UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Neural and fuzzy computation course's project

Comparative Analysis of Machine Learning Techniques for Early Detection of Parkinson's Disease through Speech Pattern Analysis

Benedetti Matilde

Currao Giuseppe

ACADEMIC YEAR 2023

Chapter 1

Introduction

Parkinson's Disease (PD) is a progressive neurodegenerative disorder that primarily affects the motor system, resulting in tremors, bradykinesia, rigidity, and postural instability. PD is caused by the degeneration of dopaminergic neurons in the substantia nigra, leading to a deficiency of dopamine in the brain. While the exact etiology of PD remains elusive, multiple factors such as genetics, environmental toxins, and oxidative stress are thought to contribute to its development.

The accurate diagnosis of PD poses a significant challenge due to its complex clinical presentation, overlapping symptoms with other movement disorders, and the lack of definitive biomarkers. Misdiagnosis rates are particularly high in the early stages of the disease, which can have significant implications for patient management and treatment efficacy. Therefore, there is a pressing need for sensitive and specific diagnostic methods that can aid in early detection and intervention [6].

Recent studies have highlighted the potential of vocal dysfunction as a diagnostic marker for PD. Vocal abnormalities, including changes in pitch, loudness, and speech rhythm, have been observed in individuals with PD. Remarkably, these vocal impairments can manifest several years before the onset of other motor symptoms, making them valuable indicators for early detection [4].

Machine learning (ML) techniques, particularly deep learning models, have shown promise in the analysis of speech patterns for the detection and classification of PD. These models can extract intricate features from audio recordings and discern subtle changes in speech patterns that may be indicative of disease progression.

This promising field is leading to the birth of several studies on the matter. In [3], is presented an ensemble of convolutional neural networks (CNNs) for the detection of PD from the voice recordings. Convolutional Neural Networks (CNNs), a type of deep learning model, have proven successful in various audio-related tasks and exhibit great potential in PD detection. Other studies focused on an LSTM based approach [5]. Other promising results were also given by hybrid approaches utilizing a combination of Resonance based Sparse Signal Decomposition (RSSD) + Time-Frequency (T-F) algorithm to preprocess data before the giving it in input of a CNN [2]. Given the here discussed broad state of the art, in this study, it is proposed a

total of five distinct ML approaches to analyze speech patterns for accurate early-stage PD identification. These different methods will allow a complete analysis of Machine Learning behaviour in PD diagnosis.

The first approach involves developing a CNN from scratch and feed it directly with the audio recordings after appropriate signal processing, such as extracting Mel Spectrograms or Mel-Frequency Cepstral Coefficients (MFCCs). The second approach utilizes a CNN on a pretrained network, leveraging transfer learning to benefit from a network's prior knowledge and expertise.

Then a 1-D CNN was implemented. This is a variant of convolutional neural networks that operates on one-dimensional signals such as audio waveforms. This type of network can automatically learn relevant spatiotemporal filters within the audio signal, enabling an effective representation of discriminative features for PD diagnosis.

It follows an RNN. The latter is characterized by cyclic connections within its hidden layer, allowing it to process sequential data such as audio recordings. This architecture is particularly well-suited for analyzing temporal patterns in the context of voice and language.

Lastly, the LSTM is a type of RNN that utilizes a long-term memory structure to handle sequences of data with long-range dependencies. By retaining information over time, the LSTM can capture complex correlations within the audio recordings and enhance the classification capability of different vocal conditions.

By comparing the performance of the methods, we aim to determine the most effective approach for early-stage PD detection.

To conduct our analysis, we will utilize the Italian Parkinson's Voice and Speech dataset [1].

The comparative analysis of our proposed ML techniques will provide valuable insights into the efficacy of different approaches for early-stage PD detection.

Chapter 2

Dataset

The study is based on the analysis of the Italian Parkinson's Voice and Speech dataset Figure 2.1 which includes audio recordings from healthy individuals and patients with PD. It is composed of the audio recorded from fifteen healthy people aged 19-29 years, that give a reliable reference lower limit of the 'systematic' error rate for each word, twenty-two healthy persons aged 60-77 years (Healthy Elderly persons HEC), twenty-eight patients aged 40-80 years, affected by PD group. The audio file follow a protocol that comprehend two readings of a phonemically balanced text with a pause in between and the execution of different syllables and vowels.

The phonemically balanced text used in the protocol is a meaningful but challenging passage that assesses the patient's ability to pronounce difficult sounds and maintain breath control, allowing for an evaluation of neuro-control and speech intelligibility.

Moreover, the use of a phonemically balanced text helps standardize the assessment process, ensuring consistency in the speech samples collected. This also allows the generalization of the results since in this balanced text almost all the sounds of the Italian Language have been covered.

In conclusion, we can consider this dataset as an objective and complete assessment of speech impairment in PD.

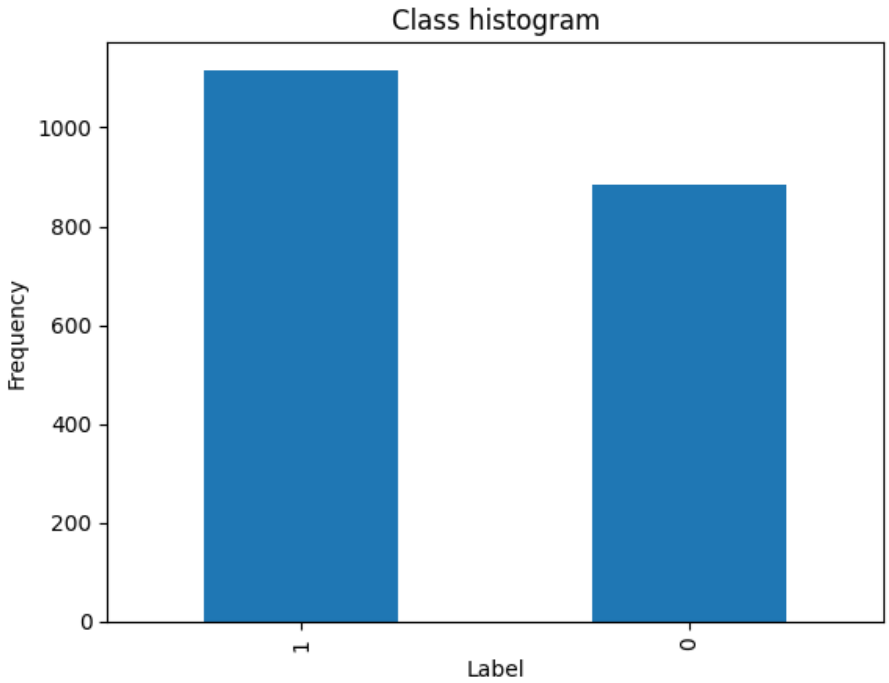

Figure 2.1: Histogram of the samples label. 1 = Parkinson Disease and 0 = Healty

Chapter 3

Data preprocessing

Data preprocessing involves the selection and extraction of relevant features from raw audio data. These features can then be fed into a machine learning model with the purpose of training a classifying network. There are several feature extraction techniques that have been proposed and used in the literature for speech pattern recognition [5]. We will use two of them: Mel Spectrograms and mel-frequency Cepstral Coefficients (MFCCs).

The feature extraction process is carried out using the Python Librosa library: it provides a wide range of functionality for working with audio data, including functions for loading audio files, feature extraction, and visualization. We extracted waveforms from the audio data with a sample rate of 44100 since is the optimal sampling rate for vocal audio data.

3.1 Mel Spectrogram

The Mel spectrogram is a valuable tool in vocal audio processing Figure 3.1. It represents audio signals in a way that aligns with human perception of frequency. By emphasizing perceptually relevant frequency components, it provides a more intuitive representation of vocal characteristics. The Mel spectrogram is used for voice activity detection, distinguishing speech from non-speech segments. It serves as a foundation for extracting vocal features such as Mel-frequency cepstral coefficients (MFCCs). The Mel spectrogram is widely used in machine learning models for tasks like speech recognition and voice conversion. This transformation gives a visual representation of the spectral characteristics of a speech signal where there is time on the x-axis, frequency on the y-axis and the color is proportional to the magnitude of the signal.

Overall, it enhances the understanding and analysis of vocal audio signals.

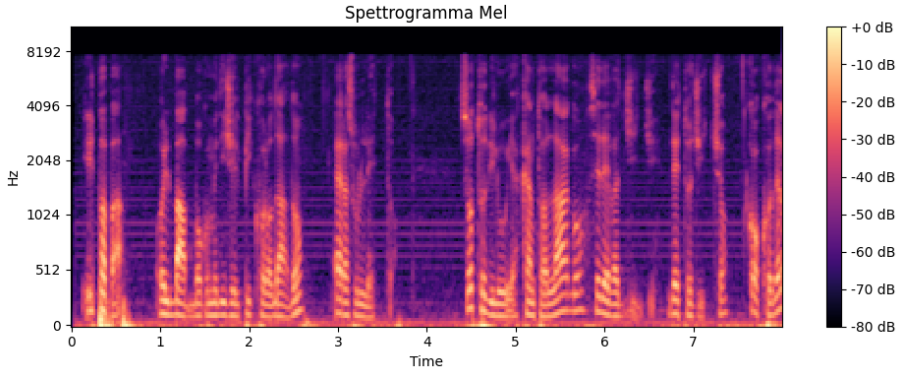

Figure 3.1: Mel spettrogram on one audio sample

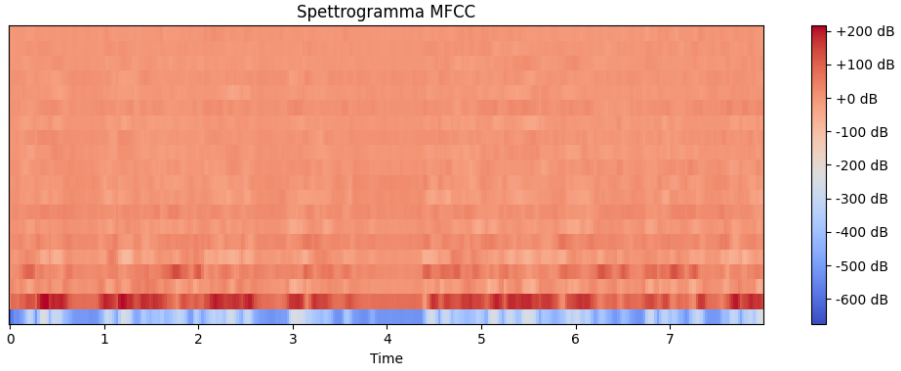

Figure 3.2: MFCCs spettrogram on one audio sample

3.2 MFCCs Coefficients

Mel-frequency cepstral coefficients (MFCCs) are a key feature in vocal audio preprocessing Figure 3.2. They are derived from the Mel spectrogram and capture important characteristics of the vocal signal. MFCCs represent the spectral envelope of the audio signal, emphasizing perceptually relevant frequency components while discarding less relevant ones.

MFCCs are significant in vocal audio preprocessing for several reasons. Firstly, they provide a compact representation of the spectral content of the signal, reducing the dimensionality of the data. This is crucial for efficient storage and processing of vocal audio in various applications.

Secondly, MFCCs are designed to mimic human auditory perception. By modeling the human cochlea's frequency resolution, they focus on the frequency bands that are most relevant for speech perception. This makes them suitable for tasks such as speech recognition and speaker identification.

Furthermore, MFCCs are robust to noise and variations in vocal conditions. They capture the distinctive features of the vocal signal while minimizing the influence of irrelevant factors like background noise or speaker characteristics.

3.3 Data segmentation

Data segmentation is a data augmentation technique commonly used in vocal data processing. It involves dividing longer audio sequences into smaller segments or frames. Each segment contains a smaller duration of the original audio, 8s and 12s in this paper, we assume it 8s if not differently specified.

Data segmentation is significant in vocal data augmentation for several reasons. First, it increases the amount of data available for model training by breaking down longer audio recordings into shorter segments. This helps prevent overfitting and improves the generalization ability of the model.

Second, segmentation allows for capturing different phonetic variations within the vocal data. By dividing the audio into smaller segments, the model can learn to recognize and generalize patterns at a more granular level.

Additionally, data segmentation helps to address computational limitations. Processing shorter audio segments requires less computational resources.

Overall, data segmentation as a data augmentation technique in vocal data processing plays a vital role in increasing data diversity, capturing variations, and improving model performance. It is a widely used technique to enhance the robustness and accuracy of vocal data models.

Chapter 4

CNN from scratch

4.1 Convolutional Neural Network

The proposed CNN architecture is designed to be applied to both Mel Spectrograms and MFCCs Figure 4.1, Figure 4.2. It consists of a 3-layer CNN with increasing feature maps, inspired by the 2D convolutional blocks of the LeNet architecture (Conv→Pool→Conv→Pool→FC); the idea of increasing the complexity of feature maps at each CNN layer is taken up by the AlexNet architecture.

The first layer consist in a 2D convolutional layer (Conv2D) with 32 filters, with 3x3 kernel size. It takes an input shape of (N, 20, 345, 1), when using MFCCs, representing the number of training data samples, the width and height of the image and the number of channels.

Since an MFCC diagram can be treated as a grayscale image, a single channel is used instead of the typical 3 channels for RGB images. The channel contains the intensities for the 20 MFCCs bands at 345 timesteps.

Alternatively, using the Mel spettrograms the input is in the shape (N, 128, 345, 1), where these value have an equivalent meaning.

The BatchNormalization layer is added to normalize the activations of the previous layer, which helps in stabilizing and speeding up the training process.

The normalization step helps in reducing the effect of small changes in the network's parameters on the subsequent layers. It ensures that the inputs to each layer have a similar distribution, which can accelerate training and improve model performance.

Additionally, batch normalization acts as a form of regularization by adding a small amount of noise to the layer inputs, which can help prevent overfitting.

A Rectified Linear Unit (ReLU) activation is applied using the Activation layer, which introduces non-linearity to the model and helps capture complex patterns.

A max-pooling layer with a pool size of 2x2 is added to downsample the feature maps, reducing the spatial dimensions and extracting the most important features.

To prevent overfitting, a dropout layer with a dropout rate of 0.3 is included. It randomly sets a fraction of the neurons to 0 during training, forcing the model to

learn more robust and generalized features.

This pattern of convolutional, normalization, activation, pooling, and dropout layers is repeated twice with increasing filter sizes (64 and 128) to capture more complex features.

After the final convolutional layer, a Flatten() layer is used to convert the 2D feature maps into a 1D vector.

These layers enable the model to extract hierarchical representations from the input MFCCs or mel spectrograms, capturing both local and global audio features.

Lastly, a fully connected layer (Dense) with a single unit and a sigmoid activation function is added for binary classification.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)            (None, 18, 343, 32)       320

 batch_normalization_3 (Batc  (None, 18, 343, 32)      128
 hNormalization)

 activation_3 (Activation)    (None, 18, 343, 32)       0

 max_pooling2d_2 (MaxPooling  (None, 9, 171, 32)       0
 2D)

 dropout_2 (Dropout)          (None, 9, 171, 32)        0

 conv2d_4 (Conv2D)            (None, 7, 169, 64)        18496

 batch_normalization_4 (Batc  (None, 7, 169, 64)       256
 hNormalization)

 activation_4 (Activation)    (None, 7, 169, 64)        0

 max_pooling2d_3 (MaxPooling  (None, 3, 84, 64)        0
 2D)

 dropout_3 (Dropout)          (None, 3, 84, 64)         0

 conv2d_5 (Conv2D)            (None, 1, 82, 128)        73856

 batch_normalization_5 (Batc  (None, 1, 82, 128)       512
 hNormalization)

 activation_5 (Activation)    (None, 1, 82, 128)        0

 flatten_1 (Flatten)          (None, 10496)             0

 dense_1 (Dense)              (None, 1)                 10497

=================================================================
Total params: 104,065
Trainable params: 103,617
Non-trainable params: 448
_________________________________________________________________
```

Figure 4.1: Summary of the CNN architecture

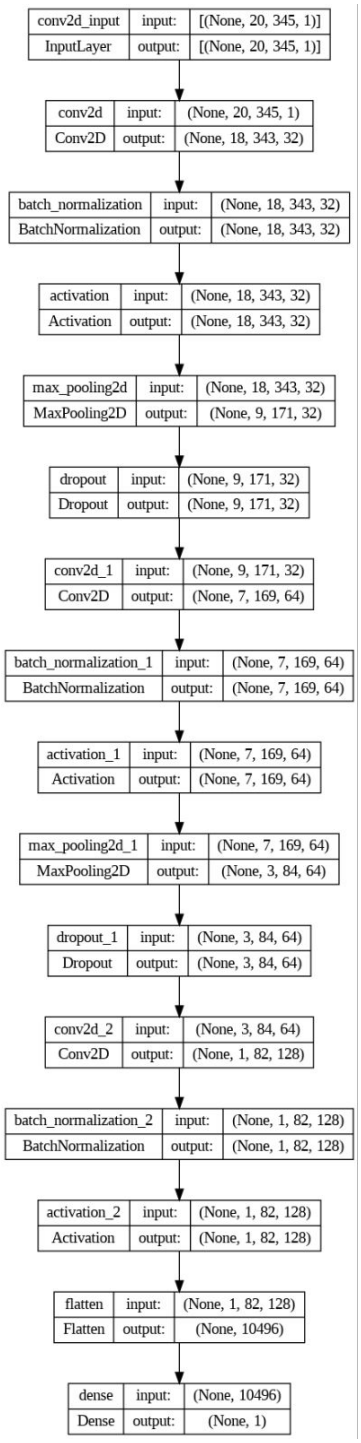

Figure 4.2: Schematic of the CNN architecture

The last step in network definition is the application of an optimizer. In this step, the optimizer 'adam' is specified to optimize the model's parameters during the training process.

It combines the concepts of both momentum and adaptive learning rates to efficiently update the model's parameters during training.

By using Adam as the optimizer in the model compilation step, the algorithm

efficiently updates the model's parameters, leading to improved convergence and better overall training performance.

The loss function chosen is a binary one, to measure the discrepancy between the predicted outputs and the true labels. Additionally, to evaluate the model's performance the metric accuracy is used during training and validation.

4.1.1 Experimental Results on Mel spettrograms

During the evaluations, the dataset was divided into two subsets: 80% for training, 20% for test. Furthermore, the training set has been divided in training 75% and validation set 15%. The optimal value of learning rate has been found at 2×10^{-5}.

During the training process, the model's performance is monitored, and the loss Figure 4.3 and accuracy Figure 4.4 values are recorded for each epoch. We observe that the model achieves promising results, with high accuracy and low loss values on both the training and validation sets. This indicates that the CNN model successfully learns discriminative features from the Mel spectrograms, enabling it to distinguish between healthy individuals and PD patients.

The model was set to run for 25 epochs, but it stopped early on the 21th epoch thanks to the early stopping parameter previously defined. The model presents a good accuracy with some overfitting, 6% more accurate on the training set. Still, the overall results on the validation set are satisfactory, reaching an accuracy of 93%.

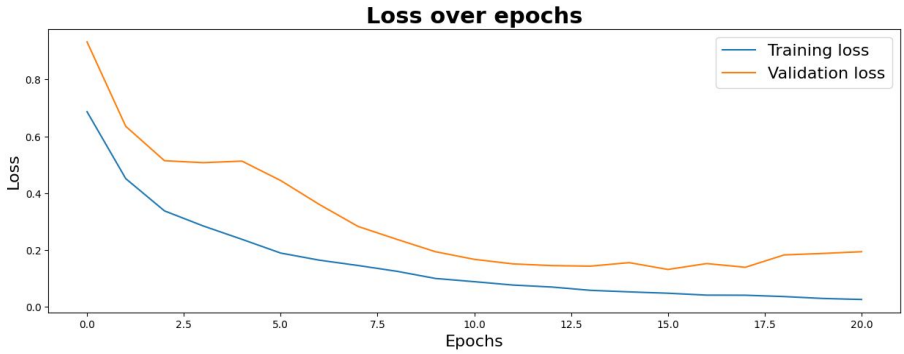

Figure 4.3: Graph of the loss function on the training and validation set

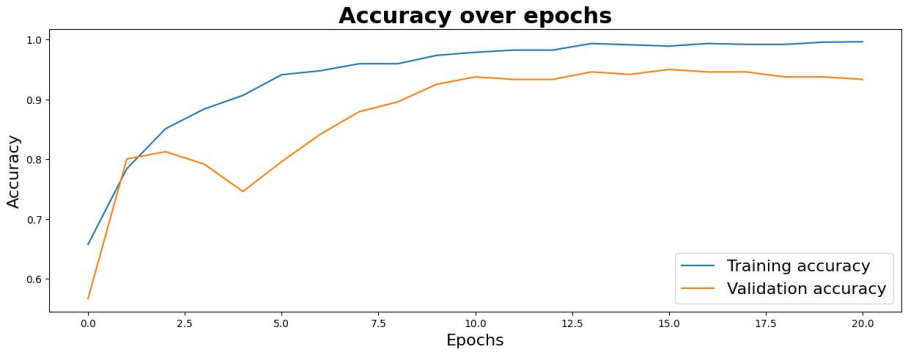

Figure 4.4: Graph of the accuracy on the training and validation set

The comprehensive evaluation using an independent testing set demonstrates the model's effectiveness and its potential for early-stage PD identification.

The results on the test set are the following with an accuracy of 94%:

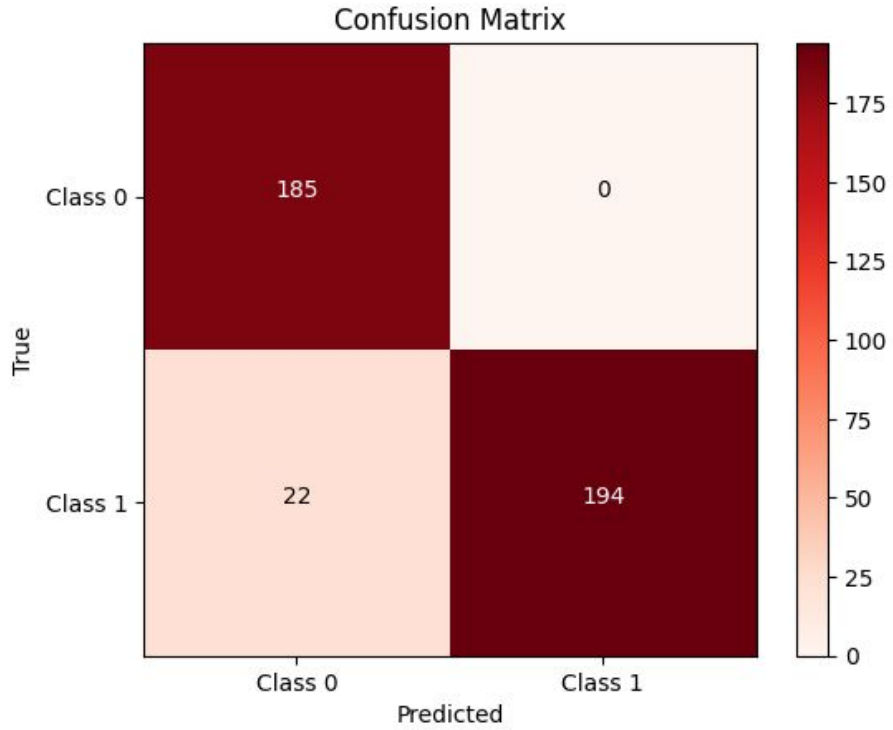

Figure 4.5: Confusion matrix of CNN on Mel

Class	Precision	Recall	F1-score	Support
HC	0.92	1.00	0.96	103
PD	1.00	0.94	0.97	144

Table 4.1: Classification Metrics on Mel spectrograms

4.1.2 Experimental Results on MFCCs

The MFCCs transformation performs slightly better than the Mel Spectrograms, without overfitting and with a better convergence. After 45 epochs the results achieved show 95% of accuracy on the train set and 94% on the validation set. For a bigger number of epochs the results were similar or worse on the validation set Figure 4.6, Figure 4.4.

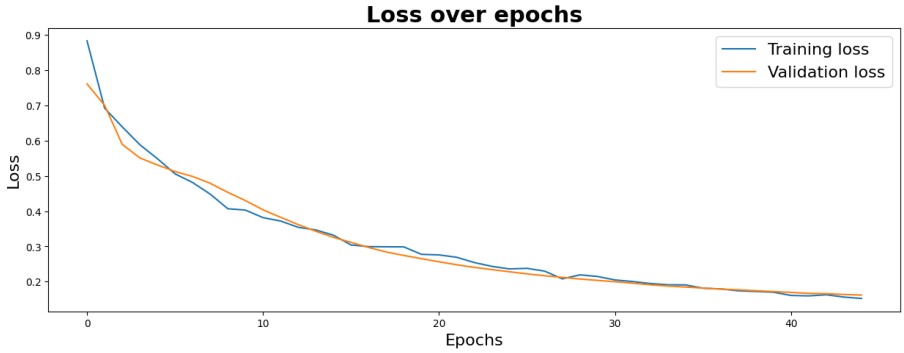

Figure 4.6: Graph of the loss function on the training and validation set

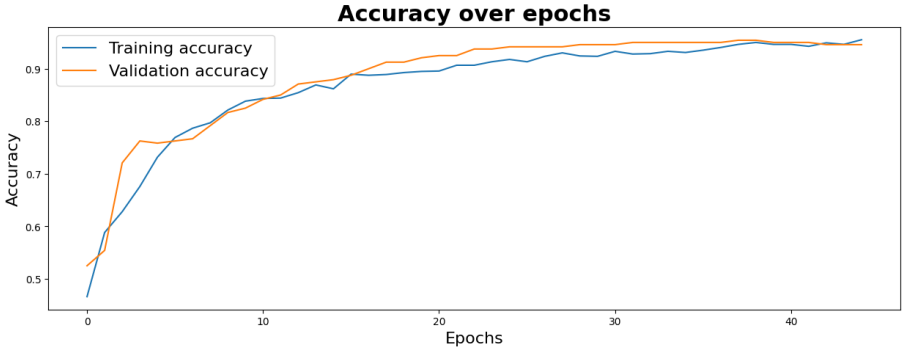

Figure 4.7: Graph of the accuracy on the training and validation set

The results on the test set are the following with an accuracy value of 95.51%, proving the good generalization power of the network, Figure 4.8.

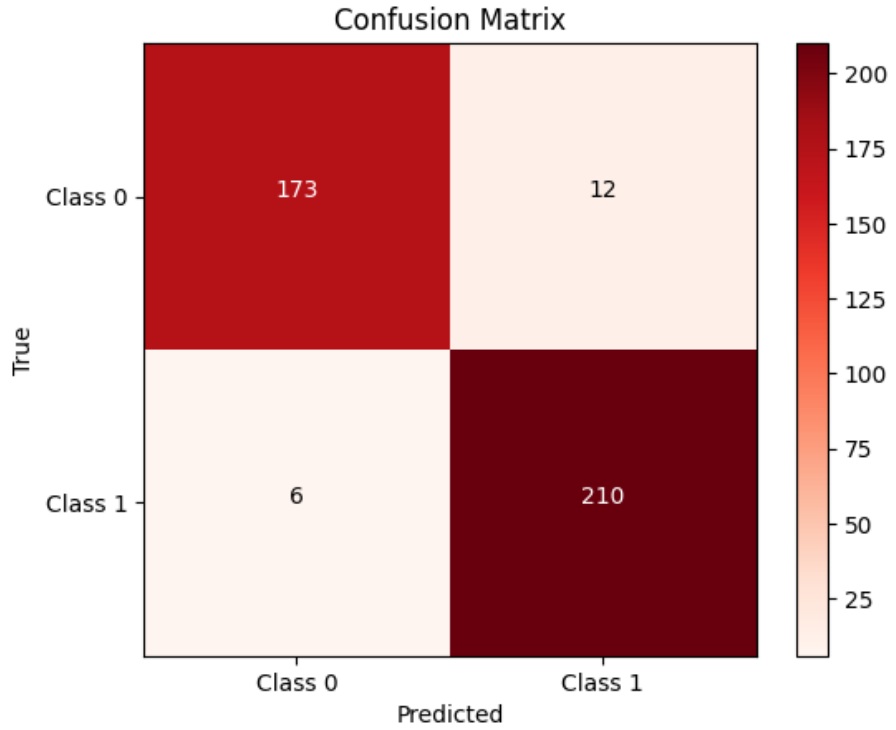

Figure 4.8: Confusion matrix of CNN on MFCCs

Class	Precision	Recall	F1-score	Support
HC	0.97	0.94	0.95	185
PD	0.95	0.97	0.96	216

Table 4.2: Classification Metrics om MFCCs

Chapter 5

Pre-trained network: VGG16

This chapter illustrates another network, implemented based on a pre-trained VGG16 convolutional neural network (CNN).

VGG16 is a convolutional neural network architecture developed by the Visual Geometry Group (VGG) at the University of Oxford. Its strength lies in its deep structure with 16 layers, which allows it to learn complex features and achieve high accuracy on image classification tasks. It application exploits the already trained convolutional layers of the network for the spatial feature extraction, and then build a binary classificator on top.

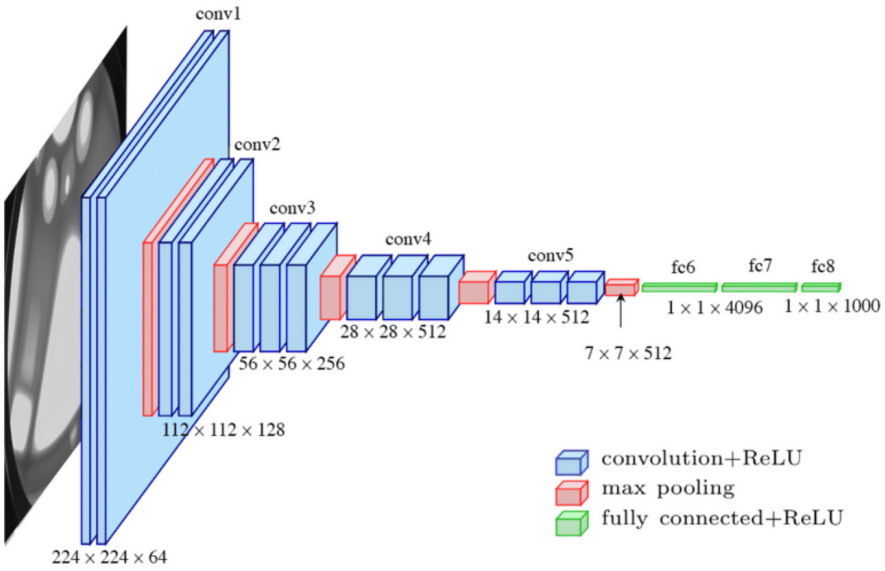

Figure 5.1: VGG16 architecture

It will be studied the Mel spectrogram trasformation but equal consideration are to be applied in the case of MFCCs.

5.1 Dataset creation

The VGG16 model is initialized with pre-trained weights from the ImageNet dataset, and the top layer is excluded to allow for further customization. The input shape of the model is set to (128, 345, 3), representing the dimensions of the input image. The pre-trained model is built to have as an input an RBG picture, so a necessary step is the transformation of a spectrogram into a RGB image representation. Initially, the spectrogram tensor is expanded by adding an extra dimension at the end. This is done to align with the expected input shape of the subsequent conversion operation.

This conversion process generates a corresponding RGB image tensor. The resulting tensor represents the same spectrogram information but in the format of a three-channel RGB image.

5.2 Feature extraction

First off, is necessary to freeze the weights of the pre-trained VGG16 model and prevent them from being updated during training. This ensures that only the newly added layers will be trained.

The model is then constructed by defining the input layer and feeding the pre-trained VGG16 model with the input data. The output of the VGG16 model is flattened to create a 1-dimensional feature vector. A dense layer with 128 units is added to capture high-level representations in the extracted features. A dropout layer with a dropout rate of 0.5 is included to prevent overfitting. The final output layer consists of a single neuron with a sigmoid activation function, suitable for binary classification tasks.

The model is compiled using the binary cross-entropy loss function and the Adam optimizer with a learning rate of 1×10^{-5}. Additionally, the accuracy metric is specified for evaluation during training.

To prevent overfitting and monitor the validation loss, early stopping is implemented with a patience of 5. This means that training will stop if the validation loss does not improve after 5 consecutive epochs. In figure Figure 5.2 we have a summary of the network.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 32, 345, 3)]      0

 vgg16 (Functional)          (None, 1, 10, 512)        14714688

 flatten (Flatten)           (None, 5120)              0

 dense (Dense)               (None, 128)               655488

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 15,370,305
Trainable params: 655,617
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 5.2: Summary of the network

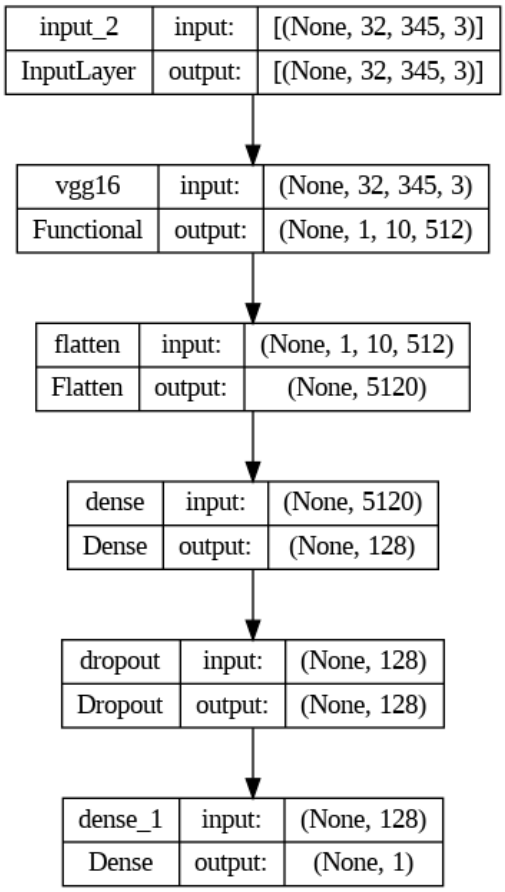

Figure 5.3: Block diagram of the network

5.2.1 Experimental results on Mel spectrograms

The model shows a excellent classification ability, achieving a training accuracy of 99% and a validation one of 97%, as shown in Figure 5.5.

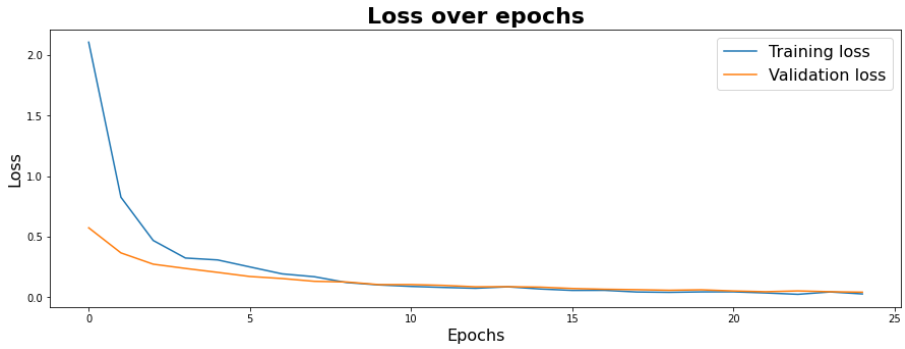

Figure 5.4: Graph of the loss on the training and validation set

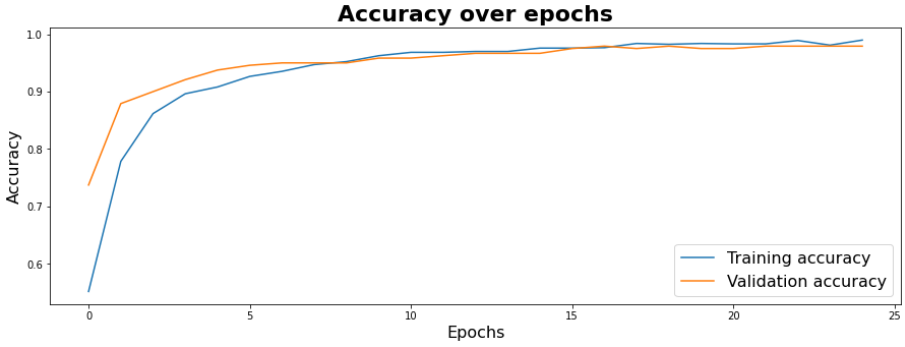

Figure 5.5: Graph of the accuracy on the training and validation set

Test results on Mel spectrogram

The results on the test set are the following, with an optimal accuracy of 98,7%:

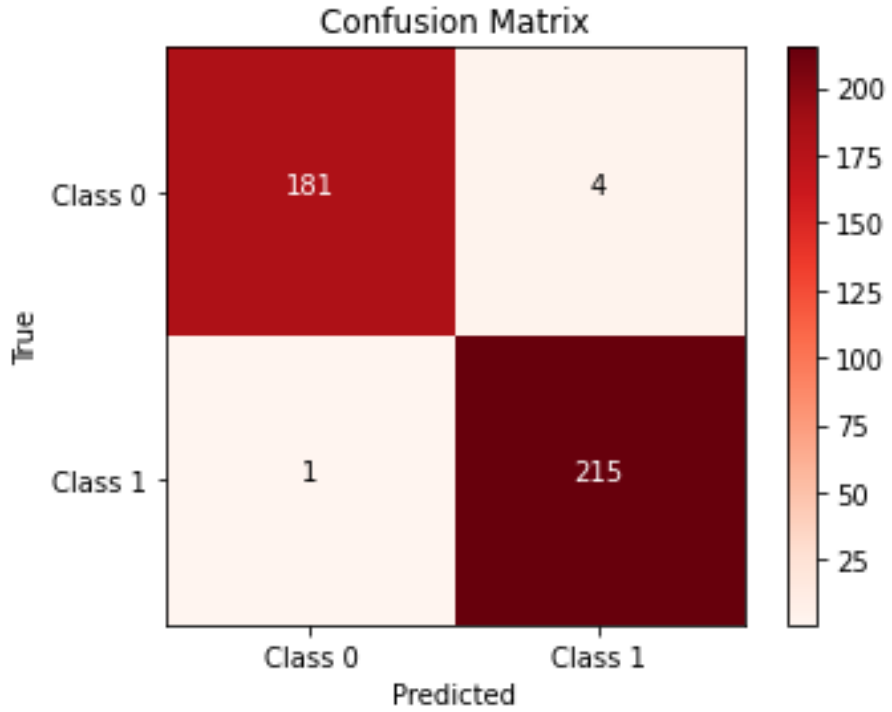

Figure 5.6: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.99	0.98	0.99	185
PD	0.98	1.00	0.99	216

Table 5.1: Classification Metrics on Mel spectrograms

5.2.2 Experimental results on MFCCs spectrograms

The results achieved using the MFCC transformation are similar to the ones on the Mel spectrogram, with an accuracy of 95% on both training and validation set, as shown in Figure 5.8.

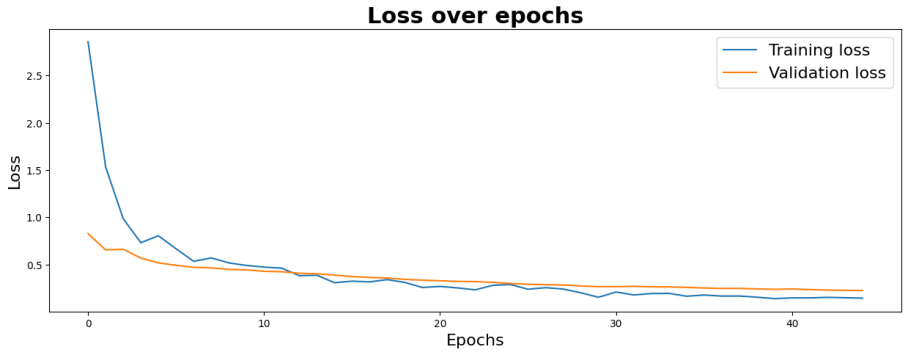

Figure 5.7: Graph of the loss on the training and validation set

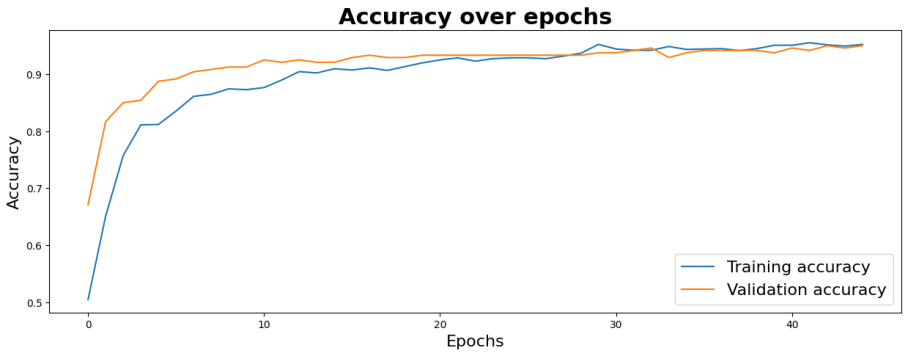

Figure 5.8: Graph of the accuracy on the training and validation set

Test results on MFCCs spectrogram

The results on the test set are the following with an optimal accuracy of 95,7%:

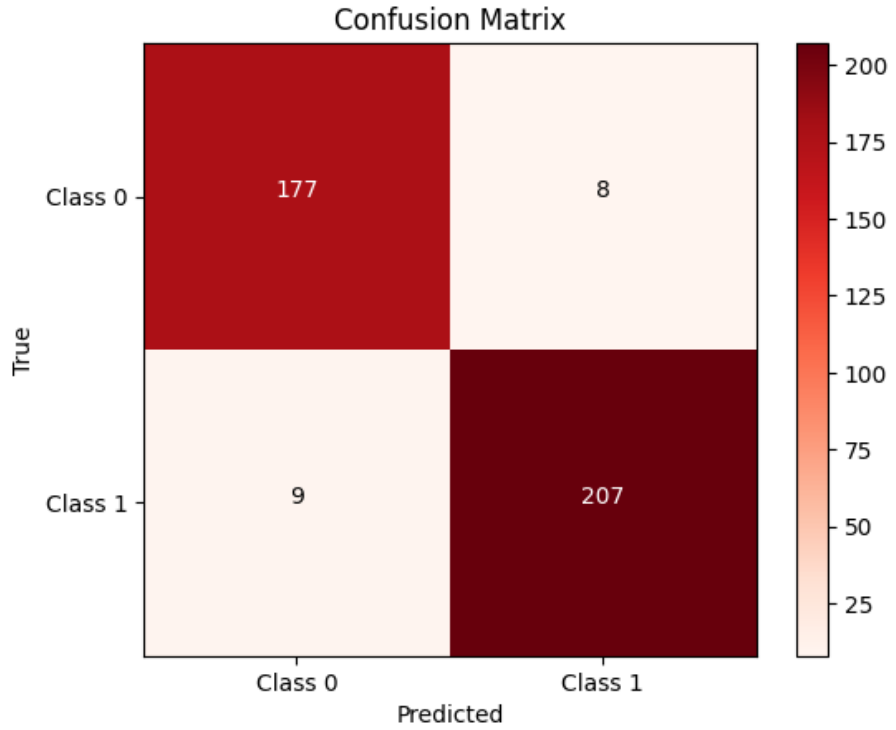

Figure 5.9: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.95	0.96	0.95	103
PD	0.96	0.96	0.96	144

Table 5.2: Classification Metrics on Mel spectrograms

5.2.3 Fine tuning

Fine-tuning refers to the process of selectively unfreezing and jointly training the top layers of a pre-trained model alongside the newly added classifier. By adjusting the more abstract representations of the reused model, fine-tuning aims to enhance their relevance to the specific problem at hand. This approach is particularly advantageous when dealing with large datasets that significantly differ from the original dataset. In such cases, fine-tuning a larger number of layers can be effective.

The network gives extremely satisfactory results so there is not need for any further fine tuning step.

Chapter 6

1-D CNN

The 1-D CNN are a particular type of convolutional neural network, that shares with the more common 2-D CNN the architecture structure, alternating convolutional layers and pooling layer, the parameter sharing and the hierarchical feature, extracting low-level features in the initial layers and gradually learn higher-level and more abstract features in deeper layers. Differently from 2-D networks, 1-D CNN processes sequential or time series data, where the input is represented as a 1D array or a sequence of vectors, it has a one-dimension sliding filter across the input data in one dimension (temporal dimension). It performs element-wise multiplications and summations, producing feature maps that capture local patterns or features along the sequence. This make them particularly good for audio analysis, reducing the number of trainable parameter maintaining a good accuracy.

6.1 Feature extraction

In this study, it was designed and implemented a 1-D convolutional neural network (1-D CNN) model for the classification task. The model architecture consisted of two convolutional layers, with 64 filters each. The input shape of the model was specified as (1034, 20) when using MFCCs and (1034,128) when using Mel spectrogram transformation. For this network we tried three different size of the kernel: 3, 5 and 7, comparing the results obtained.

To prevent overfitting, a single layer of drop-out is inserted with a value of 0.5.

The output is then feeded to a max pooling with a pool size of 2 to divide in half the size of the matrix.

A fully connected layer with 100 units and a rectified linear activation function (ReLU) was added to capture higher-level representations in the data. Finally, a dense layer with a sigmoid activation function was included as the output layer for binary classification.

To optimize the model, we employed the Adam optimizer with a learning rate of 1×10^{-5}. The binary cross-entropy loss function was used as the objective to guide the model training, and the accuracy metric was employed to evaluate the model's

performance.

During the training process, a batch size of 80 was utilized, and the model was trained for a maximum of 100 epochs. To prevent overfitting the early stopping callback was utilized with a monitoring criterion of validation loss and a patience of 8 epochs.

The training and validation data set have the characteristics described before other than the fact that the data segmentation occur on samples of 12 second instead of 8s. Longer audio data allow us to maintain a stronger temporal correlation which is a major feature on which 1-D CNN base their classification technique. In figure Figure 6.1 we have a summary of the network with kernel 3.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv1d (Conv1D)              (None, 1032, 64)          3904

 conv1d_1 (Conv1D)            (None, 1030, 64)          12352

 dropout (Dropout)            (None, 1030, 64)          0

 max_pooling1d (MaxPooling1D  (None, 515, 64)          0
 )

 flatten (Flatten)            (None, 32960)             0

 dense (Dense)                (None, 100)               3296100

 dense_1 (Dense)              (None, 1)                 101

=================================================================
Total params: 3,312,457
Trainable params: 3,312,457
Non-trainable params: 0
_________________________________________________________________
```

Figure 6.1: Summary of the network

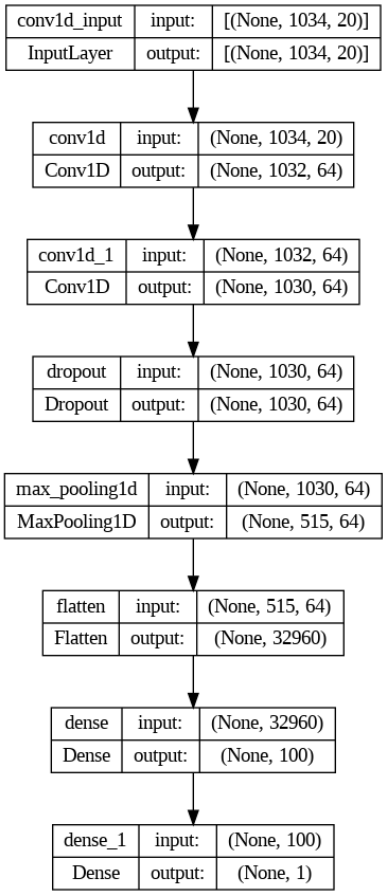

Figure 6.2: Block diagram of the network

6.2 Experimental Results on Mel spectrograms

The results achieved with Mel spectrogram are good for all three values of the kernel parameter, with 93% of validation accuracy with kernel 3 Figure 6.4, 95% with kernel 5 Figure 6.6 and 98% with kernel 7 Figure 6.8. The kernel 7 model is chosen to be evaluated on the test set.

6.2.1 Kernel 3

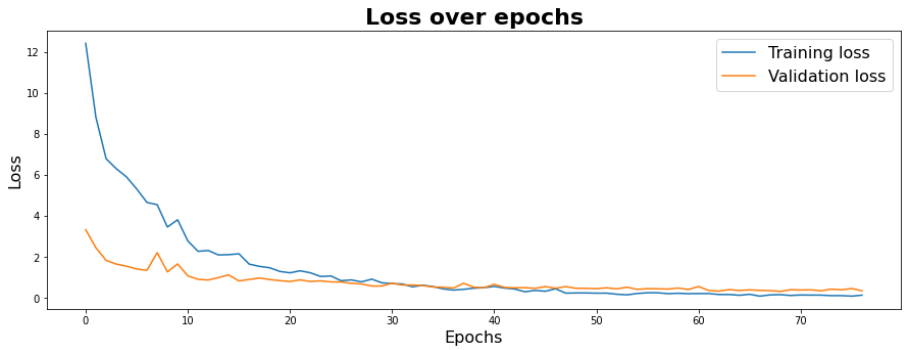

Figure 6.3: Graph of the loss on the training and validation set

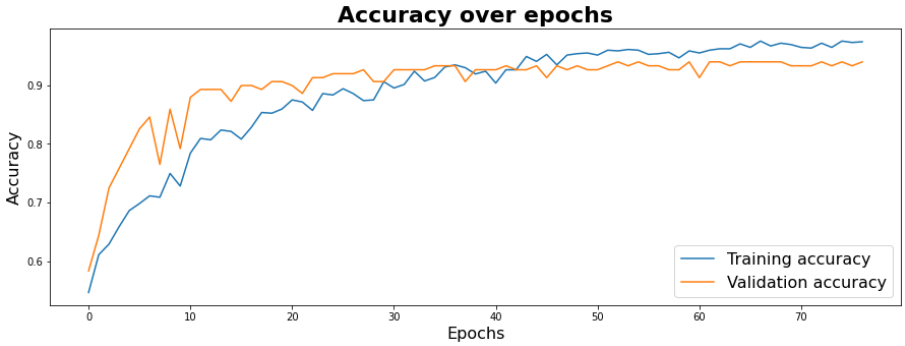

Figure 6.4: Graph of the accuracy on the training and validation set

6.2.2 Kernel 5

Here the early stopping event is particularly evident.

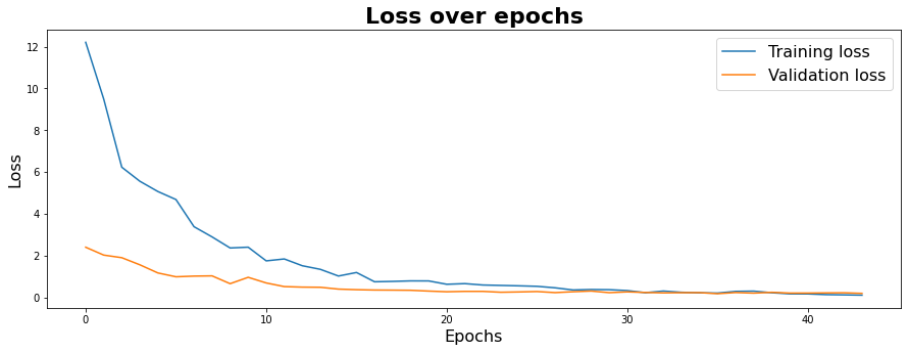

Figure 6.5: Graph of the loss on the training and validation set

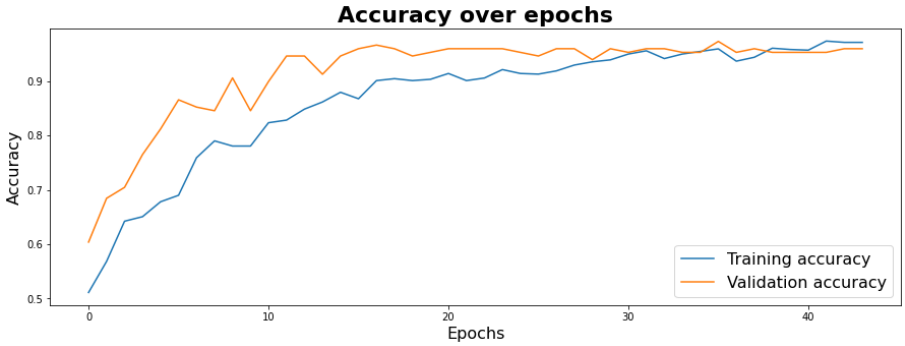

Figure 6.6: Graph of the accuracy on the training and validation set

6.2.3 Kernel 7

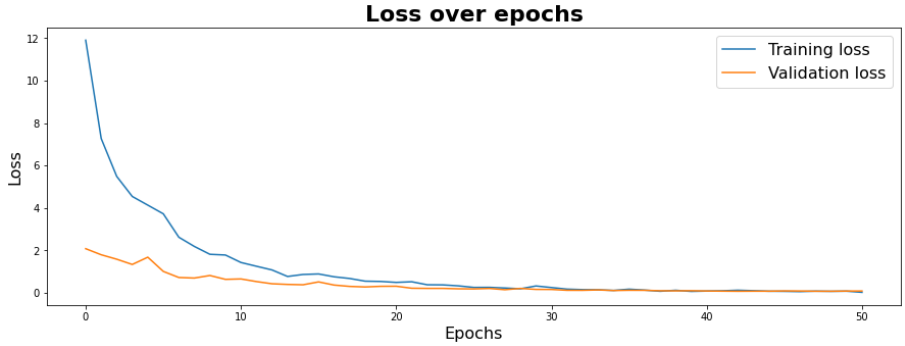

Figure 6.7: Graph of the loss on the training and validation set

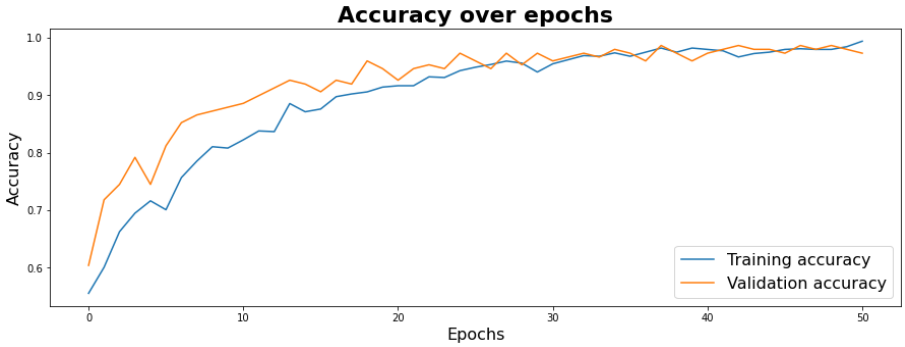

Figure 6.8: Graph of the accuracy on the training and validation set

6.2.4 Test evaluation

The test evaluation was conducted only on the model that performed the best in this case the one with kernel 7. The accuracy achieved in the test set is of 96%, proving the optimal classification power of the network.

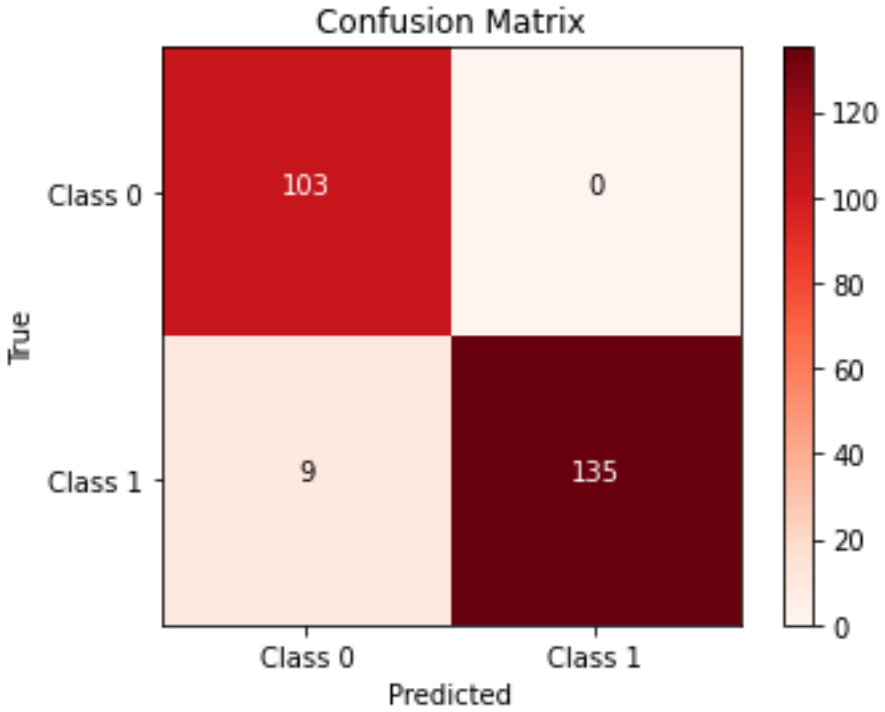

Figure 6.9: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.89	1.00	0.94	185
PD	1.00	0.90	0.95	216

Table 6.1: Classification Metrics on Mel spectrograms kernel 7

6.3 Experimental Results on MFCCs spectrograms

The results achieved with this transformation are good for all three values of the kernel parameter, with 92% of validation accuracy with kernel 3 Figure 6.11, 94% with kernel 5 Figure 6.13 and 95% with kernel 7 FFigure 6.15. The kernel 7 model is chosen to be evaluated on the test set.

6.3.1 Kernel 3

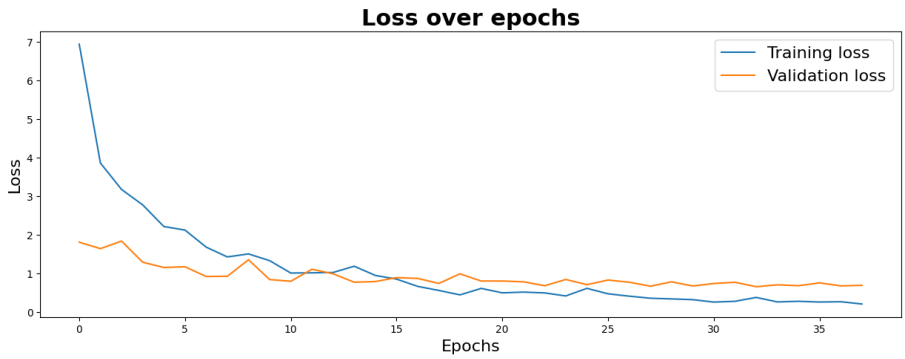

Figure 6.10: Graph of the loss function on the training and validation set

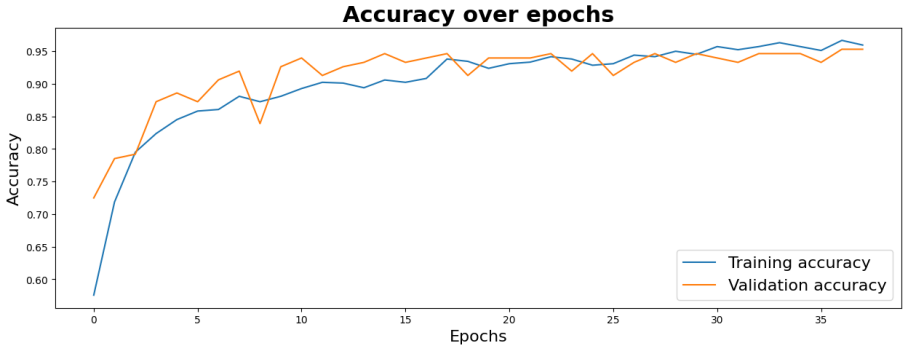

Figure 6.11: Graph of the accuracy on the training and validation set

6.3.2 Kernel 5

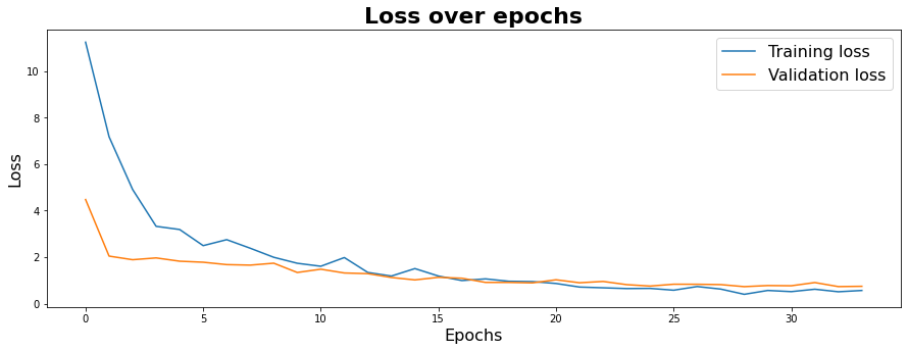

Figure 6.12: Graph of the loss function on the training and validation set

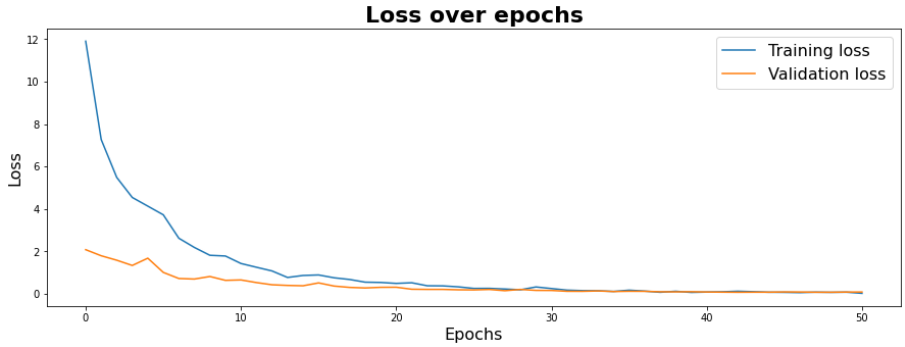

Figure 6.13: Graph of the accuracy on the training and validation set

6.3.3 Kernel 7

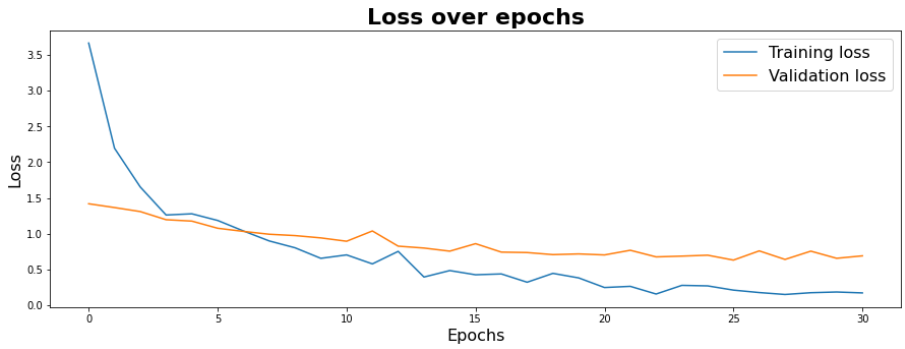

Figure 6.14: Graph of the loss function on the training and validation set

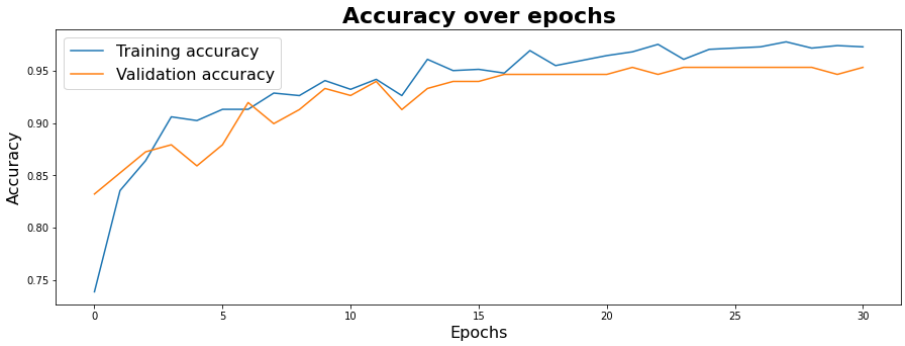

Figure 6.15: Graph of the accuracy on the training and validation set

6.3.4 Test evaluation

The test evaluation was conducted only on the model that performed the best in this case the one with kernel 7. The accuracy achieved in the test set is of 94%, proving good classification power of the network.

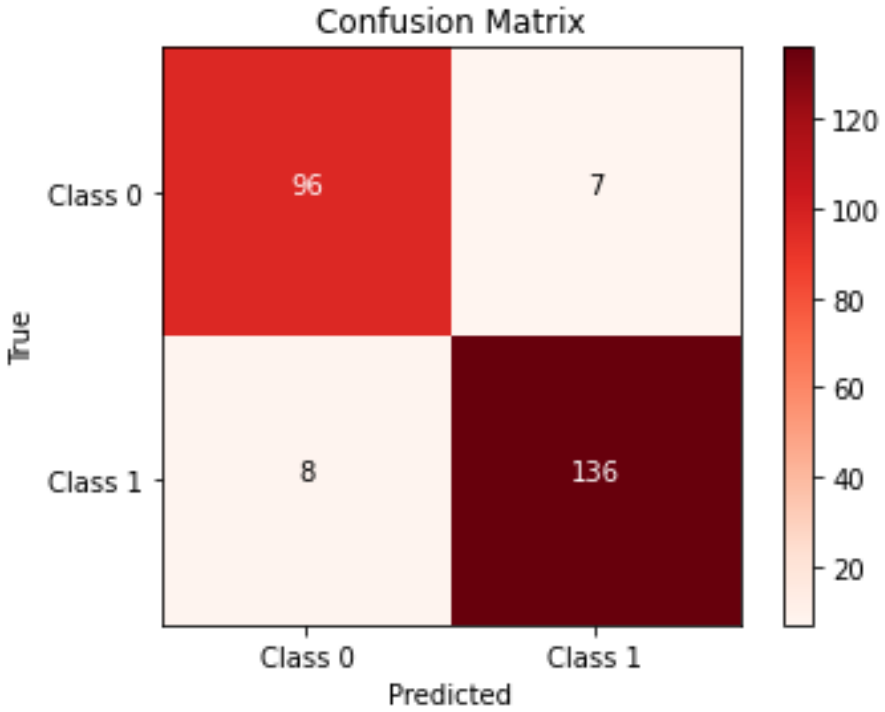

Figure 6.16: Confusion metrics

Class	Precision	Recall	F1-score	Support
HC	0.89	1.00	0.94	185
PD	1.00	0.90	0.95	216

Table 6.2: Classification Metrics on Mel spectrograms kernel 7

Chapter 7

RNN from scratch

Recurrent Neural Networks (RNNs) offer valuable capabilities in the classification of vocal audio. RNNs are well-suited for capturing temporal dependencies and patterns in sequential data, making them ideal for analyzing speech signals that exhibit dynamic variations over time. By leveraging the recurrent nature of RNNs, these models can capture the temporal dynamics and subtle nuances present in vocal audio, enabling classification and identification of specific speech characteristics which can be useful also with Parkinson's disease. The strength of RNNs lies in their ability to adapt to different speaking rates, and incorporate contextual information from previous time steps. Furthermore, RNNs can learn hierarchical representations, allowing them to capture both local and global patterns in vocal audio, leading to robust and interpretable classification results.

7.1 Simple RNN

7.1.1 Feature Extraction

In this study, it was designed and implemented a recurrent neural network (RNN) model for the classification task. The model architecture consisted of a single layer of SimpleRNN units, with 256 units in total. The input shape of the model was specified as (1034, 20) when using MFCCs and (1034, 128) when using Mel spectrogram transformation.

A fully connected layer with 48 units and a rectified linear activation function (ReLU) was added to capture higher-level representations in the data. Finally, a dense layer with a sigmoid activation function was included as the output layer for binary classification.

To optimize the model, we employed the Adam optimizer with a learning rate of 1×10^{-4}. The binary cross-entropy loss function was used as the objective to guide the model training, and the accuracy metric was employed to evaluate the model's performance.

During the training process, a batch size of 80 was utilized, and the model was

trained for a maximum of 240 epochs. To prevent overfitting the early stopping callback was utilized with a monitoring criterion of validation loss and a patience of 5 epochs.

In figure Figure 7.1 we have a summary of the network.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 256)               70912

 dense (Dense)               (None, 48)                12336

 dense_1 (Dense)             (None, 1)                 49

=================================================================
Total params: 83,297
Trainable params: 83,297
Non-trainable params: 0
_________________________________________________________________
```

Figure 7.1: Summary of the network with out drop out

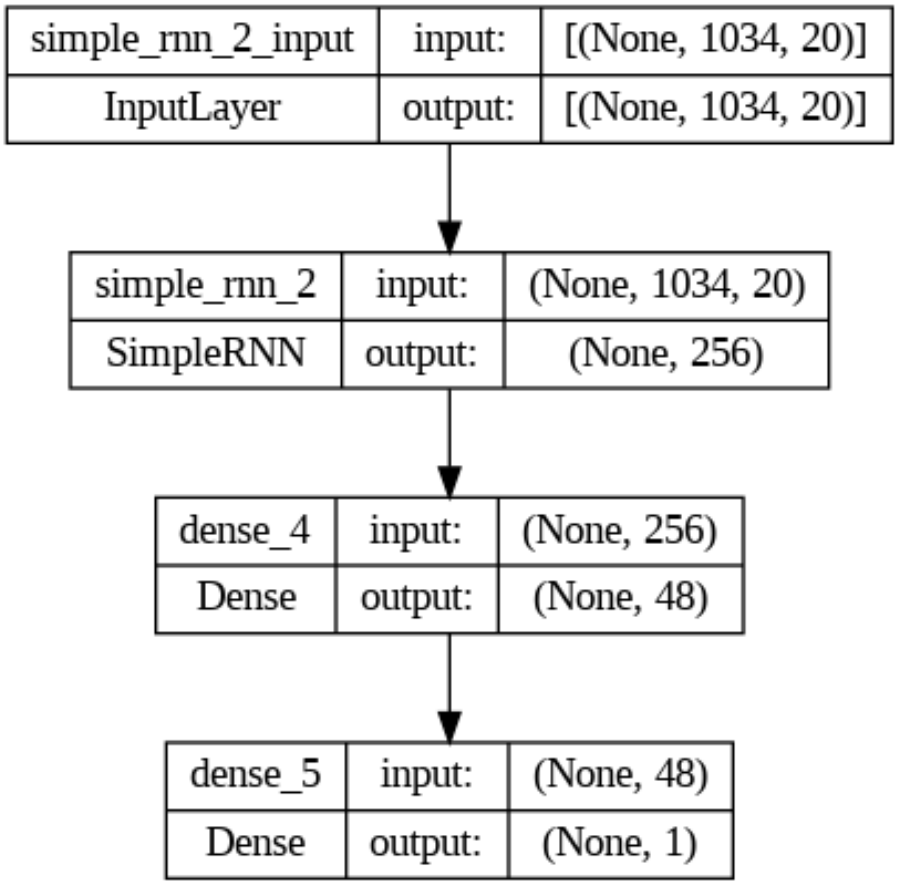

Figure 7.2: Block diagram of the network with out drop out

The training and validation data set have the characteristics described before other than the fact that the data segmentation occur on samples of 12 second instead of 8s. Longer audio data allow us to maintain a stronger temporal correlation which is a major feature on which RNNs base their classification technique.

7.1.2 Experimental results on Mel spectrograms

The following results do not matched the hoped behaviour as both the training and the the validation accuracy stays relatively low compared to the other proposed networks, respectively 90% and 77%. Furthermore, the validation accuracy reaches very low values compared to the training one as we can see from Figure 7.4. This can be liked to a problem of overfitting.

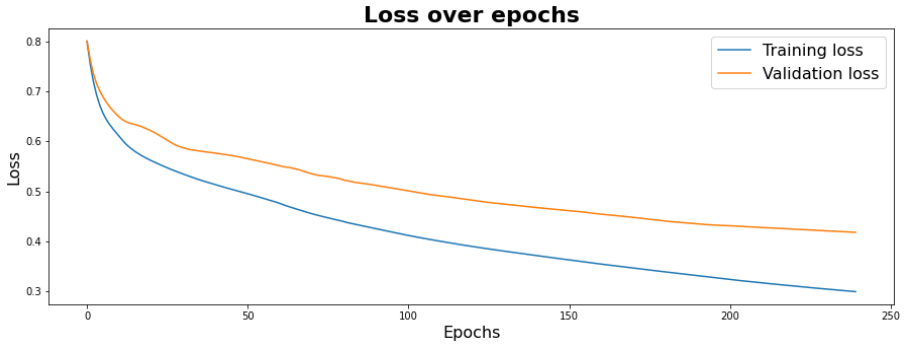

Figure 7.3: Graph of the loss function on the training and validation set

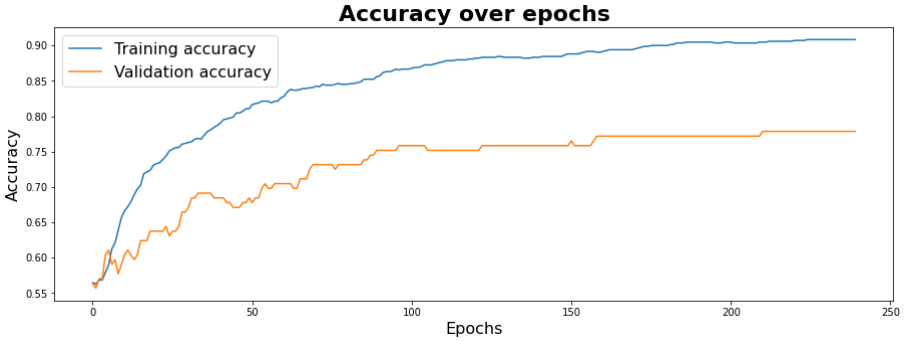

Figure 7.4: Graph of the accuracy on the training and validation set

7.1.3 Experimental results on MFCCs spectrograms

The same conclusion can be made for the MFCCs transformation: the accuracy reached is not optimal, with values lower than the Mel spectrogram ones, 83% on the training set and 72% on the validation set, as shown in Figure 7.6. The problem can be due to overfitting.

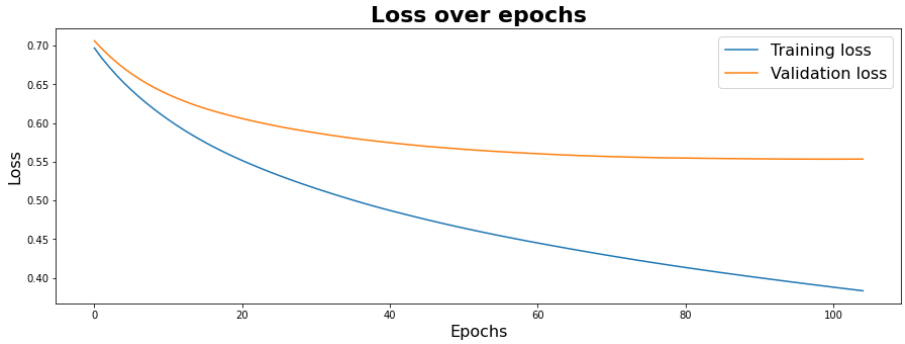

Figure 7.5: Graph of the loss function on the training and validation set

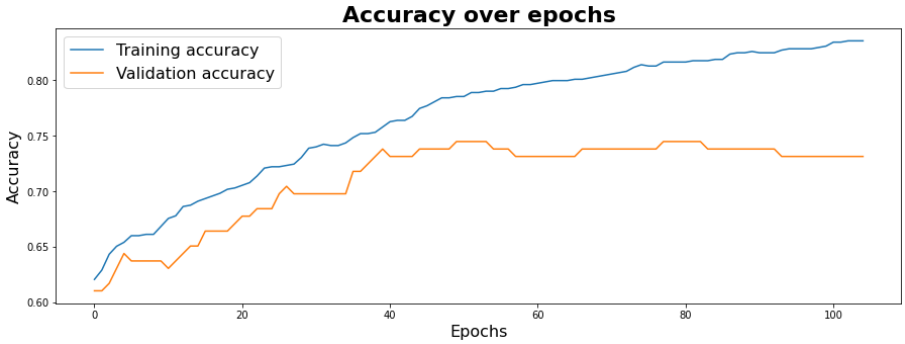

Figure 7.6: Graph of the accuracy on the training and validation set

7.1.4 Drop out

To mitigate overfitting, it was incorporated a dropout layer with a rate of 0.2 after the RNN layer. This helped regularize the model and prevent excessive reliance on specific input features. The results after this step are much more satisfactory. In figure Figure 7.7 we have a summary of the network.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 simple_rnn_1 (SimpleRNN)     (None, 256)               70912

 dropout_1 (Dropout)          (None, 256)               0

 dense_2 (Dense)              (None, 48)                12336

 dense_3 (Dense)              (None, 1)                 49

=================================================================
Total params: 83,297
Trainable params: 83,297
Non-trainable params: 0
_________________________________________________________________
```

Figure 7.7: Summary of the network with drop out

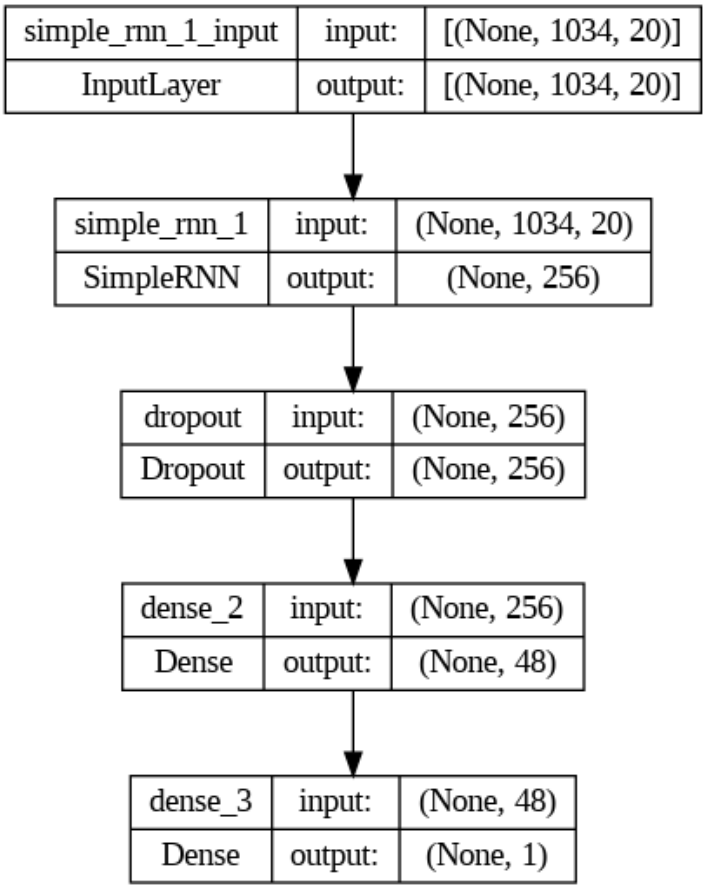

Figure 7.8: Block diagram of the network with drop out

Experimental results on Mel spectrograms

The results achieved with the dropout layer are still not good, with accuracy values of 78% on the training set and 77% on the validation set as shown in Figure 7.10.

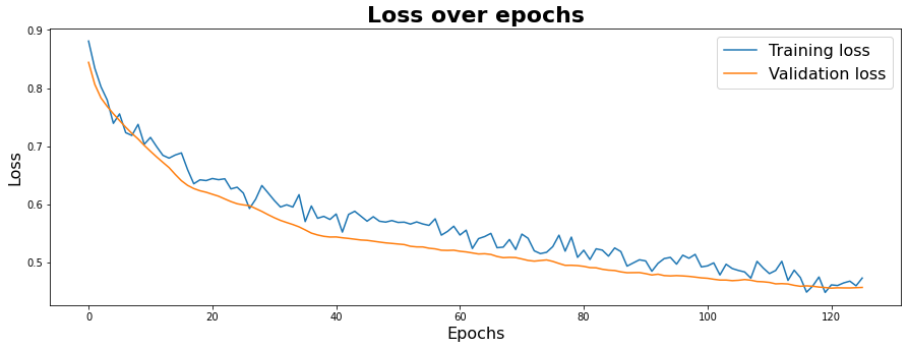

Figure 7.9: Graph of the loss function on the training and validation set

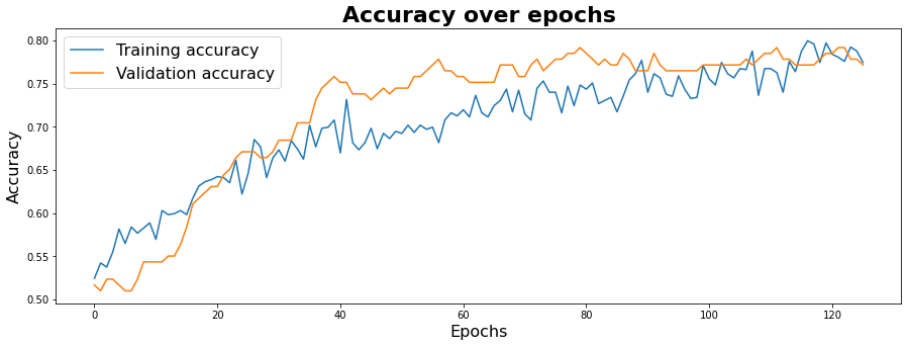

Figure 7.10: Graph of the accuracy on the training and validation set

Test results on Mel spectrogram

The results on the test set reflects the mediocre ones we have in the validation set with accuracy on the test set of 78.54%.

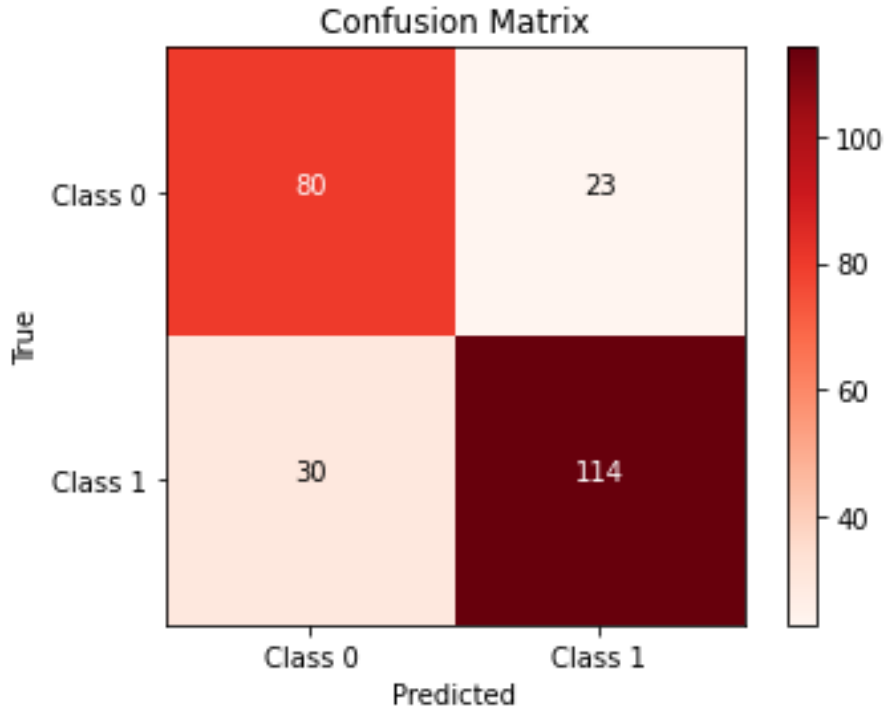

Figure 7.11: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.73	0.78	0.75	103
PD	0.83	0.79	0.81	144

Table 7.1: Classification Metrics on Mel spectrograms

Experimental results on MFCCs spectrograms

Like in the Mel spectrograms case, the added dropout layer helps the network generalize better, but still the results are not satisfactory. The accuracy level reached are shown in Figure 7.13

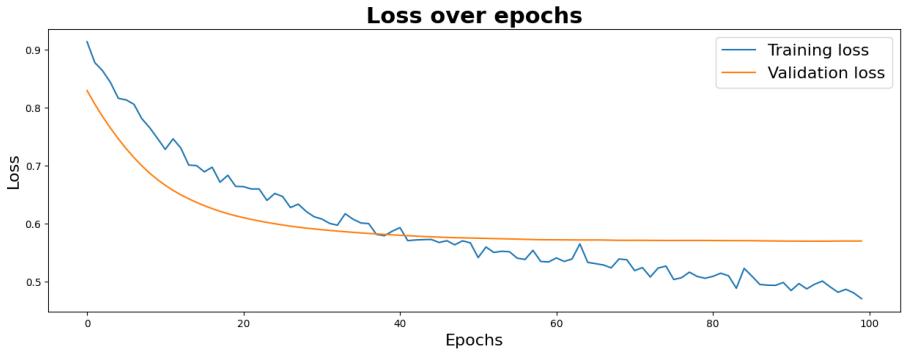

Figure 7.12: Graph of the loss function on the training and validation set

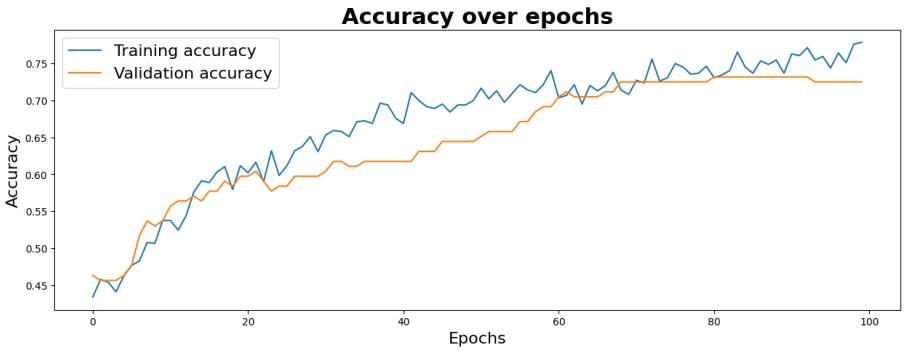

Figure 7.13: Graph of the accuracy on the training and validation set

Test results on MFCCs

The results on the test set are the following with an accuracy of 70.8%:

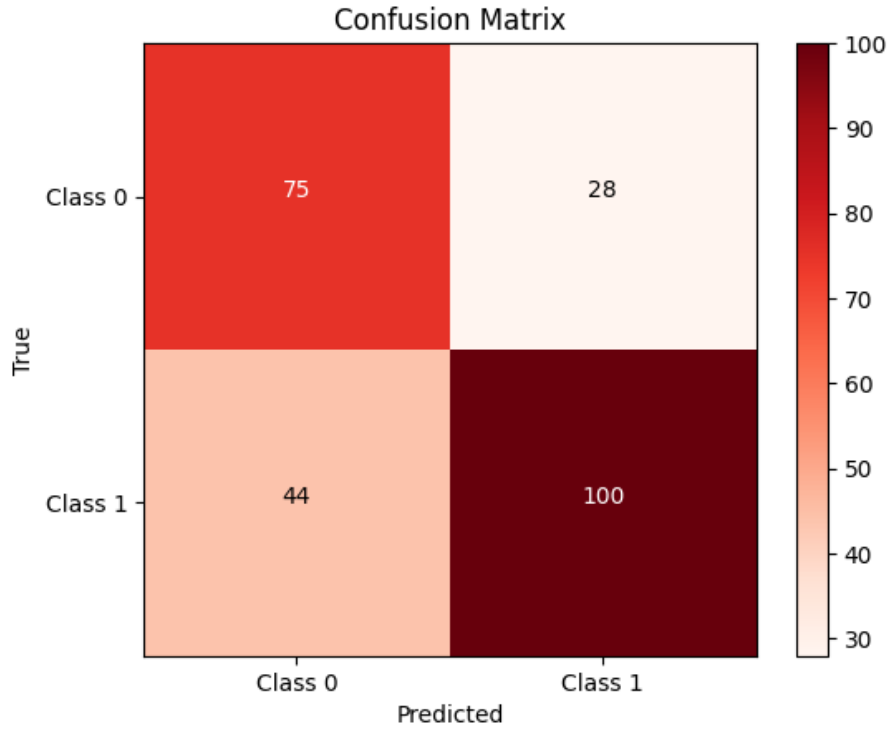

Figure 7.14: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.63	0.73	0.68	103
PD	0.78	0.69	0.74	144

Table 7.2: Classification Metrics on Mel spectrograms

7.1.5 RNN with Audio Samples

Since RNN takes as an input data that evolve in time and that have a time correlation between samples, it is possible to give to the network directly the file audio. In this section several function were implemented to generate overlapping windows of the signal which will be fed into the network. Due to some hardware limitations that did not allow the windows to have the desired characteristics, so some modifications have been necessary. First of all the sampling rate could only reach 16000 sample per second instead of the desired 44100 sps. Moreover, the window size could not be settled at 512 as desired but had to be put at 2048 due to limited RAM.

All these limitation made the data set unfit to train the RNN and the results are not satisfactory, reaching 69% of accuracy on the train set and 63% on the validation set. This is mainly due to the fact that the temporal correlation between samples is largely lost due to a vary sparse sampling.

7.2 LSTM

In view of the poor results given by the simple RNN network, a more complex network was implemented, an LSTM, in order to improve the outputs.

While RNNs are known for their ability to model sequential dependencies, LSTMs are a specialized type of RNN that address the limitation of traditional RNNs in capturing long-term dependencies.

RNNs are based on a simple recurrent unit that processes inputs and recurrent connections to pass information from one time step to the next. However, RNNs suffer from the vanishing gradient problem, which hampers their ability to retain and propagate information over long sequences.

LSTMs, on the other hand, are a more sophisticated architecture that includes memory cells, input gates, forget gates, and output gates. These gates enable LSTMs to selectively retain and forget information at each time step, thus allowing them to capture and remember long-term dependencies more effectively. The memory cells act as storage units that can preserve information over multiple time steps, enhancing the network's ability to learn and model complex temporal patterns.

The strengths of LSTMs in vocal audio processing and other sequential tasks are notable.

The network topology is completely equivalent to the one proposed for the RNN replacing the simple RNN layer with a LSTM of 156 neurons. In figure Figure 7.15 we have a summary of the network.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 156)               177840

 dropout (Dropout)           (None, 156)               0

 dense (Dense)               (None, 48)                7536

 dense_1 (Dense)             (None, 1)                 49

=================================================================
Total params: 185,425
Trainable params: 185,425
Non-trainable params: 0
_________________________________________________________________
```

Figure 7.15: Summary of the network

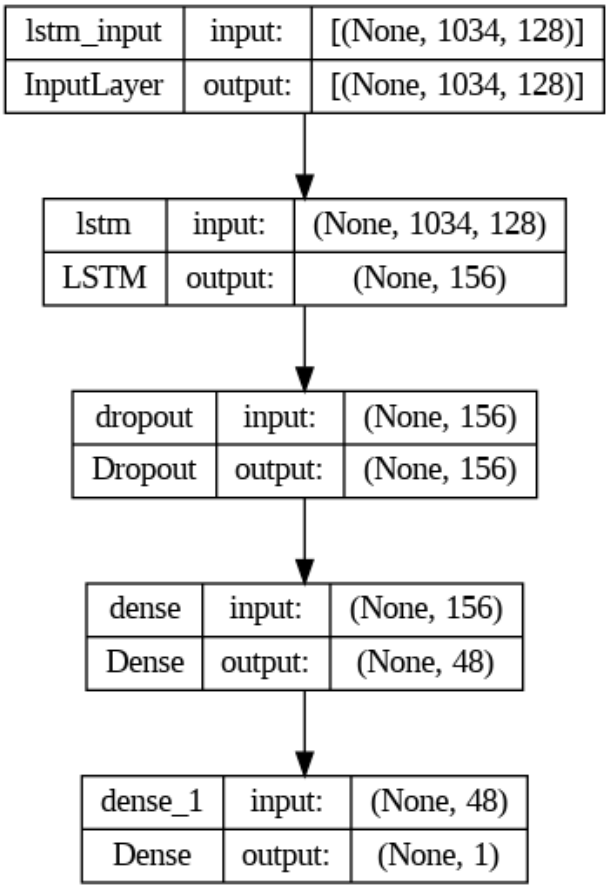

Figure 7.16: Block diagram of the network

7.2.1 Experimental results on Mel spectrograms

The model shows a very good classification ability, achieving a training accuracy of 96% and a validation one of 95%, as shown in Figure 7.18.

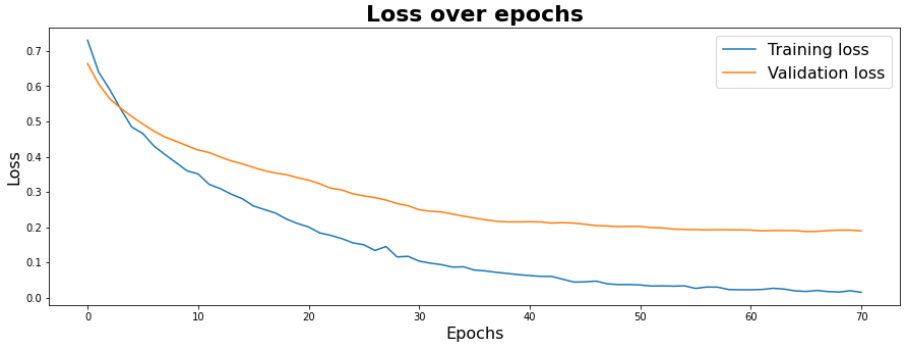

Figure 7.17: Graph of the loss on the training and validation set

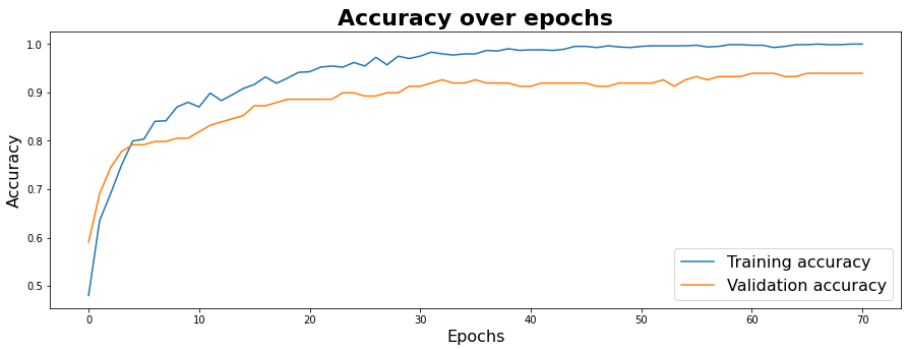

Figure 7.18: Graph of the accuracy on the training and validation set

Test results on Mel spectrogram

The results on the test set are the following with an optimal accuracy of 96.36%:

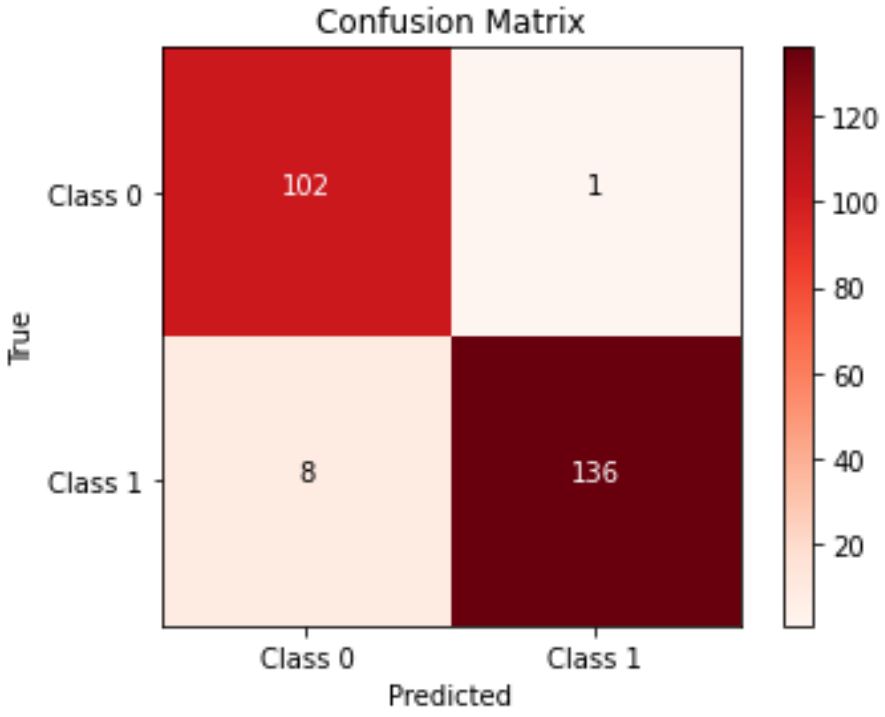

Figure 7.19: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.93	0.99	0.96	103
PD	0.99	0.94	0.97	144

Table 7.3: Classification Metrics on Mel spectrograms

7.2.2 Experimental results on MFCCs spectrograms

The results achieved using the MFCC transformation are similar to the ones achieved with the Mel spectrogram, with an accuracy of 99% on the training set and 95%, as shown in Figure 7.21.

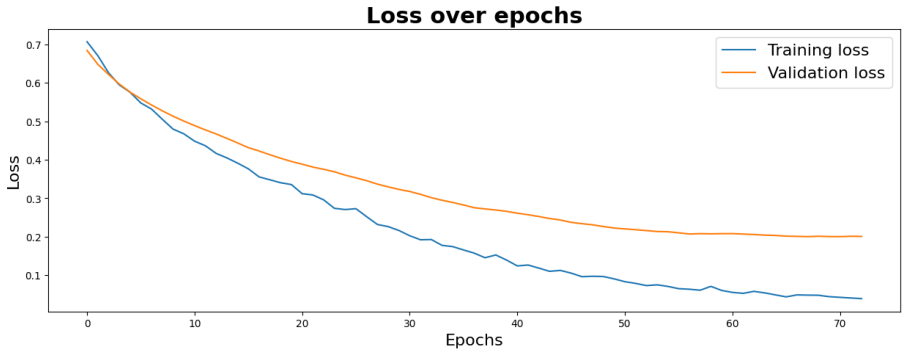

Figure 7.20: Graph of the loss on the training and validation set

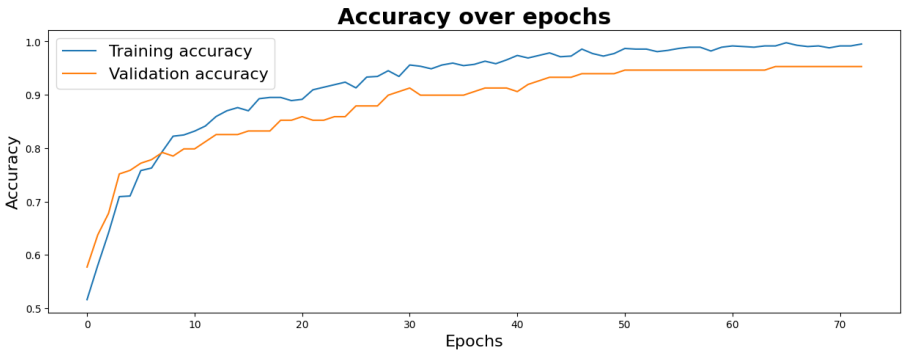

Figure 7.21: Graph of the accuracy on the training and validation set

Test results on MFCCs spectrogram

The results on the test set are the following with an optimal accuracy of 94%:

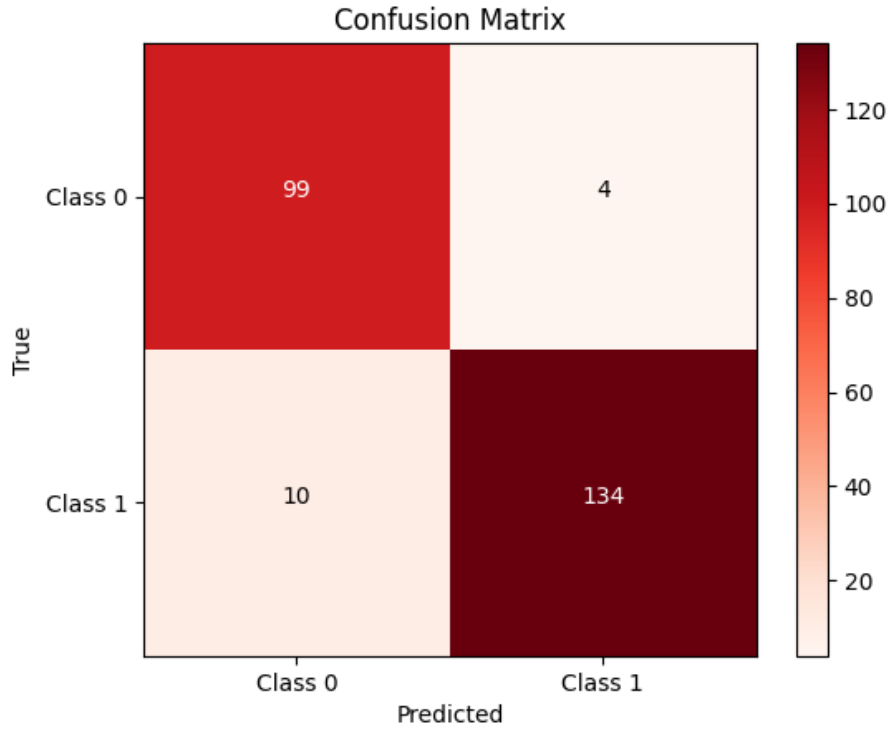

Figure 7.22: Confusion matrix

Class	Precision	Recall	F1-score	Support
HC	0.91	0.96	0.93	103
PD	0.97	0.93	0.95	144

Table 7.4: Classification Metrics on Mel spectrograms

7.2.3 LSTM with audio sample

For completeness, we have also run the LSTM directly applied to the file audio. The same limitations discussed for the RNN impacted the goodness of the results. The results given are not far from the random guess so the graph will not be included.

Chapter 8

Conclusion an future work

Overall, the employment of NNs for vocal audio classification in the context of Parkinson's disease holds immense potential for aiding in early detection, monitoring, and intervention strategies for individuals affected by this neurodegenerative disorder.

The RNN have shown shallow feature extration power, reaching low values of accuracy both with Mel spectrogram and MFCCs, making them unsuited for this kind of application.

Higher accuracy is been reached with LSTM, in particular using Mel spectrograms. Nonetheless, the computational time required to train the network is impairing, resulting in an unfit solution for the problem.

Regarding the VGG16, the accuracy reached by this model is the highest when the Mel spectrograms are used, even if the original data set is very different from the one we used. Still, it suffers of the same liability of the LSTM reaching extremely high computational time.

The 2-D CNN created from scratch reaches slightly worse but still remarkable results with both transformation, with a medium computational complexity.

1-D CNN combines a good level of deepness with an acceptable computational complexity. The results achieved by the 7 kernel model on the Mel spectrograms are perfectly comparable with the LSTM one, requiring at the same time a quarter of the time to train.

As a comparison between the two transformation analyzed in the paper, it is possible to observe that Mel spectrogram achieves better results on some networks, but MFCC are slightly faster to train due to a smaller input size.

Overall, this research underscores the potential of NNs in vocal audio classification for Parkinson's disease. The 1-D CNN model emerged as a promising solution, striking a balance between accuracy and computational efficiency. Further refinement and optimization of these models could greatly contribute to the early detection and management of Parkinson's disease.

Bibliography

[1] Giovanni Dimauro and Francesco Girardi. Italian parkinson's voice and speech. 2019.

[2] Jinee Goyal, Padmavati Khandnor, and Trilok Chand Aseri. A hybrid approach for parkinson's disease diagnosis with resonance and time-frequency based features from speech signals. *Expert Systems with Applications*, 182:115283, 2021.

[3] Máté Hireš, Matej Gazda, Peter Drotár, Nemuel Daniel Pah, Mohammod Abdul Motin, and Dinesh Kant Kumar. Convolutional neural network ensemble for parkinson's disease detection from voice recordings. *Computers in Biology and Medicine*, 141:105021, 2022.

[4] Hunter EJ Spielman J Ramig LO Little MA, McSharry PE. Suitability of dysphonia measurements for telemonitoring of parkinson's disease.

[5] Danish Rizvi, Iqra Nissar, Sarfaraz Masood, Mumtaz Ahmed, and Faiyaz Ahmad. An lstm based deep learning model for voice-based detection of parkinson's disease. 29:337–343, 01 2020.

[6] McSharry PE Ramig LO Tsanas A, Little MA. Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average parkinson's disease symptom severity.