Name : mohamed halawa

ID:20100315

project name: Biological Signal Processing for Cardiovascular Risk Stratification

company name: El Alamein International University, Medical Department

mentor name: Ahmed EL-Yazbi

Abstract:

Cardiovascular diseases remain a leading cause of morbidity and mortality worldwide, necessitating innovative approaches for early risk stratification and intervention. In pursuit of this goal, we present a research project that merges biological signal processing with cutting-edge machine learning techniques. Our primary objective is to discriminate between healthy and at-risk individuals by analyzing continuous blood pressure recordings.

To achieve this, we employ advanced signal processing methods, including Short Fourier and Wavelet Transforms, which enable us to dissect the temporal and frequency characteristics of the blood pressure time series. The derived scalograms serve as rich visual representations of the data, akin to images, facilitating the application of Convolutional Neural Networks (CNNs) for classification.

Our project draws from extensive datasets collected from both experimental animals and humans. These datasets are meticulously curated and labeled, ensuring the quality of training data for our machine learning model. We emphasize ethical considerations, especially when handling data from human subjects, and comply with all necessary approvals and regulations.

The heart of our research lies in model optimization. It involves fine-tuning the CNN architecture, selecting appropriate hyperparameters, and addressing overfitting concerns. Additionally, data augmentation techniques are explored to enhance dataset diversity, improving model generalization.

Introduction:

Cardiovascular diseases (CVDs) continue to be a leading cause of mortality and morbidity worldwide. Early risk stratification and intervention are crucial to improving patient outcomes and reducing the burden of CVDs. To address this challenge, we embark on a research journey that combines the power of biological signal processing and machine learning to develop a novel approach for cardiovascular risk stratification.

Continuous blood pressure recordings offer a valuable source of physiological information. These recordings, when analyzed over time, hold the potential to reveal subtle but critical patterns and variations that may signify underlying cardiovascular issues. However, extracting meaningful insights from these time series data requires sophisticated signal processing techniques and robust classification models.

This project sets out to leverage two key signal processing methods: Short Fourier Transform and Wavelet Transform. These techniques enable us to dissect the blood pressure time series, revealing both their frequency and temporal characteristics. The result is a set of scalograms, which serve as high-dimensional data representations of the underlying signals.

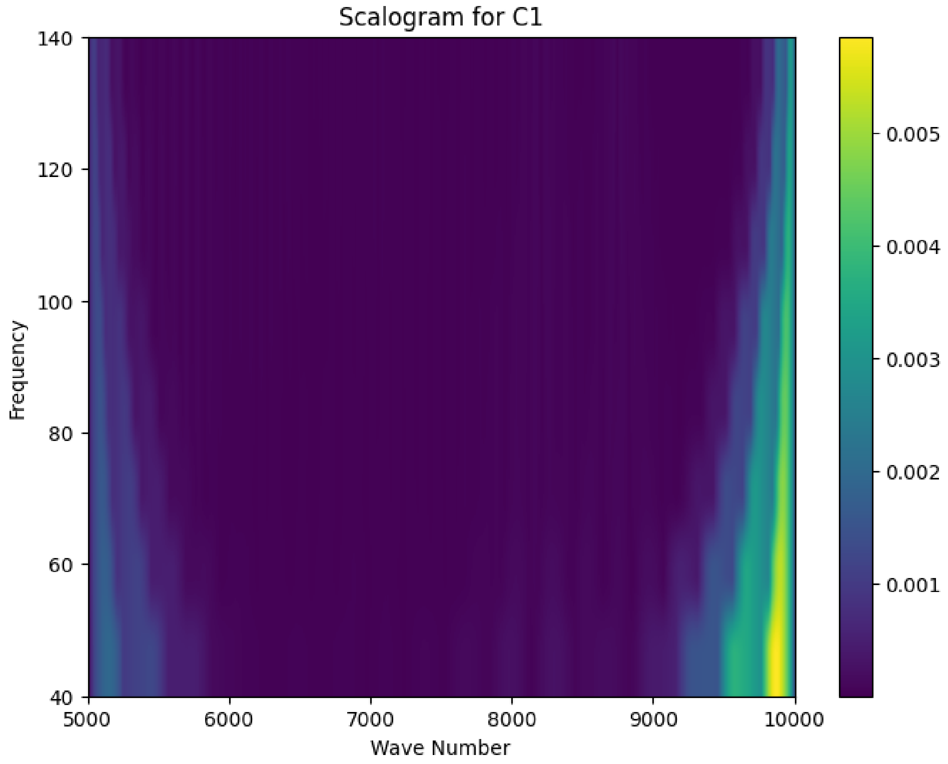
To transform these scalograms into actionable insights, we employ Convolutional Neural Networks (CNNs), a deep learning architecture renowned for its prowess in image analysis. By treating scalograms as visual inputs, we train a CNN model to distinguish between healthy and at-risk subjects, thus enabling early risk stratification.

The heart of this research lies in model optimization. Several critical steps, including architecture selection, hyperparameter tuning, and addressing overfitting, are undertaken to ensure the model's robustness and accuracy. Furthermore, data augmentation techniques are explored to diversify the dataset, promoting the generalization of the model to real-world scenarios.

The datasets utilized in this research originate from both experimental animals and human subjects. These datasets are meticulously collected, labeled, and curated to ensure their quality and reliability. Ethical considerations are paramount, especially when dealing with human subject data, and all necessary approvals and safeguards are adhered to.

Project steps:

1.Slicing spectrograms:



In this section, we delve deeper into the intricate process of slicing spectrograms and the application of the Wavelet Transform. These techniques are instrumental in our journey to unravel the complexities of time-series data and to glean insights into the behavior of specific frequencies within intricate signals.

Data Representation with Wavelet Transform:

The Versatile Wavelet Transform: The Wavelet Transform, an indispensable mathematical tool, serves as the backbone of our data analysis. It possesses the unique ability to dissect a signal into its constituent components in both the time and frequency domains. This dissection is accomplished through the utilization of a fundamental element: the wavelet function.

Wavelet Transform on Each Data Point: An inherent strength of the Wavelet Transform lies in its applicability to individual data points within a signal. This granular approach enables us to generate a comprehensive set of coefficients representing diverse frequencies and levels of detail. This is achieved by convolving the wavelet function with the input signal at different positions.

Resulting Coefficients: The outcome of this process is a collection of coefficients that illuminate specific frequencies present in the signal at distinct moments in time. These coefficients serve as our windows into the underlying patterns and variations within the data.

Mathematical Equation for Wavelet Transform:

The cornerstone of Wavelet Transform, the Continuous Wavelet Transform (CWT), is governed by the following mathematical equation:

$CWT(a, b) = \int x(t) * \psi^*[(t - b) / a] \, dt$

Here's a detailed breakdown of the equation components:

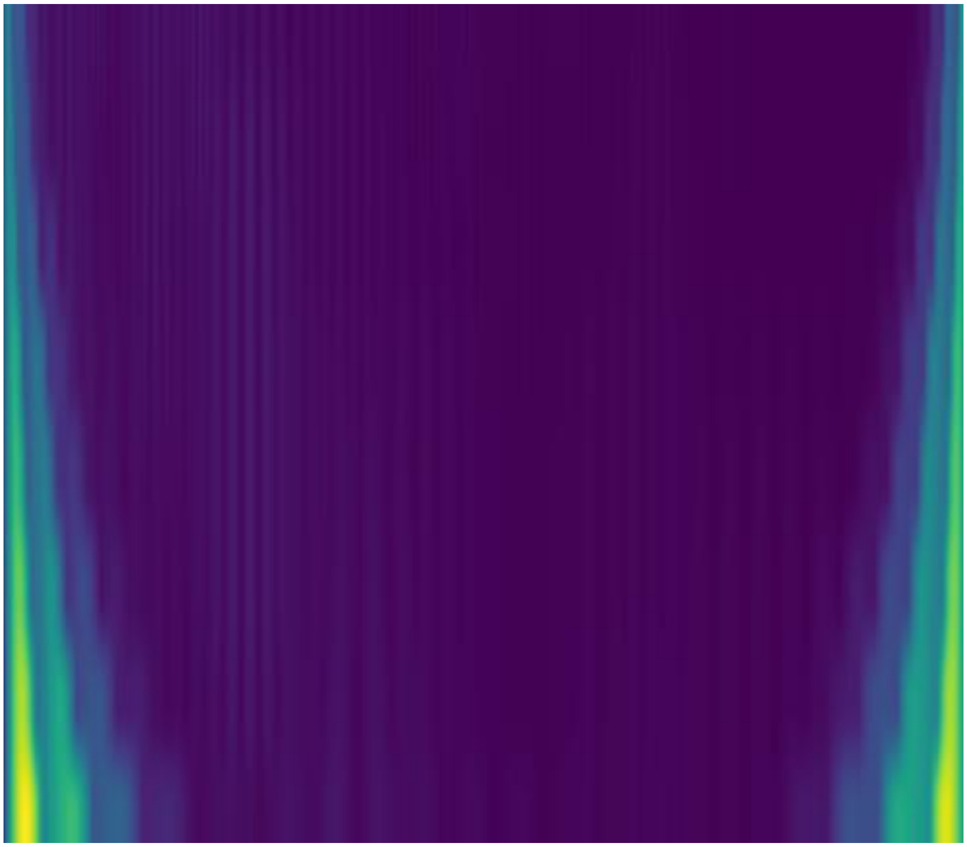
$CWT(a, b)$: This expression represents the transformation coefficient at a specific time (t) using a wavelet function characterized by parameters (a) and (b).

$x(t)$: Denotes the original time-domain signal under examination.

ψ: Signifies the wavelet function employed during the transformation. The choice of wavelet function influences the scale and characteristics of the analysis.

a and b: These parameters play a pivotal role in shaping the analysis by determining the time and frequency ranges for each analysis window

2.Cropping all Spectrograms:



In conjunction with the previously outlined techniques of slicing spectrograms and employing the Wavelet Transform, the process of cropping spectrograms emerges as an essential component of our data preprocessing pipeline. The task of cropping is a dynamic and strategic maneuver that empowers us to zero in on specific regions of interest within the spectrograms. By doing so, we enhance the precision and efficacy of our analysis.

The Importance of Cropping:

Focused Analysis: Spectrograms, often capturing a broad spectrum of frequency and temporal information, can be intricate and overwhelming. Cropping sharpens our focus on select features or patterns that bear the utmost relevance to our research goals. This precision can significantly enhance the accuracy of classification and the depth of our analysis.

Noise Mitigation: The judicious removal of redundant or noisy elements within the spectrograms, a process intrinsic to cropping, functions as a powerful noise-reduction mechanism. This practice is especially advantageous when dealing with intricate, noisy datasets, safeguarding the integrity of our analysis.

Cropping Techniques:

Cropping spectrograms encompasses the meticulous selection and retention of specific regions within each spectrogram. Various techniques can be employed for this purpose, including:

Fixed-Size Cropping: A well-defined, fixed window size is employed, positioned strategically within each spectrogram. All information residing within this window is preserved, while extraneous data is omitted.

Adaptive Cropping: Adaptive cropping methodologies leverage data-driven criteria to identify the optimal region for retention. This may encompass the identification of dominant frequency components or distinctive patterns that signal the presence of key features.

Harmonization with Wavelet Transform:

The process of cropping spectrograms is seamlessly integrated with the Wavelet Transform phase. Once we have isolated regions of interest within the spectrograms, the subsequent application of the Wavelet Transform is concentrated on these selected areas. This dynamic integration allows us to maximize the relevance of our analysis and further enhances the precision of our research.

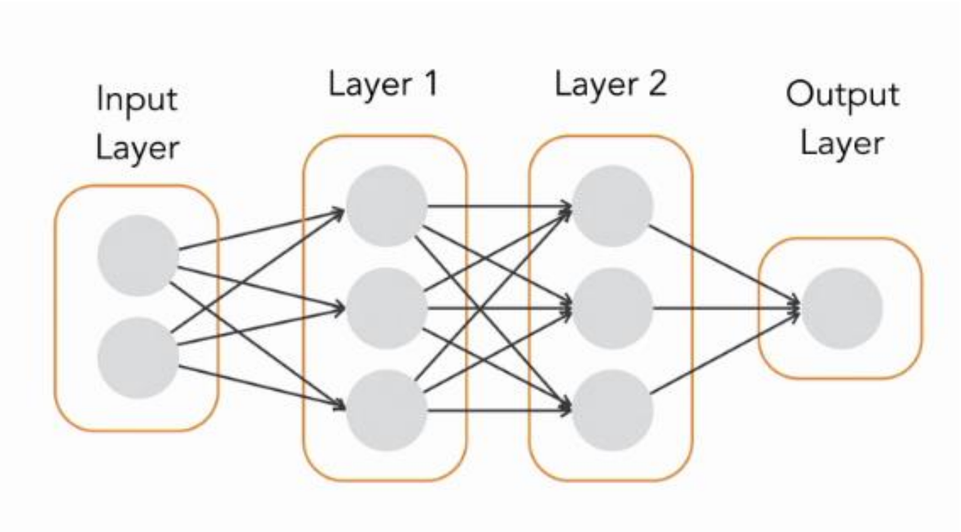
Challenges and Strategic Considerations:

While cropping spectrograms is invaluable, it is not without its challenges and strategic considerations. These include:

Potential Information Loss: The process of cropping, when overly aggressive, can result in the loss of valuable information. Striking an optimal balance between precision and comprehensive insight is paramount.

Cropping Criteria: The criteria employed for cropping should be thoughtfully chosen to align with the research objectives. The efficacy of cropping is intrinsically tied to the accuracy of these criteria.

3.Model CNN



Input Layer

The input layer receives the blood pressure scalograms. The size of the input layer is (IMAGE_HEIGHT, IMAGE_WIDTH, N_CHANNELS), where:

- IMAGE_HEIGHT is the height of the scalograms
- IMAGE_WIDTH is the width of the scalograms
- N_CHANNELS is the number of channels in the scalograms (usually 3 for color images)

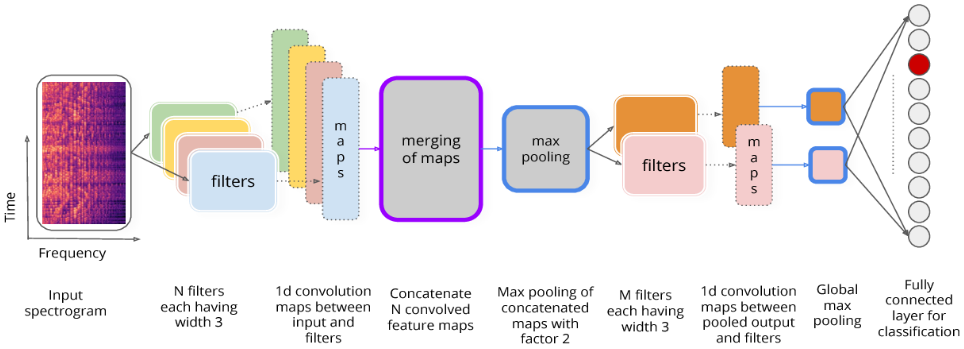
Convolutional Layers

The convolutional layers extract local features from the scalograms. There are three convolutional layers in the provided architecture:

1. Conv2D(32, 3, strides=2, padding='same', activation='relu')
- This layer has 32 filters, each with a kernel size of 3x3.
- The strides parameter is set to 2, which means that the output feature maps will be half the size of the input feature maps.
- The padding parameter is set to 'same', which means that the output feature maps will be the same size as the input feature maps.
- The activation function is ReLU, which is a nonlinear function that introduces nonlinearities into the model.
2. Conv2D(64, 3, padding='same', activation='relu')
- This layer has 64 filters, each with a kernel size of 3x3.
- The padding parameter is set to 'same', which means that the output feature maps will be the same size as the input feature maps.
- The activation function is ReLU.
3. Conv2D(128, 3, padding='same', activation='relu')
- This layer has 128 filters, each with a kernel size of 3x3.
- The padding parameter is set to 'same', which means that the output feature maps will be the same size as the input feature maps.
- The activation function is ReLU.

Pooling Layers

The pooling layers reduce the dimensionality of the data. There are three pooling layers in the provided architecture:


Time
Frequency
filters
maps
merging of maps
max pooling
filters
maps
Input spectrogram
N filters each having width 3
1d convolution maps between input and filters
Concatenate N convolved feature maps
Max pooling of concatenated maps with factor 2
M filters each having width 3
1d convolution maps between pooled output and filters
Global max pooling
Fully connected layer for classification

(a) *CNN Max Pooling Model*

4. MaxPooling2D(pool_size=(2, 2))

- This layer applies max pooling with a pool size of 2x2.
- Max pooling reduces the size of the feature maps by taking the maximum value within each 2x2 window.

5. MaxPooling2D(pool_size=(2, 2))

- This layer applies max pooling with a pool size of 2x2.

6. MaxPooling2D(pool_size=(2, 2))

- This layer applies max pooling with a pool size of 2x2.

Batch Normalization Layers

The batch normalization layers normalize the activations of the previous layers. There are six batch normalization layers in the provided architecture:

7. BatchNormalization()
8. BatchNormalization()
9. BatchNormalization()
10. BatchNormalization()
11. BatchNormalization()
12. BatchNormalization()

Batch normalization helps to improve the stability and performance of the model.

Fully Connected Layers

The fully connected layers learn high-level features from the previous layers. There are two fully connected layers in the provided architecture:

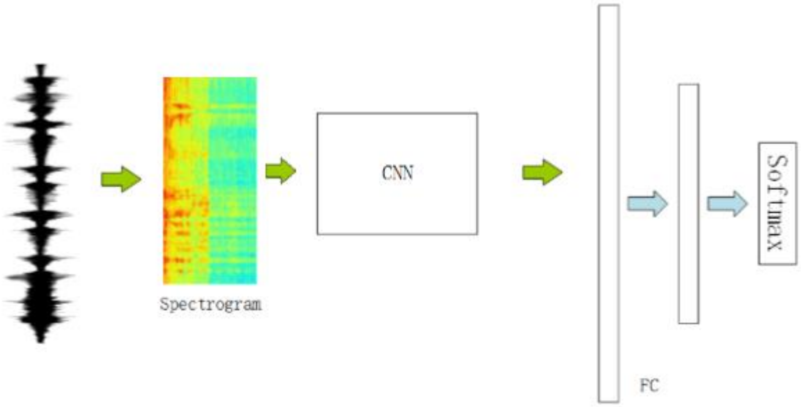
13. Dense(256, activation='relu')

- This layer has 256 neurons.
- The activation function is ReLU.

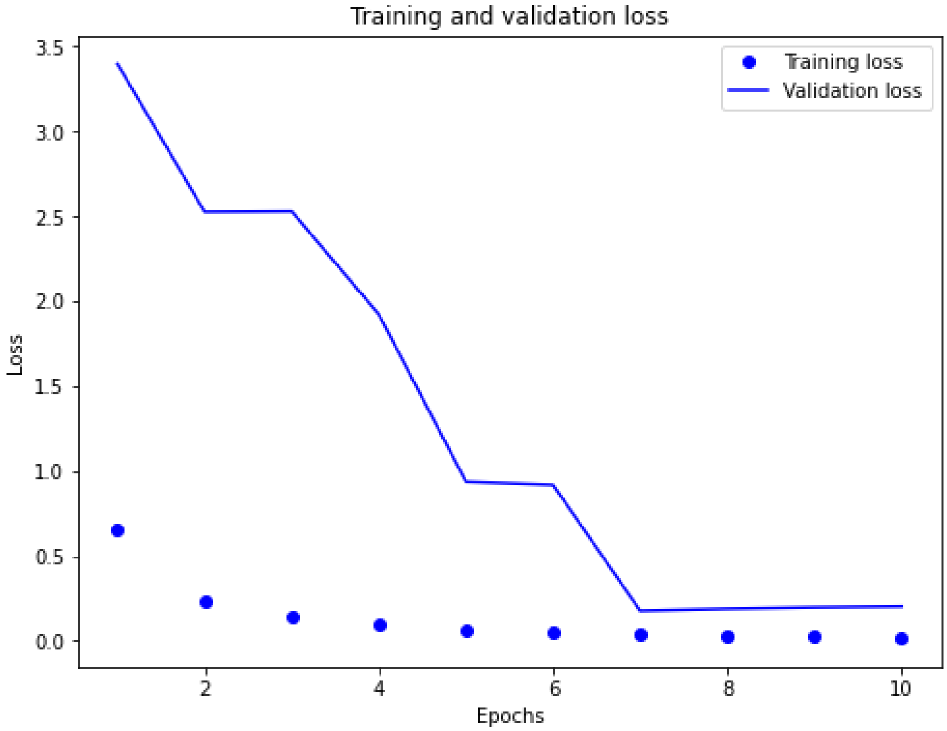
14. Dense(N_CLASSES, activation='softmax')

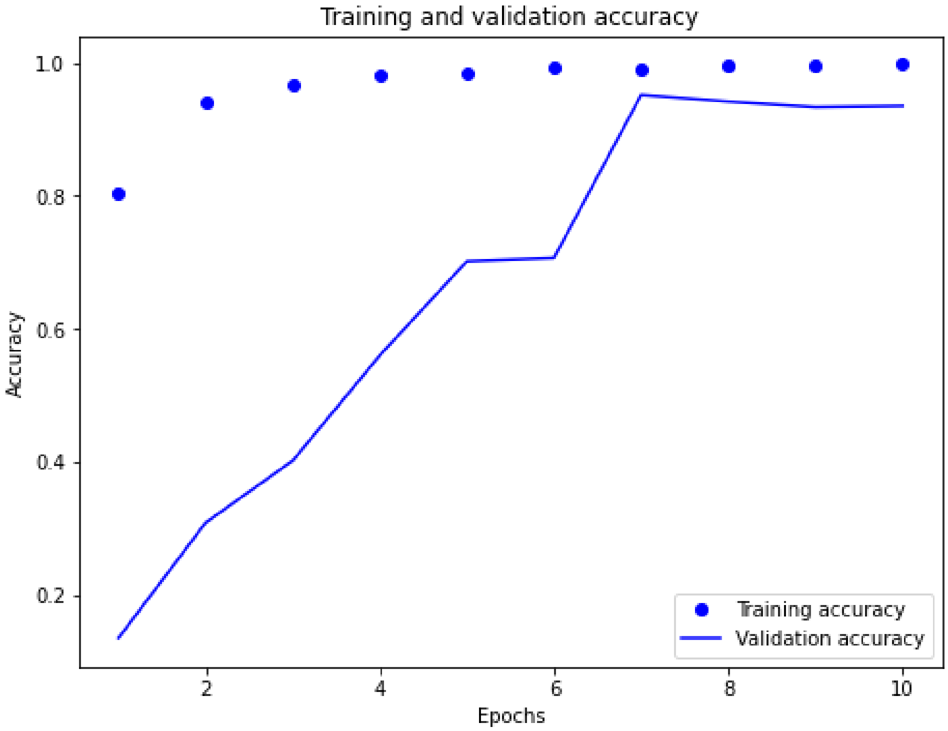
- This layer has N_CLASSES neurons, where N_CLASSES is the number of classes that the model is trained to classify (e.g., healthy or hypertensive).
- The activation function is softmax, which normalizes the output of the layer so that the values sum to 1.

Output Layer



The output layer produces the classification probabilities for each class. The provided architecture uses a softmax activation function for the output layer, which means that the output of the layer is a probability distribution over the N_CLASSES classes.




Training and validation accuracy
Accuracy
1.0
0.8
0.6
0.4
0.2
2
4
6
8
10
Epochs
Training accuracy
Validation accuracy