo, enfermedades, etc.. y compararlos con alguna referencia del gasto cardiaco, ya sea con el método de Flick o con el ecocardiograma. Luego se entrena una red neuronal donde la entrada son los datos específicos del paciente más su señal BCG y la salida el valor del gasto cardiaco, el entrenamiento de esta red crea automáticamente el algoritmo para estimar el gasto cardiaco, sin embargo, siempre hay que tener en consideración la cantidad y calidad de datos a utilizar.

## Glosario

**BCG** Balistocardiograma

**CNN** Red Neuronal Convolutacional (Convolutional Neural Network)

**DNN** Red Neuronal Profunda (Deep Neural Network)

**DWT** Transformada Wavelet Discreta (Discrete Wavelet Transform)

**ECG** Electrocardiograma

**EM** artefacto por Movimiento de Electrodo

**EMFi** Película Electromecánica (Electromechanical Film)

**ICBHI** Conferencia Internacional en Informática Biomédica y Salud (International Conference on Bio-medical and Health)

**IPG** Pletismografía de Impedancia

**MAs** Artefactos por Movimiento

**Peak J** punto de máxima energía durante la contracción ventricular registrado en el BCG

**Peak R** punto de máxima energía durante la contracción ventricular registrado en el ECG

**PVDF** Sensor Piezoeléctrico (Piezoelectric Sensor)

**SNR** Razón Señal Ruido (Signal Noise Reason)

## Referencias

- [1] P. Cossio y H. Mosso, “El balistocardiograma normal,” [En línea]. Disponible en: <https://www.scribd.com/document/433970828/Balistocardiograma>.
- [2] J. Valle, “Una breve historia de la electrocardiografía,” p. 17, 2001, [En línea]. Disponible en: <https://enfermeriaencardiologia.com/wp-content/uploads/22histelectro.pdf>, Fecha de acceso 24 de julio 2023.
- [3] ICBHI, “Scientific challenge,” [En línea]. Disponible en: <https://www.icbhi2022.com/call-for-sc/>.
- [4] L. I. Flores Nuñez, “Desarrollo de algoritmos para la detección automática y análisis del complejo ijk del bcg y su relación con el ecg en sujetos en posiciones sedente y decúbito supino,” Master’s thesis, 2022.
- [5] I. Starr y E. Hildreth, “The effect of aging and of the development of disease on the ballistocardiogram,” *Circulation*, vol. 5, no. 4, pp. 493–495, 1952, [En línea]. Disponible en: <https://www.ahajournals.org/doi/abs/10.1161/01.CIR.5.4.481>.
- [6] B. van Rooij, K. Tavakolian, S. Arzanpour, A. Blaber, y C. Leguy, “Non-invasive estimation of cardiovascular parameters using ballistocardiography,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 1247–1250.
- [7] D. Yazdi y otros, “Noninvasive scale measurement of stroke volume and cardiac output compared with the direct fick method: A feasibility study,” *Journal of the American Heart Association*, vol. 10, no. 24, pp. 2–7, 2021.
- [8] F. I. y C. Saavedra, “Redes neuronales artificiales,” *Departamento de Física, Universidad de Concepción Chile*, p. 7, 2000.
- [9] Wikipedia, “Red neuronal artificial,” 2017, pp. 46-60, [En línea] Disponible en: [https://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](https://es.wikipedia.org/wiki/Red_neuronal_artificial), Fecha de acceso: 31 de julio 2023.
- [10] J. P. Moreno del Solar, “Red neuronal profunda para detección de artefactos por movimiento en señales de balistocardiografía.” Tesis de Magíster, Ciencias de la Ingeniería con mención en Ingeniería Eléctrica, Departamento de Ingeniería Eléctrica, Universidad de Concepción, 2019.

- [11] F. Chollet, *Deep Learning with Python*. Manning Publications Co, 2018, pp. 46–60, pp. 46-60, [En línea] Disponible en: <https://docs.google.com/viewer?a=v&pid=sites&srcid=dW10LmVkdS5wa3xzbxwGd4Ojc1ODc1ODY2OTZiOTUzOGQ>, Fecha de acceso: 3/8/2023.
- [12] S. Haykin, *Neural Networks and Learning Machines*. Uniba, 2009, p. 11, [En línea]. Disponible en: <https://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf>.
- [13] F. Chollet, *Deep Learning with Python*. Manning Publications Co, 2018, p. 126, pp. 46-60, [En línea] Disponible en: <https://docs.google.com/viewer?a=v&pid=sites&srcid=dW10LmVkdS5wa3xzbxwGd4Ojc1ODc1ODY2OTZiOTUzOGQ>, Fecha de acceso: 3/8/2023.
- [14] F. Chollet, *Deep Learning with Python*. Manning Publications Co, 2018, p. 122, pp. 46-60, [En línea] Disponible en: <https://docs.google.com/viewer?a=v&pid=sites&srcid=dW10LmVkdS5wa3xzbxwGd4Ojc1ODc1ODY2OTZiOTUzOGQ>, Fecha de acceso: 3/8/2023.
- [15] J. Chávez, “Non-invasive monitoring system of physiological variables,” Tesis de Magíster, Ciencias de la Ingeniería con mención en Ingeniería Eléctrica, Departamento de Ingeniería Eléctrica, Universidad de Concepción, Marzo 2017.
- [16] E. J. Pino, J. A. P. Chávez, y P. Aqueveque, “Noninvasive ambulatory measurement system of cardiac activity,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 7622–7625.
- [17] S. Rajala y J. Lekkala, “Film-type sensor materials pvdf and emfi in measurement of cardiorespiratory signals— a review,” *IEEE Sensors Journal*, vol. 12, no. 3, p. 442, 2012.
- [18] B. Larrotta y D. María,, “Aplicación de la transformada wavelet discreta en el filtrado de señales bioeléctricas,” *Umbral Científico*, no. 5, pp. 92–98, 2004.
- [19] “Wavelet toolbox,” [En línea]. Disponible en: <https://la.mathworks.com/products/wavelet.html>, Fecha de acceso: 24 de Julio del 2023.
- [20] Z. T. y otros, “Quantitative ballistocardiography (q-bcg) for measurement of cardiovascular dynamics,” vol. 60, p. 618, 2011.
- [21] pywavelets, “Navegador wavelet por pywavelets,” [En línea]. Disponible en: <https://wavelets.pybytes.com/wavelet/db6/>, fecha de acceso: 23 de julio de 2023.
- [22] I. Sadek y B. Abdulrazak, “A comparison of three heart rate detection algorithms over ballistocardiogram signals,” *Biomedical Signal Processing and Control*, vol. 70, p. 103017, 2021.

- [23] G. Cathelain, B. Rivet, S. Achard, J. Bergounioux, y F. Jouen, “Dynamic time warping for heartbeat detection in ballistocardiography,” in *2019 Computing in Cardiology (CinC)*, 2019, pp. 1–4.
- [24] H.-T. Chiang y otros, “Noise reduction in ecg signals using fully convolutional denoising autoencoders,” *IEEE Access*, vol. 7, pp. 60 806–60 813, 2019.
- [25] F. Studnicka. (2022) Mendeley data, v1., [En línea]. Disponible en: <https://data.mendeley.com/datasets/8yzmk4dd7p/1>, doi=10.17632/8yzmk4dd7p.1. BCG data base, Fecha de acceso : 24 de julio del 2023.
- [26] Keras, “Dropout layer,” [En línea]. Disponible en: [https://keras.io/api/layers/regularization\\_layers/dropout/](https://keras.io/api/layers/regularization_layers/dropout/), Fecha de acceso: 22 de julio de 2023.
- [27] Keras, “Callbacks api,” <https://keras.io/api/callbacks/>, Fecha de acceso: 22 de julio de 2023.
- [28] Keras, “Earlystopping,” [En línea]. Disponible en: [https://keras.io/api/callbacks/early\\_stopping/](https://keras.io/api/callbacks/early_stopping/), Fecha de acceso: 22 de julio de 2023.
- [29] Keras, “Transfer learning fine-tuning,” [En línea]. Disponible en: [https://keras.io/guides/transfer\\_learning](https://keras.io/guides/transfer_learning), 2020, Fecha de acceso: 22 de julio de 2023.
- [30] Keras, “Modelcheckpoint,” [En línea]. Disponible en: [https://keras.io/api/callbacks/model\\_checkpoint/](https://keras.io/api/callbacks/model_checkpoint/), Fecha de acceso: 22 de julio de 2023.

**UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA  
RESUMEN DE MEMORIA DE TITULO**

|                                      |  |
|--------------------------------------|--|
| <b>Departamento</b>                  | : Departamento de Ingeniería Civil Eléctrica   |
| <b>Carrera</b>                       | : Ingeniería Civil Biomédica   |
| <b>Nombre del memorista</b>          | : Miguel Angel Sepúlveda González  |
| <b>Título de la memoria</b>          | : Comparación de algoritmos detectores de latidos en señales BCG con y sin incorporación de red neuronal convolucional |
| <b>Fecha de la presentación oral</b> | : 30 de Agosto del 2023  |
| <b>Profesor(es) Guía</b>             | : Esteban Pino Quiroga   |
| <b>Profesor(es) Revisor(es)</b>      | : Miguel Figueroa y Álvaro Saldaña   |
| <b>Concepto</b>                      | :  |
| <b>Calificación</b>                  | :  |

| <b>Resumen</b> |
|----------------|
|----------------|

Este trabajo presenta un avance innovador para la detección de latidos y adaptación el BCG en hogares y sistemas Home-Care, lo que puede mejorar significativamente la calidad de vida de los pacientes. Se implementaron tres algoritmos detectores de peaks, uno de los cuales utiliza técnicas de reconocimiento de patrones como template matching, con el cual se obtuvo un Score igual a 147. Además, se creó una Red Neuronal Convolucional (Convolutional Neural Network, CNN) especializada para clasificar el ruido y los segmentos limpios, lo que permite cancelar la detección de latidos en zonas de ruido.

Al combinar la detección de peaks y la CNN, el Score mejora a 108. La utilización de la red neuronal mejora notablemente la detección de latidos, lo que permite desarrollar un sistema de medición más confiable para su uso en hogares y sistemas Home-Care. Como referencia, el ganador de la competencia obtuvo un puntaje de 133 puntos. Se evidencia la utilidad de la red neuronal al integrarse junto al detector de latidos.