Master thesis

Degree of Master of Science in Engineering, Computer Science and Engineering, 300 credits

Exploring State-of-the-Art Machine Learning Methods for Quantifying Exercise-induced Muscle Fatigue

Computer Science and Engineering, 30 credits

Halmstad 2023-08-27

Abboud Afram & Danial Sarab Fard Sabet

HALMSTAD UNIVERSITY

ABSTRACT

Muscle fatigue is a severe problem for elite athletes, and this is due to the long resting times, which can vary. Various mechanisms can cause muscle fatigue which signifies that the specific muscle has reached its maximum force and cannot continue the task. This thesis was about surveying and exploring state-of-the-art methods and systematically, theoretically, and practically testing the applicability and performance of more recent machine learning methods on an existing EMG to muscle fatigue pipeline. Several challenges within the EMG domain exist, such as inadequate data, finding the most suitable model, and how they should be addressed to achieve reliable prediction. This required approaches for addressing these problems by combining and comparing various state-of-the-art methodologies, such as data augmentation techniques for upsampling, spectrogram methods for signal processing, and transfer learning to gain a reliable prediction by various pre-trained CNN models.

The approach during this study was to conduct seven experiments consisting of a classification task that aims to predict muscle fatigue in various stages. These stages are divided into 7 classes from 0-6, and higher classes represent a fatigued muscle. In the tabular part of the experiments, the Decision Tree, Random Forest, and Support Vector Machine (SVM) were trained, and the accuracy was determined. A similar approach was made for the spectrogram part, where the signals were converted to spectrogram images, and with a combination of traditional- and intelligent data augmentation techniques, such as noise and DCGAN, the limited dataset was increased. A comparison between the performance of AlexNet, VGG16, DenseNet, and InceptionV3 pre-trained CNN models was made to predict differences in jump heights.

The result was evaluated by implementing baseline classifiers on tabular data and pre-trained CNN model classifiers for CWT and STFT spectrograms with and without data augmentation. The evaluation of various state-of-the-art methodologies for a classification problem showed that DenseNet and VGG16 gave a reliable accuracy of 89.8 % on intelligent data augmented CWT images.

The intelligent data augmentation applied on CWT images allows the pre-trained CNN models to learn features that can generalize unseen data. Proving that the combination of state-of-the-art methods can be introduced and address the challenges within the EMG domain.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to our supervisor, Jens Lundström, for his invaluable guidance, support, and mentorship throughout the duration of this project. His expertise, constructive feedback, and dedication have played a crucial role in shaping our research and pushing us to achieve our best.

We would also like to thank our opponent, Atiye Sadat Hashemi, for her insightful comments, thoughtful questions, and constructive criticism while evaluating our work. Her perspectives and challenging discussions have greatly contributed to the refinement of our research.

Kind Regards,
Abboud Afram & Danial Sarab Fard Sabet
Halmstad, Sweden
June, 2023

CONTENTS

1 INTRODUCTION 1
2 PROBLEM FORMULATION 5
2.1 Research questions 5
2.2 Contribution 6
3 RELATED WORK 9
3.1 EMG Device 9
3.2 Feature Extraction 9
3.3 Spectrogram Methods 11
3.4 Transfer Learning 12
3.5 Data Augmentation 13
3.6 Summary and comparison 15
4 THEORY 17
4.1 Signal Processing 17
4.1.1 SEMG Features 18
4.1.2 Spectrogram 20
4.2 Data Pre-processing 20
4.3 Data Upsamling 21
4.3.1 Generative Adversarial Network 21
4.3.2 Synthetic data using GAN 22
4.4 Transfer Learning Methodology 22
4.4.1 Pre-Trained models 23
5 METHOD 25
5.1 The Experimental Setup 25
5.2 Data Format 26
5.3 Data Description 26
5.4 Signal Processing methods 27
5.4.1 Baseline Signal Processing 27
5.4.2 Spectrogram 28
5.5 Pre-Processing 29
5.6 Training Baseline Classifiers 31
5.7 Data augmentation 31
5.7.1 Noise 31
5.7.2 DCGAN 32
5.8 Pre-trained Models 35
6 RESULTS 37
6.1 Baseline Classifiers 37
6.2 Hardware Component 37
6.3 Spectrogram and Data Augmentation 38
6.3.1 CWT and STFT Spectrograms 38
6.3.2 Traditional- and Intelligent Data Augmentation 38
6.4 Pre-Trained CNN Models 40
7 DISCUSSION 49

7.1 Comparison of the Spectrogram methods 50
7.2 Learning Plots and Confusion Matrices Analysis 51
7.3 Improvement of accuracy using data augmentation 52
7.4 Comparison of traditional- and intelligent data augmentation 53
7.5 Comparison of the pre-trained CNN models 53
7.6 Dataset 54
7.7 Adressing the research questions 54
7.8 Future works 55
7.8.1 The choice of target variable 55
7.8.2 Choice of datasets 55
7.8.3 Hyperparameter tuning 56
7.8.4 Other spectrogram methods 56
7.8.5 Transfer learning methodologies 56
8 CONCLUSION 57
A APPENDIX 59
BIBLIOGRAPHY 61

LIST OF FIGURES

Figure 1 Shows images before (a) and after signal processing (b), where yellow represents knee extension, pink represents wall squat, and green, red, and cyan represent the three jumps during the squat jump. 18
Figure 2 DCGAN adversarial game 22
Figure 3 Experimental setup during this thesis. 25
Figure 4 Heatmap showing the correlation between the manually extracted features. 31
Figure 5 DCGAN generator architecture 33
Figure 6 Example of real (a) and noise augmented (b) CWT images with a combination of multiple noise factors. 38
Figure 7 Example of real (a) and noise augmented (b) STFT images with a combination of multiple noise factors. 39
Figure 8 Example of real (a) and fake generated (b) CWT images using DCGAN. 40
Figure 9 Example of real (a) and fake generated (b) STFT images using DCGAN. 40
Figure 10 Training and validation learning curve for STFT without data augmentation. 43
Figure 11 Confusion matrix for STFT without augmentation. 43
Figure 12 Training and validation learning curve for CWT without data augmentation. 44
Figure 13 Confusion matrix for CWT without augmentation. 44
Figure 14 Training and validation learning curve for STFT with noise augmentation. 44
Figure 15 Confusion matrix for STFT with noise augmentation. 45
Figure 16 Training and validation learning curve for CWT with noise augmentation. 45
Figure 17 Confusion matrix for CWT with noise augmentation. 46
Figure 18 Training and validation learning curve for STFT with DCGAN augmentation. 46
Figure 19 Confusion matrix for STFT with DCGAN augmentation. 47

Figure 20 Training and validation learning curve for CWT with DCGAN augmentation. 47
Figure 21 Confusion matrix for CWT with DCGAN augmentation. 48

LIST OF TABLES

Table 1 Description of the acquired feature after signal processing. 28

Table 2 Description of generator architecture and layers. 33

Table 3 Description of the discriminator architecture and layers. 34

Table 4 Evaluation of the baseline classifiers for Experiment 1 37

Table 5 Data augmentation, run time, and accuracy of the different pre-trained models for STFT spectrograms. 41

Table 6 Data augmentation, run time, and accuracy of the different pre-trained models for CWT spectrograms. 42

ACRONYMS

EMG Electromyography

SEMG Surface Electromyography

STFT Short Time Fourier Transform

CWT Continuous Wavelet Transform

SVM Support Vector Machine

CNN Convolutional Neural Network

GAN Generative Adversarial Network

DCGAN Deep Convolutional Generative Adversarial Network

BCE Binary Cross Entropy

CSV Comma Seperated Values

SGD Stochastic Gradient Descent

1 INTRODUCTION

Today's exercise and training have become important in many people's daily lives. To track how well the training was done, many wearables have allowed people to gain insight into their health and progression. Training has positively affected young and old individuals regarding mental and physical health [47][29]. High-performing elite athletes gain a competitive advantage when predicting muscle fatigue before its occurrence. This prevents overly long resting time, which gave various business insights for companies to develop a device that notifies the user if the training session can continue or if rest is necessary to prevent muscle fatigue.

Techniques today are implemented to prevent or to get an understanding of muscle fatigue occurrence. In human-computer interactions, sports injuries, performance, biomechanics, and prosthetics, the identification and analysis of muscle fatigue provide critical information. Automated systems that can forecast and identify when fatigue arises are very helpful in sporting situations where muscle fatigue can result in harm. These systems would direct the user's training by serving as a warning mechanism prior to exhaustion while maintaining an ideal fatigue state, fostering progress, and preventing needless tension on the muscle to prevent damage [2]. Muscle fatigue can be caused by various mechanisms and signifies that the specific muscle has reached its maximum force and does not have the ability to continue the task. The recovery time can vary; nevertheless, it takes a significant time for the muscle to regain its normal functions to perform a task. Localized muscle fatigue is harmful and can lead to major injury when the level of fatigue is high. However, it can occasionally be helpful during muscle growth. Fatigue muscles absorb less energy before being stretched to where an injury occurs [2]. Muscle fatigue occurs when protein-generating processes become impaired and can be classified into two categories, acute- and chronic fatigue [13]. The treatment for acute fatigue is to rest, while chronic fatigue is a term of illness or long-time condition [60]. A deeper focus on a solution for predicting acute muscle fatigue will be performed during the following thesis.

There are two ways the devices can be used to measure muscle activity data from a specific muscle, non-invasive or invasive, and these methods differ significantly. The invasive technique is mostly used for diagnostic purposes and involves entering the desired muscle using a needle-type electrode via the skin. In comparison, the invasive technique causes discomfort to the user, while the non-invasive technique lessens the patient's suffering and makes the device portable. The non-invasive technique has the advantage of applying mounted electrodes on the desired muscle. Both of these techniques have their advantages and disadvantages. High-resolution data and precise localized descriptions of muscle activity are produced by the invasive approach, while the sample rate and segment length affect these factors for the non-invasive approach [52]. During this thesis, the non-invasive technique will be the center of attention mainly due to the device used for gathering data and for comfortable daily usage.

Different methodologies can quantify fatiguing of muscles in a non-invasive way. Ultrasound is one of the methods used in sonomyography (SMG) to describe the structural and morphological alteration in skeletal muscles. According to a study, a review of non-invasive techniques to detect and predict localised muscle fatigue written by Al-Mulla and others, the SMG method can be combined with another method to gather more information about the muscle. Near-infrared spectroscopy (NIRS) detects changes in the blood flow which is crucial during muscle contraction to measure the absorption of the protein hemoglobin. However, NIRS has been proven to understand or analyze the oxidative metabolism in healthy muscles. Additional research is necessary to determine how it may be utilized to study muscle fatigue. Mechanomygraphy (MMG) is an additional method that records both the muscle vibrations as the fibers move and the mechanical signals of a contracting muscle. This method can also be combined with an additional method for a deeper analysis of muscle activity. However, MMG is assumed to be unsuited for the analysis of dynamic muscle contraction due to the number of variables influencing the complicated signals and challenges with consistency when used repeatedly to quantify fatiguing of muscles [2]. The Electromyography (EMG) technique is a newly implemented cheap and easy-to-use solution to predict or detect muscle fatigue in various cases. The sensors called surface EMG (sEMG) offers essential information about the changes in the muscles, and data can be gathered from the sensor signals to analyze localized muscle fatigue. Muscle electrical activity during contraction is recorded using EMG to detect muscle abnormalities [36]. Measuring the time- or frequency domain variation in EMG signals brought on by fatigue is possible. EMG is well-chosen hardware during this thesis because it is applicable for capturing information regarding acute muscle fatigue. EMG is becoming more frequent in physiology, and this is due to the noninvasive, real-time, and multitarget measurement benefits [61][2].

Feature extraction is a crucial part of using any machine-learning solution. There are various feature extraction methods for extracting features from the sEMG signals for predicting muscle contractions and fatigue [24]. The extracted features can then be adapted to the chosen machine learning algorithm of choice [30]. Currently, several baseline methods are known for calculating fatigue indicators from EMG data. There are two primary groups of fatigue algorithms used. The first group consists of traditional frequency-based methods that quantify changes in the spectrum. This group includes the traditional mean frequency (MNF), spectral moments ratio (SMR), and wavelet-based techniques. The second group of algorithms for detecting muscle fatigue considers additional nonlinear elements like recurring patterns or fractal measurements of the signal. This group includes recurrence quantification analysis (RQA) and entropy-based techniques like sample entropy (SampEn) and Fuzzy approximation entropy (fApEn) [22]. Time-frequency methods can be combined to process non-stationary and time-vary signals. This combination could be suitable for predicting muscle fatigue and the estimation process for the obtained data. This method is a non-machine learning algorithm but a mathematical technique mainly used to analyze the signals frequency content. Even though these mathematical methods are adequate for the task, there are more advanced algorithms for muscle fatigue prediction [53]. The machine learning Naïve Bayes, random forest, SVM, and rotation forest can be mainly used for classification tasks for different domains. These techniques can enhance the automated system's architecture for the sEMG signals [24]. The machine-learning algorithms logistic regression, SVM, and artificial neural network (ANN) can process sEMG signals to classify and recognize motion patterns which could be adapted and implemented for this thesis to predict various muscle fatigue phases [69]. Unsupervised machine learning is not a popular method for muscle fatigue prediction tasks.

During the work of this master thesis, cooperation with the company Innowearable was done to surveying and explore state-of-the-art methods and systematically, theoretically, and practically test the applicability and performance of more recent machine learning methods on an existing EMG to muscle fatigue pipeline. This investigation will be done by conducting seven experiments. The minor experiment consists of running the baseline methods, such as decision tree, random forest, and SVM, using the processed limited tabular data as input for the models and difference in jump heights as a target variable indicating the muscle performance, which is considered a crucial factor for muscle fatigue prediction. The majority of the experiment consists of utilizing STFT and CWT spectrogram images and performing traditional- and intelligent data augmentation, noise, and DCGAN, then implementing various pre-trained models, such as AlexNet, VGG16, DenseNET, and InceptionV3, using transfer learning for comparison in performance, further discussed in Chapter 2. Innowearable offers various innovative wearables consisting of sensors to prevent injury and enhance performance for athletes[1]. Our part in their company is to help them with the collected data from their device, Inno-X, from pre-determined isometric exercises for the prediction of muscle fatigue.

2 PROBLEM FORMULATION

The usage of EMG signals for muscle fatigue prediction is a broad subject. Numerous research studies and experiments have been done regarding utilizing machine learning and deep learning techniques to process EMG data and predict muscle fatigue in human muscles. The central part of this thesis consists of the chosen research field to combine various techniques regarding muscle fatigue stages in the thigh, which implies studying various related works to analyze different methodologies used by authors. These methodologies can vary depending on data complexity and limitations.

As mentioned in Chapter 1, the sEMG is a common sensor to predict and detect muscle fatigue. However, there are several challenges regarding this thesis's research, such as limited data and finding the most suitable pre-trained model, which led to focusing on addressing these challenges and combining various methodologies. These challenges could be solved by studying different machine learning algorithms for similar tasks, reproducing and comparing these techniques, and applying them to the acquired data. Various research papers have stated the challenge of inadequate data in the EMG domain and how it is addressed by adapting transfer learning (discussed later in Chapter 3). Research presented approaches for solving the problem of insufficient EMG data by adapting data augmentation techniques for upsampling the EMG dataset, which is then evaluated using added noise and deep learning networks, as discussed in Section 3.5. The amount of acquired data in this thesis is insufficient, creating a challenge regarding implementing state-of-the-art solutions and baseline methods. Due to the absence of done and proven approaches that combines the above solutions, this thesis aims to further investigate and conduct experiments in this area. The present study aims to survey and explore state-of-the-art methods and systematically, theoretically, and practically test the applicability and performance solution for fatigue prediction with limited datasets by investigating and integrating various state-of-the-art methodologies. To achieve this objective, an evaluation of the positive and negative aspects of the proposed solution will be conducted. Furthermore, future research directions and potential improvements will be proposed.

2.1 RESEARCH QUESTIONS

The research aims to conduct seven experiments, where six (main contribution) consists of transforming the 1D EMG signals to the time-frequency domain as spectrogram images using various methods. These images will be used to fine-tune a pre-trained CNN model to be reused and adapted for predicting muscle fatigue. Different image datasets is fed into the pre-trained model and consist of limited images and an up-sampled dateset of augmented images for comparison. The experiment setup of the thesis is shown later in Section 5.1. The research objectives also involve exploring the following questions:

- Can state-of-the-art methods, such as transfer learning, STFT and CWT spectrogram techniques, and traditional- and intelligent data augmentation methods, be combined and introduced to the EMG field, if they can, how do they perform compared to each other and to the baseline methods?
- How do various pre-trained models perform when using traditional- and intelligent data augmentation techniques on time-series data?
- How do the pre-trained models perform on different spectrogram methods?

To achieve our goal, further study and research are required regarding the subject of muscle fatigue and EMG hardware component, together with determining the field of research to obtain an efficient solution within muscle fatigue prediction. The analysis should cover how the data is processed using different methods, comparison between traditional- and intelligent data augmentation techniques, the implementation of the pre-trained models, evaluations and comparison of the experiments, and the results.

2.2 CONTRIBUTION

The primary contribution of this thesis lies in surveying and introducing state-of-the-art methods within the EMG domain partly by addressing the challenge of limited dataset availability for muscle fatigue prediction through the integration of data augmentation techniques and if transfer learning can be beneficial when utilizing synthetic data. Specifically, this thesis discusses the possibility of upsampling a small dataset using traditional- and intelligent data augmentation techniques to adapt various pre-trained models for muscle fatigue prediction. This is done by conducting the mentioned experiments to compare and discuss the various methodologies. This approach has the potential to improve the accuracy and reliability of muscle fatigue prediction models in scenarios where only a limited amount of data is available. The pre-trained model's purpose is to demonstrate that the image dataset can be utilized in a pre-trained model without the need to construct and train a new model with a large dataset. This could reduce the time and computational resources required for model development, which is particularly important in practical applications. Moreover, the proposed methodology has the potential to reduce the data acquisition requirements for similar experiments in the future. This could be particularly useful in artificial intelligence within healthcare or other contexts where obtaining large datasets may take time and effort. The results can be affected by some factors mentioned below [60][56]:

- Skin preparation before placement of the sEMG.
- The placement of the sensors is essential for correct reading.
- Biased values between each person's state of physical routine and gender.
- Choice of clothes during exercises.

If successful, this thesis would introduce and represent an advancement in state-of-the-art muscle fatigue prediction, enabling more accurate and effective prediction models to be developed with a reduced data acquisition burden. This contribution could have important implications for developing more efficient and practical solutions for predicting muscle fatigue.

3 RELATED WORK

The literature review for the thesis is broad, and the field of research is significant. The following section presents a description of the device, state-of-the-art techniques and methodologies addressing the problem of muscle fatigue prediction.

3.1 EMG DEVICE

According to Reaz and others, EMG can be referred to as myoelectric activity due to the electrical potential produced by the muscle tissues [43]. As Wang suggests, muscle fatigue can be revealed by studying the EMG hardware and states the importance and aspects of using EMG in sport and science research [61]. According to Fuentes del Toro and others, the main reason for using EMG is low cost and availability to athletes because the equipment used to detect fatigue today is expensive. Muscle fatigue is localized into two levels: fatigue and non-fatigue. Fatigue, as mentioned in Chapter 1, states the decline in muscular capacity during contraction. Non-fatigue is associated with the state of the muscle during contraction before the occurrence of fatigue. The authors mention the usability of sEMG for evaluating muscle activity and gathering information about the specific muscle [56]. In other words, a methodology of recording and acquiring information from the specific muscle during exercises [43]. Chowdhury and others mention that the recorded signals from the EMG attached to the muscle can produce various noises which can corrupt the acquired data [7], which will be further discussed later in section 4.1. According to Fuentes del Toro and others, the crucial factors when using sEMG for evaluating muscle activity is the necessity of healthy subjects to avoid abnormal fatigue outcome and knowledge about the generated signals. A descriptive statistical analysis was used to ensure the validation of the data [56].

3.2 FEATURE EXTRACTION

Marotta and others state that EMG is better suited for acute fatigue than cumulative fatigue regarding capturing information. The difference between these fatigue stages is that the subjects perform physical exercises rather than getting affected by physical or mental stressors. The preparation of data is a critical step for obtaining a functional solution. This step consists of an evaluation of feature importance and selection. The data preparation process will benefit the model in predicting muscle fatigue efficiently. According to Fuentes del Toro and others, the isometric contraction measured from the sEMG is the main focus for most researchers. These measurements are obtained mainly using mean frequency or median frequency features regarding muscle fatigue evaluation [56]. Jamaluddin and others had a similar approach where features from intensive training were extracted using sEMG. To gain good prediction accuracy, the extracted features were mean frequency, median frequency, root mean square, and mean absolute value [21]. The two articles have similar features extracted for the task. The features extracted from our raw data contain mean frequency, median frequency, and root mean square from three isometric contraction exercises. However, additional features are extracted and adapted to the exercises instead of the mean absolute value. The representation of the non-fatigue and fatigue stages are determined depending on the value changes of each feature. Using the Bayes decision theory classification method based on the chosen features, the authors achieved an excellent prediction accuracy [21].

Training a machine learning model starts with preparing and analyzing the data, followed by extracting the features and building the model. Many feature extraction methodologies and algorithms are generally used for this purpose. According to Zang and others, the sEMG signal exhibits unpredictable and non-stationary variations, mainly because the signal from the sEMG depends on the muscle contraction during movement or activity, leading to analysis challenges. Because of this, it is difficult for traditional machine-learning algorithms to evaluate such complicated data. The authors describe how end-to-end feature extraction is easier using deep learning and the possibility of doing that in one training process. The authors present a new Multi-dimensional Feature Fusion Network (MFFNet) method for feature extraction and detection of muscle fatigue based on measurements of EMG signals. This is done by combining CNN, Bidirectional Long Short-Term Memory (BiLSTM) Networks, and attention models. The authors mention the usage of two different datasets. The first consists of over 18000 data from two healthy subjects, and the second consists of over 90000 data from ten healthy subjects that performed three exercises three times. Zang and others designed three attention networks with customized classification loss trained on specific hardware designated for the acquired data. The authors state in the article that their methodology "solves the shortcoming of traditional machine learning algorithm in feature extraction and cross-subjects". However, the authors mention that for a performance improvement, exploration of optimization algorithm and cross-subject analysis is necessary [68].

According to Duan Na and others, feature extraction for measurements from sEMG is a critical challenge for gaining correct muscle movement detection. Duan Na and others explain the usage of

Convolutional Neural Networks (CNN) to extract features from the sEMG and describe the benefits of using CNN to solve these types of problems. The authors constructed their own CNN architecture and trained the network by adding multiple convolutional-layer combined with various layer types and adapted parameters. The network was mainly built for hand-motion recognition [12]. However, in our thesis, a deeper focus on how pre-trained CNN models are utilized is discussed later in section Section 3.4 to predict muscle fatigue. In their article, Duan Na and others mention the usage of spectrogram technique and how the EMG channels are converted into a spectrogram using frequency and time dimensions [12]. Similarly to this thesis, the EMG signals will be converted into images using various spectrogram techniques to utilize multiple pre-trained models and then to classify muscle fatigue, which is part of the experiments. Zawawi and others mention that signal processing using a spectrogram is better because the technique can represent the signals in time-frequency representation [66].

3.3 SPECTROGRAM METHODS

The spectrogram is used to preprocess the raw data, and there are various spectrogram techniques to convert various types of collected data to images to extract useful information. The distribution of the data over time is represented and visualized using spectrograms [28] [17]. The spectrogram images can then be input for the pre-trained model for classification tasks. Lynn and others mention that spectrogram is an important and common method in applications focused on time-varying spectral analysis. The authors mean that the spectrogram is an STFT squared magnitude and is a well-performed method regarding peak-to-side lobe ratio (PSLR) and main-lobe width (MLW). However, in the article, the spectrogram is compared with the multi-windowed spectrogram (MW), and the authors state the benefits and drawbacks of the methods [31]. The data collected from the muscle using the sEMG consist of time-series data. According to Ghaderpour and Pagiatakis, time series data can contain variability of frequency over time together with high and low amplitude, which allow the utilization of two time-frequency methods, namely STFT and CWT [15]. Zhang and others state that STFT is the most common method because of the expansion of the signal in two dimensions [67]. Ghaderpour and Pagiatakis mention that STFT was introduced because of the capability to analyze both equally and stationary-spaced time series. According to Zawawi and others, the window size of the spectrogram can vary, and the authors state that a window size set to 512 is most suitable for EMG signals [66]. Ghaderpour and Pagiatakis mention that the well-known technique CWT computes the wavelet functions translation and dilation through a scalogram which, through various approaches, can be converted to a spectrogram. However, the authors state the importance of appropriate wavelets to prevent poor results from an obtained spectrogram [15]. Karlsson and Gerdle mean that CWT is a suitable method for EMG signals for stationary and non-stationary conditions mainly because of the enhancement in resolution in common parameters and detailed analysis [23]. Because of the variety of spectrogram methods, a deeper focus on STFT and CWT methods for time-series data will be fully explored and compared during this thesis.

3.4 TRANSFER LEARNING

Weiss and others mention that in some real-world machine-learning cases, data collection can be difficult or expensive, requiring a trained high-performance learner that can obtain data in different domains, namely transfer learning. The paper states that "the need for transfer learning occurs when there is a limited supply of target training data." [62]. Transfer learning has been successful in human activity classification, allowing us to adapt a transfer learning technique to work with the limited amount of data gathered from the sEMG device. Demir and others mention in their article that current deep learning development is called transfer learning and involves sharing or transferring the knowledge of one network's prior training to another network's fine-tuning of feature extraction [11]. Demir and others describe using deep transfer learning-based physical action classification for EMG signals. The authors represent the acquired EMG signals as time-frequency images (TFI) used as input for the implemented pre-trained networks. The pre-trained deep learning networks VGG16 and AlexNet were used for deep feature extraction. Demir and others mention that the models used for the task are trained on millions of images which is far more efficient than implementing and constructing the network architecture from scratch. During their experiment, a dataset from the UCL machine learning repository with 10000 images was used, containing ten exercises performed by four subjects. With the pre-trained model, the authors achieved good accuracy when compared with different results. However, for future work, the authors want to use or investigate recent pre-trained models such as GoogleNet, DenseNet, etc. [11].

Côté-Allen and others had a similar approach for gesture recognition. Fourier transformed-based spectrogram and CWT was used and fed into the pre-trained ConvNet model when processing the raw signals. The authors used two datasets, Myo Dataset, and NinaPro DB5, without mentioning the amount of retrieved data. The Myo Dataset contained two sub-datasets for pre-training and evaluation from 19 participants. The NinaPro DB5 dataset contained data from 54 movements performed by 10 participants. Understanding the performance of the suggested algorithm within the current state-of-the-art is crucial because the article's primary goal is to show a deep learning-based transfer learning approach to the issue of sEMG hand gesture recognition. Four distinct feature sets were chosen from the literature to provide a foundation for comparison. For the Myo evaluation dataset, the model achieved an accuracy of over 98%, while for NinaPro DB5, an accuracy of about 68% was achieved due to the model learning general features. To use labeled data for long-term classification, Côté-Allen and others want to perform experiments for the transfer learning algorithm's application for inter-session classification [10]. Considering the methodologies and perspectives of the articles contributes a superior summary of the spectrogram technique on the EMG data and the utilization of a pre-trained model to predict muscle fatigue [11] [10]. Limited data is the main challenge during this research, requiring recent pre-trained models combined with the spectrogram technique to be knowledgeable and successful. However, after studying these articles and noticing that the least required amount of data is around thousands, data augmentation is crucial and necessary to implement the state-of-the-art solution.

3.5 DATA AUGMENTATION

The amount of data received from the company needs to be increased for transfer learning. Thus, data augmentation will be crucial for this thesis's achieved results. Even if a significant amount of data could be achieved by data augmentation, we aim to gather enough training data to benefit from transfer learning technology. Van and others state that data augmentation refers to methods in which unobserved data or latent variables construct iterative optimization or sampling algorithms [59]. Shorten, and others mean that many application domains cannot access a significant amount of data, and the risk of overfitting is high on the training data [48]. Both articles state that the data augmentation technique is the solution for limited datasets and for overfitting [59] [48]. When the size of the training dataset is increased and enhanced, the utilization of deep learning models can be better suited for the task. Shorten and others mention that transformations such as random cropping, horizontal flipping, and color space augmentation can increase the training size [48]. Bahmei and others mention in their article that data augmentation is an important strategy to introduce variability and availability of new data in different domains to train models. In their article, the authors mention traditional augmentation, such as shifting and noise similar to Shorten and others [48] [3]. Noise augmentation can be utilized to apply noise frequency to the raw data, leading to increased training data. Yazgac and Kirci state that noise augmentation is done by multiplying noise amplitude with a random noise factor, leading to the original signal being added with the Gaussian noise [63]. However, Bahmei and others mean that intelligent data augmentation, GAN, is also beneficial for producing synthetic data to increase training data size [3]. The structure of the GAN and the model used during the following thesis will be further explained in section 4.3.1.

Tsinganos and others have implemented a similar approach regarding data augmentation of the EMG domain for hand gesture recognition due to inadequate data availability. Various strategies regarding data augmentation are presented for EMG signals, and the challenges with maintaining correct labels on the generated data are mentioned. The authors note the different categories of data augmentation, basic image manipulations, and deep learning approaches and how few studies exist for utilizing data augmentation for a specific task. During the authors' experiment, the process presents the various data augmentation tools regarding EMG signals and the comparison. Five augmentation tools were used in the article to compare standard methods and methods that take time- and frequency domains into consideration. The data augmentation tools evaluated in the article were Gaussian Noise (GN), Magnitude Warping (MW), Wavelet Decomposition (WD), and two random combinations of the mentioned methods, Augmentor.All (AA) and Augmentor.Random (AR). The authors concluded that WD, MW, and GN augmentations gave beneficial accuracy results [58].

Mendes and others implemented the deep learning approach by utilizing GAN for EMG data augmentation [35]. Shorten and others introduce GAN-based data augmentation as "the practice of creating artificial instances from a dataset such that they retain similar characteristics to the original set." [48]. Mendes and others mention challenges recurring in the EMG domain, and they tackle this problem by creating synthetic data using GAN. The authors use a developed model based on data augmentation for Parkinson's disease. Instead, the model's input would consist of the raw data collected from the EMG frequency spectrum. The model was trained and evaluated on raw data from the sEMG of one subject, and by utilizing the sliding window technique, the model generated one second of synthetic data. Mendez and others conclude their work by stating that the distribution of the generated data differs from the real EMG signals. However, the generated data increases the performance by including variability in the dataset. The authors train each channel independently for each model, and they state a potential improvement by designing an architecture that generates multiple channels at once. Mendes and others mention that their approach can be combined with transfer learning [35].

Chatzigapi and others propose a similar approach regarding utilizing GAN to upsample speech emotion data to prevent misbalance issues. The authors' method corresponds to our tasks, where spectrograms are used and normalized to convert the raw data into images to utilize a GAN model to generate synthetic spectrogram images. The authors implement Balancing GAN (BGAN) with modified layers to achieve high-quality spectrogram images. The synthetic data is compared with various augmentation techniques to prevent data imbalance and evaluated using a pre-trained CNN network as a classification algorithm. The CNN network determines and evaluates the performance of the augmentation techniques used. The authors concluded their work by improving their synthetic data produced by the BGAN and mentioned that the model has the ability to generate as many spectrograms as necessary for the imbalanced class [5]. All the articles mentioned various solutions regarding data augmentation on limited data, and the results look promising for this task. This thesis will focus on intelligent data augmentation, such as the GAN data augmentation technique, and traditional data augmentation techniques, such as adding noise to the raw data.

3.6 summary and comparison

Summarizing the related work, the first step is to understand the sEMG device and the signals produced by the hardware. However, the main focus will be on processing the data using spectrograms and acquiring more by utilizing traditional- and intelligent data augmentation techniques to benefit from transfer learning solutions. As discussed in Section 3.5, noise data augmentation will be compared with the GAN technique, a newly developed deep learning algorithm that shows promising results for creating synthetic data. With increased training data, the spectrogram technique will convert the raw data into images and benefit from transfer learning. The spectrogram technique has shown positive results for EMG signals in several articles mentioned. Transfer learning will be implemented to perform classification with multiple pre-trained models using the augmented data to compare the various techniques and methods.

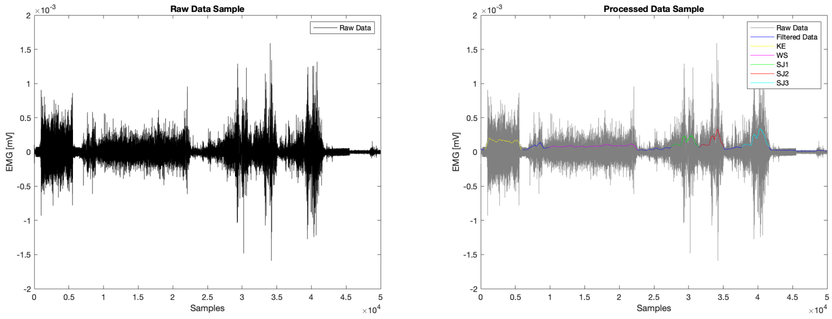
4 THEORY

4.1 SIGNAL PROCESSING

A pre-developed script in MatLab language was given for processing the signals manually from the raw data. However, this script required a significant amount of modification. MatLab is a high-performing language regarding visualization and signal processing [33]. The script was first translated from MatLab to Python, but due to the functionalities used for noise filtering and signal processing being better performed on Matlab, the Matlab script was used for this purpose.

Reaz and others mention, "All electronics equipment generates noise. This noise cannot be eliminated; using high-quality electronic components can reduce it". The authors mention the issues that occur and affect the recorded EMG signals. These issues are signal-to-noise ratio and distortion of the signal. The ratio represents the energy ratio between the noise signal and the EMG signals, while the distortion of the signal indicates that the component's contribution of any frequency to the EMG signal should not be changed. Before feature extraction is possible, it is crucial to filter, process, and analyze the signals the sEMG produces. Raez and others describe the EMG electronic noise as an important characteristic affecting the EMG and that the produced noises can be categorized [43].

Yousif and others state that EMG is developing in various fields of application. However, the signals acquired can consist of noise leading to challenges, and processing these signals before feature extraction is required to prevent file obstruction. According to Yousif and others, using various filtering methods adapted to low- or high-frequency lowers the impact of noise in the signals and the acquired data. Low-frequency noises are caused by the amplifier's direct current offset (DC), and applying a high-pass filter can reduce these noises. The authors state that high-frequency noise occurs from various factors, such as electrical interference from nearby electronic devices and muscle activity. Applying a low-pass filter allows the removal of high-frequency noise. The band-pass filter is suitable for the signals produced from the sEMG and necessary to isolate and analyze the frequency content [65].



(a) Example of raw data produced from EMG before signal processing.

(b) Example of signal processed data where the various exercises are divided into colors.

Figure 1: Shows images before (a) and after signal processing (b), where yellow represents knee extension, pink represents wall squat, and green, red, and cyan represent the three jumps during the squat jump.

4.1.1 *SEMG Features*

Feature extraction aims to acquire the most relevant data for the specific task. The data is obtained by transforming the raw signal into a data structure by removing the noise as mentioned in Section 4.1 and stated by Spiewak and others [52]. There are no exclusive strategies for feature extraction techniques. However, various mathematical formulas are widely used when interpreting the EMG signals for feature extraction, but this depends on the feature domain [9]. According to Spiewak and others, the feature domain can be divided into three categories: time, frequency, and time-frequency.

4.1.1.1 *Time Domain*

Spiewak and others state that the time domain consists of features that are used for EMG pattern recognition due to its simplicity and rapid calculation, while Yousif and others mentioned that this is achieved by filtering to prevent noise. The time domain features can be calculated according to the articles based on the increase or decrease of the amplitude of the EMG signals [52] [65]. The features can be extracted using root mean square (RMS) and mean absolute value (MAV) to detect muscle fatigue. According to Yousif and others, the observation of the amplitude is rarely used for fatigue detection. However, a combination of methods can achieve better accuracy and detection [65]

4.1.1.2 *Frequency Domain*

Spiewak and others mention that the extracted features from the frequency domain are based on power spectral density (PSD) from the estimated signal [52]. Fast Fourier transform (FFT) method can be used for calculations, and according to Cifrek and others, the Fourier transform was the earliest algorithm used to approximate the spectrums of the signals produced from the sEMG because the fast Fourier transform is computationally effective [65] [9]. The mean frequency (MNF) and median power frequencies (MPF) are the most common methods observed for changes. These methods are based on the power spectrum and are crucial for detecting muscle fatigue. The time domain features are more computationally heavy than those in the frequency domain [52]. In Section 4.1.1.1, Yousif and others mentioned that the time domain would be more reliable if combined with other methods. Thus combining the time- and frequency domains, more information regarding muscle fatigue can be obtained [65].

4.1.1.3 *Time-Frequency Domain*

Karthick and others mention, "The time-frequency method show the non-stationary nature of biomedical signals and has become an ideal tool for this analysis" [24]. According to Yousif and others, when achieving a time-frequency domain, the methods can perform a joint analysis of the various signal characteristics by analyzing the spectrum and amplitude (JASAP). The article states that the analysis of this domain consists of four stages in which force and fatigue are determined.

- Increase in amplitude and spectrum increases the mean force.
- Decrease in amplitude and spectrum decreases the mean force.
- Increase in amplitude and decrease in spectrum gives mean fatigue.
- Decrease in amplitude and increase in spectrum gives mean recovery.

The signals acquired from sEMG during muscle fatigue can then be analyzed using the functions from the time-frequency domain [65]. Marri and Swaminathan say that the signals produced by the sEMG are nonstationary and indicate nonlinear characteristics, and these challenges are addressed by using time-frequency methods [32]. Karthic and others state that the features extracted using time-frequency methods can increase the computational complexity due to the high resolution of capturing frequency components in minor variations. Thus, it is crucial to utilize suitable features for the task [24]. During this thesis, we will operate and process the signals in the time-frequency domain to analyze the signals by using the MatLab script, including various methods, such as filters and fast Fourier transforms, to process the raw data from EMG to extract the features from each exercise.

4.1.2 *Spectrogram*

The spectrogram is a widely used tool for visualizing signals, which is generated by applying the Fourier Transform to raw time-frequency signals and displaying the frequency content over time. Nodera and others explain that spectrogram techniques have been developed to improve accuracy compared to simply visualizing raw time-frequency signals, such as speech recognition, emotion detection, and music genre recognition [38]. Zawawi and others motivate the reason for using spectrograms for signal analysis and state that spectrogram is utilized due to the inability of the Fast Fourier Transform to handle the non-stationary signal with spectral properties in the time domain [66]. There are various ways to calculate the spectrogram of EMG raw signals. Canal compares the EMG signals using STFT and CWT techniques in his paper. Canal describes that the STFT managed to satisfy the non-stationary EMG signals when selecting a proper signal segment length. In contrast to STFT, which uses short windows for high frequencies and long windows for low frequencies, the CWT provides a more flexible approach by using wavelet functions localized in both time and frequency. Applying CWT to EMG signals makes it possible to obtain details and approximate coefficients of the signal. The high-frequency components of the signal are captured by the approximation coefficients, while the detail coefficients capture the low-frequency components. Canal states that the CWT is a powerful technique for analyzing the time-varying frequency content of non-stationary EMG signals [4].

4.2 DATA PRE-PROCESSING

Data pre-processing is a crucial step before training any machine learning algorithm. It consists of data cleaning, detection of outliers, feature engineering such as normalizing, and feature selection and extraction. Kotsiantis and others explain the importance of the data pre-processing step regarding training supervised machine learning models [26]. Nargesian and others explain that feature engineering involves using transformation functions, such as arithmetic and aggregate operators, on existing features to create new ones. Transformations can be useful for adjusting the scale of a feature or transforming a non-linear relationship between a feature and a target class into a linear relationship. This can simplify the learning process for machine learning models [37]. Singh and others describe the importance of data normalization among other data pre-processing techniques. The article explains that data normalization handles the transformation of the numeric feature values to a consistent range, which prevents features with larger numeric values from dominating those with smaller values [51].

4.3 DATA UPSAMLING

As mentioned in Section 5.3 later, the acquired data is insufficient to implement a deep learning network or a pre-trained model. Shorten, and others describe data augmentation as a technique to solve several challenges [49]. Thus, using the data augmentation technique is crucial to reduce overfitting and to construct synthetic data. Several studies are currently exploring solutions related to training without possessing significant labeled data. Shorten, and others mention how the synthetic data consists of minor variations without affecting the model prediction ability. Applying the data augmentation technique for our thesis will help prevent overfitting, which could occur on the limited data [49].

Fawzi and others mean that the choice of data augmentation algorithm affects the result. The authors mean that an unsuitable choice can lead to the synthetic dataset being uninformative [14]. After studying various articles [58] [35], one methodology for data augmentation gave promising results for images. There are various algorithms to upsample the training data. During this task, traditional- and intelligent data augmentation, as mentioned in Section 3.5, is implemented to create synthetic data as input for multiple pre-trained CNN models for muscle fatigue detection.

4.3.1 *Generative Adversarial Network*

GAN is a recently developed algorithm within deep learning introduced by Goodfellow and others. The authors state that the designed algorithm addresses generative modeling problems and has the ability to generate both synthetic images and text [16]. Park and others describe GAN's structure and learning process as two different networks competing. GAN consists of two networks, namely the generator and the discriminator. The competition between the networks is described as an adversarial game. The generator and discriminator have different objectives during the adversarial game. The generator tries to produce convincing fake data that the discriminator cannot separate from real data. Thus, the objective of the discriminator is to discern the data received from the generator and real data, as shown in Figure 2 below. The training idea for GAN is that the generator strives to minimize while the discriminator seeks to maximize the objective [40]. Sulaiman and Larsson mention that other types of loss functions can be adapted for the GAN network depending on the task. The authors mention that Mean Squared Error (MSE) is beneficial when the task is within the image domain [54]. However, during this thesis, binary cross entropy (BCE) loss was used as the architecture of the model was followed from PyTorch documentation and the DCGAN paper [41] [19].

Figure 2

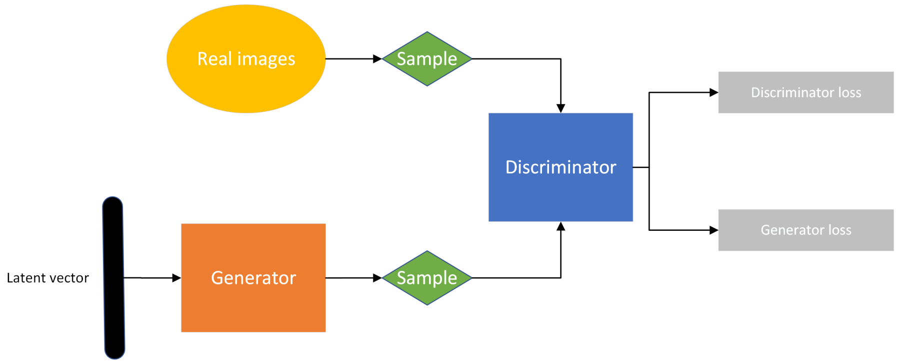


Figure 2: DCGAN adversarial game

4.3.2 *Synthetic data using GAN*

Utilizing and improving the architecture of a GAN allows the generation of synthetic spectrograms. The objective of the training of our GAN is to generate realistic spectrogram images regarding the various jump height data. The GAN architecture determines the quality of the output [5]. Pytorch owns DCGAN architecture will be developed and tuned during this thesis to produce synthetic spectrogram images. The output will be evaluated by the performance of the pretrained CNN models to determine if the synthetic data is similar to the real data. An explicit use of convolutional and convolutional-transpose layers in the discriminator and generator makes a DCGAN a straightforward extension of the GAN discussed above [19]. Radford and others introduced DCGAN with the goal of filling the gap between supervised learning's and unsupervised learning's success rates for CNNs [41].

4.4 TRANSFER LEARNING METHODOLOGY

Transfer learning is very beneficial when introduced to a new task. Transferring the knowledge from an already-trained model can be effective and an improvement. However, the model needs to be task-related for transfer learning to be beneficial or negative transfer could occur. Torrey and Shavlik mention that transfer learning can be an improvement in several aspects. Transfer learning reduces the time complexity, collecting a significant amount of data is unnecessary, and has a considerable learning curve because of the already acquired knowledge compared to training a new model with an implemented architecture from scratch [57]. Zhuang and others mean that despite its great success and effective practical uses, traditional machine-learning technology still has limitations in some situations. The transfer learning technique addresses these limitations and challenges by transferring knowledge. Torrey and Shavlik previously mentioned negative transfer [57]; Zhuang and others mean that transfer learning can be divided into two categories to adapt the methodology to task-related subjects, homogenous and heterogeneous. The authors describe homogenous transfer learning as "developed and proposed for handling the situations where the domains are of the same feature space." and heterogenous as "the knowledge transfer process in the situations where the domains have different feature space." [70]. During this thesis, the transfer learning methodology will be adapted and related to the spectrogram representation of EMG signals to classify muscle fatigue through images.

4.4.1 *Pre-Trained models*

The input images for the pre-trained models will consist of raw data with added noise that is converted to spectrograms and synthetic images of STFT and CWT spectrograms from the DCGAN model, which leads to using multiple pre-trained CNN models for classification. As mentioned in section Section 3.4, the various articles mention the variability of pre-trained models that can be used to perform image classification on the spectrogram. Demir and others mentioned some pre-trained models suitable for our task in their future work that will be considered and implemented during this thesis [11]. For the classification of the time series spectrogram, the choice is large. Raghu and others mentioned implementing multiple pre-trained models where the best-acquired accuracy was achieved using InceptionV3 [42]. Yosinksi and others state in their article that natural images are most common when training deep neural networks. Pre-trained models are implemented similarly and vary in architecture, the first layer is not specified to a task, but the models are generally adapted for many tasks. The authors state that in the last layer of the networks, the features adjust from general to specific [64]. Pytorch library offers various pre-trained models, such as AlexNet, VGG, DenseNet, and many more. During this thesis, the articles introducing and describing the various models was followed to implement the models efficiently for comparison and perform image classification of muscle fatigue on various spectrogram images [20]. The models implemented from PyTorch during this thesis were AlexNet introduced by Krizhevsky and others [27], VGG16 introduced by Simonyan and Zisserman [50], DenseNet introduced by Huang and others [18], and InceptionV3 introduced by Szegedy and others [55], these methods were mentioned in the various articles, and a comparison between the performance of these models will be made.

5 METHOD

5.1 THE EXPERIMENTAL SETUP

As shown in Figure 3, structured planning is done to perform the different experiments. The initial phase involves manually extracting features from the squat jump EMG signals and training multiple baseline models. This step sets the foundation for the subsequent experiments. Experiment 2 and experiment 3 focus on converting the EMG signals into two types of spectrograms, STFT and CWT. Once the signal content is presented as spectrograms, these images are utilized as inputs to fine-tune and train various pre-trained CNN models. Experiments 4-7 shift the focus towards exploring and adapting traditional and intelligent data augmentation techniques. Traditional techniques, such as noise augmentation, are employed to add noise to the dataset, allowing for the generation of new spectrogram images. Intelligent techniques, specifically the implementation of DCGAN, are employed to generate synthetic images. The generated images of noise and DCGAN techniques will then be used as input to the pre-trained CNN models and evaluated by the accuracy acquired. The various experiments will be compared and discussed. The data is divided into training and validation sets. The split is done randomly into 70 % training and 30 % validation. This was done both for the original data set for the experiments without data augmentation and the up-sampled data set for the experiments with the data augmentation part.

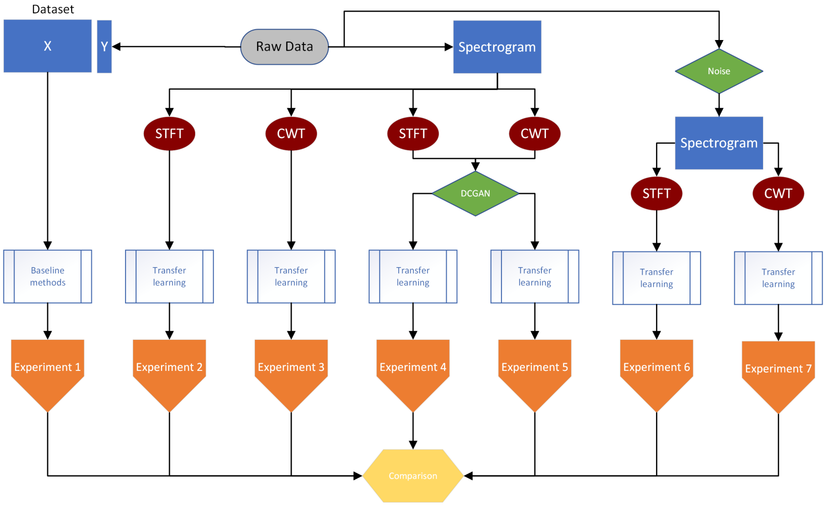


Figure 3: Experimental setup during this thesis.

The main contribution of this thesis is to utilize the spectrogram techniques to convert the EMG signals to images, utilize noise and DCGAN to upsample the image dataset and train multiple pre-trained models to compare and make a prediction on the spectrogram images. These steps will be described in the following section. Experiments 2-7 have a similar structure except for the data augmentation part, as shown in the Figure 3. However, experiment 1, consisting of the baseline method, differs from the remaining experiments, thus requiring more steps. Through the mentioned experiments, the research questions can be answered through a comparison to determine if the introduced state-of-the-art methods perform better with STFT and CWT spectrogram techniques with and without various data augmentation methods compared to the baseline methods when introduced to the EMG field. Utilizing these different techniques and methods allows the creation of new knowledge within the specific field.

5.2 DATA FORMAT

The raw data received were in the Comma Separated Values (CSV) format, containing data from an eight-channel sEMG sensor. The CSV format is simple and effective when storing data in multiple domains [8]. The raw data collected from the sensor needs to be processed and filtered, mentioned in Section 4.1 to allow the detection of the various exercises, divide them, and store them as separate CSV files. Each of the eight channels consists of sample time and frequency in voltage attributes, but the measurements are done using only one channel for this thesis.

5.3 DATA DESCRIPTION

The company gathered the raw data in real-time using a USB dongle that receives the signals from the sEMG combined with a Python script [1]. Six healthy subjects performed the training in the mentioned order, consisting of two static exercises, knee extension, and wall squat. The last exercise, the squat jump, is more dynamic, which allows for gathering jump height data. However, the files contain raw data which require signal processing. The data was gathered before (PRE), directly after (POST), and 15 minutes after (POST2), the main exercise, in which some files contained faulty signals that could not be interpreted. The main exercise consists of one hour of cycling. The exercises were done sequentially once per day until the same amount of PRE, POST, and POST2 was achieved for each person. A total of 1670 data points were gathered from each person, and files with irregular frequency were deleted to run the script for manual signal processing. Signal processing allows the retrieval and utilization of crucial information when working with hardware that produces a vast amount of noise. When the signal is processed, the divided exercises must be prepared to be trainable by various machine learning algorithms. To achieve this, further processing is required to combine and label each exercise uniquely. These steps will be explained below in Section 5.4. When the raw data was processed, the dataset consisted of 220 knee extension samples, 220 wall squats, and 660 squat jumps. The squat jump exercise consists of three jumps, thus double the amount compared to the other exercises when each jump is divided. During this thesis, the focus relies on only one exercise to prevent having multiple target variables. The squat jump exercise contained the most data samples. Therefore, this exercise is used because the measured jump height data is chosen as a target variable. After all, it indicates fatigue or non-fatigue. The target variables, such as root mean square and mean absolute values, consist of general information about the muscle and are commonly used for predicting muscle fatigue. The difference in jump height data is that it consists of specific information for predicting muscle fatigue for intense training sessions, which is of interest during this task.

5.4 SIGNAL PROCESSING METHODS

During this thesis, two different signal processing methods are implemented to conduct the experiments. To run experiment 1, a baseline signal processing methodology is used to clean, combine, and visualize the data. However, STFT and CWT spectrograms techniques are used for the remaining experiments to pre-process the raw data received from the sEMG device to implement pre-trained models.

5.4.1 *Baseline Signal Processing*

The signal processing step is essential for preparing the EMG signal for machine learning models. For this thesis, a script the company provides is used to handle the signal-processing step. The script detects the three exercises and then divides the signal into three parts respective to each exercise. The script further divides the signal into three additional parts for the squat jump exercise, each containing one jump. Different functions are executed after dividing the jump data to filter out the noise and extract the necessary features. This will result in squat jump files containing the processed values/features for each PRE, POST, and POST2 measurement per subject. The extracted features for the jump exercise include Mean Frequency, Median Frequency, RMS, and other frequency features, shown in Table 1. The target for our task variable is calculated in Section 5.5 to train the baseline classifiers.

Features	Description
Subject	Subject that performs the exercise.
Mean Freq	Mean Frequency.
Median Freq	Median Frequency.
RMS	Root Mean Square.
ARV	Average Rectified Value.
PSD Area	The area under the power spectral density.
PSD area f24	The area under the power spectral density for specific frequencies.
Dimi fatigue ind	Dimitrov spectral parameter.
BP1-BP18	Bandpower of a specific frequency band in the EMG signal.
Kurt D	Statistical measure describes the shape of the probability distribution of a signal.
Skew D	Statistical measure that describes the symmetry of the probability distribution of a signal.
Mode D	Statistical measure that describes the value that occurs most frequently in a signal.
Mean D	Statistical measure that represents the average value of a signal.
Std D	Statistical measure that represents the amount of variability or dispersion of a signal around its Mean value.
testype	Pre, Post, Post2.

Table 1: Description of the acquired feature after signal processing.

5.4.2 *Spectrogram*

The raw data must be converted to images to utilize various pre-trained CNN models. For experiments 2-7, different spectrogram methodologies are used to process the raw signals gathered from the sEMG and convert them to images. STFT and CWT are used to represent the signal strength of a signal over time at different frequencies and present it in a particular waveform.

The one channel from the sEMG consists of two properties, namely frequency and time, and by implementing the pywt and scipy library in Python, the two properties are converted to CWT and STFT spectrogram images. The STFT approach is applied to examine the time-varying spectral composition of the squat jump EMG signals. This technique offers a time-frequency signal representation and is often utilized in signal processing applications. This technique applies a short-time window to the signal and computes the Fourier transform for each window. The resultant STFT matrix represents the signal's spectral content at various time points. To acquire STFT spectrogram images from the squat jump EMG signals, the implemented Python function stft_spectrogram is used. This function takes an EMG signal as input and applies the STFT method using the scipy.signal.stft function. The window size and overlap between adjacent windows can be adjusted to optimize the resolution of the spectrogram for a particular application. In the implementation, a window size of 512 and an overlap of 128 is used. The library matplotlib.pyplot.pcolormesh is used to plot the generated STFT matrix as a spectrogram. The function stores the spectrogram picture under the supplied file name in the designated folder.

In contrast to the STFT approach, which applies a fixed window size to compute the Fourier transform, the CWT approach applies a variable window size that can adapt to the signal's frequency content at different time points. This allows for a more flexible and accurate time-frequency representation of the signal. To examine the time-varying spectral composition of the squat jump EMG signals using the CWT approach, the cwt_spectrogram function is implemented. This function takes an EMG signal as input and applies the CWT method using the Morlet wavelet in the pywt.cwt function. The scales for the Morlet wavelet are set to range from 1 to 30 to capture a broad range of frequency components in the signal. The resulting wavelet coefficients are then plotted as a spectrogram using the matplotlib.pyplot.imshow function. The spectrogram is saved as a "png" image using the matplotlib.pyplot.savefig function and stored in a specified folder.

5.5 PRE-PROCESSING

Preprocessing the data is crucial before using any baseline algorithms. Preprocessing includes a number of steps, such as data normalization, feature extraction, feature selection, and data cleaning. The dataset in this study follows the preparation methods listed below:

Loading the dataset The dataset is loaded into a data frame using the pandas library. The loaded dataset contains both numerical and categorical features.

Data cleaning The datasets are checked for missing values and replaced with either the mean or median value for the numerical features. The pandas method fillna is used for this purpose.

Data Visulization The heatmap visualization matrix from the Seaborn library is used to explore the correlation between the various features in the datasets. The heatmap shows the correlation values between each pair of features, with warmer colors (e.g., red) denoting positive correlations and colder colors (e.g., blue) representing negative correlations, as shown in Figure 4.

Data Normilization As the range of the numerical features can vary widely, they are scaled to ensure they are on the same scale. For this purpose, the MinMaxScaler class from scikit-learn is used.

Feature Selection The dataset contains several features; not all may be relevant to the classification task. For this thesis, feature selection is performed to identify the most important features for the task. For this purpose, the SelectKBest technique is used to select the top k important features. Upon applying the SelectKBest technique, the following features were identified as the most informative for the task: Mean Frequency, Median Frequency, RMS, ARV, PSD Area, PSD area f24, Kurt D, Skew D, Mean D, and Std D. These features were selected based on their ability to contribute significantly to the classification process.

Target Variable The company collected jump height data while the subjects performed the squat jump exercise. These measurements are stored in separate CSV files containing data for a specific subject. The jump data is then loaded into a pandas DataFrame. Different calculations are applied to calculate the jump height difference for each data point depending on the data type. The jump height difference is obtained for Pre data points by subtracting the jump height value of the Post data point from the jump height value of the Post 2 data point. For Post data points, the difference is calculated by subtracting the jump height value of the Pre data point from the jump height value of the Post data point. Similarly, the difference is obtained for Post 2 data points by subtracting the jump height value of the Pre data point from the jump height value of the Post 2 data point. After calculating the jump height differences, the resulting values were within the range of 0 to 9. However, to balance the data set, these values are rounded to the nearest whole number to convert them into 7 discrete sets of classes. This rounding process ensured that the values of the jump height differences were within the range of 0 to 6 (7 discrete classes) by setting a threshold. Higher classes represent a greater difference in the jump height which signifies that the subject has jumped higher before (Pre) the main exercise compared to after (Post) or 15 minutes after the main exercise (Post2), which represents a fatigued muscle. The process allows the pre-trained models to train on the dataset. Pre-train models are better adapted for classification tasks because the output of these models is the probability distribution over the classes, making these pre-trained models better suited for classification tasks. Hence the continuous values are changed to a discrete number of classes, and the data set is balanced.

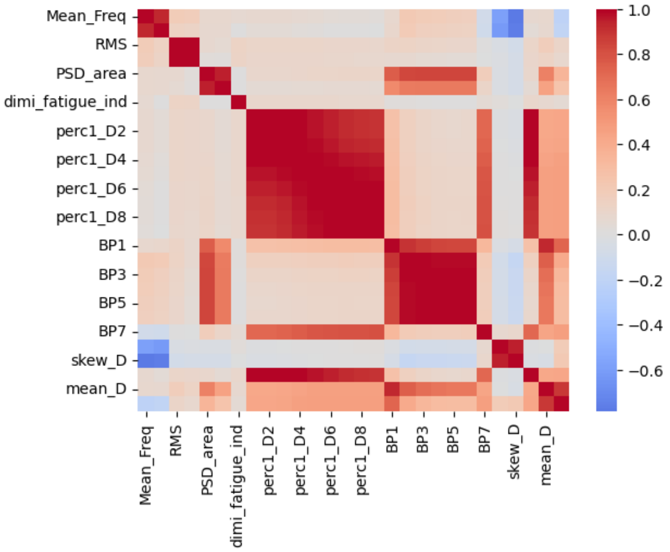


Figure 4: Heatmap showing the correlation between the manually extracted features.

5.6 TRAINING BASELINE CLASSIFIERS

After the pre-processing step, the different classifiers are trained on the pre-processed dataset with the fixed target variable, explained in Section 5.5. Three popular classifiers from literature within the EMG subject were chosen for this task, Decision Tree, SVM, and Random Forest. The dataset is divided into training and validation sets. Each classifier was trained separately on the pre-processed data using the training set and validated using the validation set. After training the classifiers, an evaluation of accuracy is acquired.

5.7 DATA AUGMENTATION

For experiments 4-7, two different data augmentation techniques are implemented for comparison, traditional- and intelligent data augmentation. These data augmentation methods differ from each other.

5.7.1 *Noise*

Synthetic signals that closely resemble real-world signals can be generated by introducing noise. In this thesis, the noise augmentation approach is employed to enhance the EMG signals. To implement the noise augmentation technique, a Python function is implemented. This function takes two inputs, the original EMG signal and the noise factor. The noise factor, a parameter crucial in this technique, determines the magnitude of the noise added to the signals. The noise factor represents a scaling factor that controls the intensity of the noise introduced to the original signal. A higher noise factor results in a more significant deviation from the original signal, while a lower noise factor produces a more subtle effect. Within the implementation, the numpy library is utilized to generate random noise. The generated noise follows a standard normal distribution (Gaussian noise). The noise factor is then multiplied by the generated noise array. Scaling the noise array with the noise factor amplifies or attenuates the noise amplitude before adding it to the original signal. This allows fine-grained control over the amount of noise applied to the signal. Finally, this is added to the original signal using element-wise addition resulting in a new augmented signal. For the experiments conducted in this thesis, noise data augmentation is applied to all squat jump signals. To generate a sufficient amount of training data for the pre-trained CNN models, noise data augmentation is performed using a range of noise factors. The noise factors used in the experiments are 0.04, 0.05, 0.055, 0.06, 0.07, 0.08, and 0.09. For each noise factor, the noise augmentation technique is applied to all signals in the original dataset.

5.7.2 *DCGAN*

The architecture of the DCGAN is inspired and followed by the DCGAN paper [41]. Before training a GAN, the image folder for a specific label of jump data containing the images is set. The images are read using a custom function that reads each "png" file, and a dataset is created by transforming the images. The transformation of the images includes resizing to 64x64, normalization, and cropping. The input and output of the networks are 3x64x64, where the three represent RGB. Hyperparameter tuning is crucial because incorrect hyperparameter tuning can lead to faulty generated images. The optimizer implemented for each network is the Adam optimizer, an effective technique for stochastic objective optimization using gradients [25]. The learning rate for the networks is set between 0.0001-0.0005, mainly due to the various labels consisting of different amounts of images leading to the networks requiring an adaption of the learning rate for the task. The beta parameters are set to beta1 = 0.5 and beta2 = 0.999 and were not changed. As mentioned in Section 4.3.1, the BCEloss function, according to Ruby and others, is an appropriate algorithm to train CNN models [44]. The length of the latent vector is set to 100 as an input for the generator and acts as a matrix multiplication. The networks are trained with a batch size of 32 during 200 epochs. A random weight initialization function is implemented and re-initializes the different network layers.

5.7.2.1 *Generator*

The generator takes the latent vector or z vector as an input and creates an RBG image as an output. The network consists of different layers as shown in Figure 5 below.

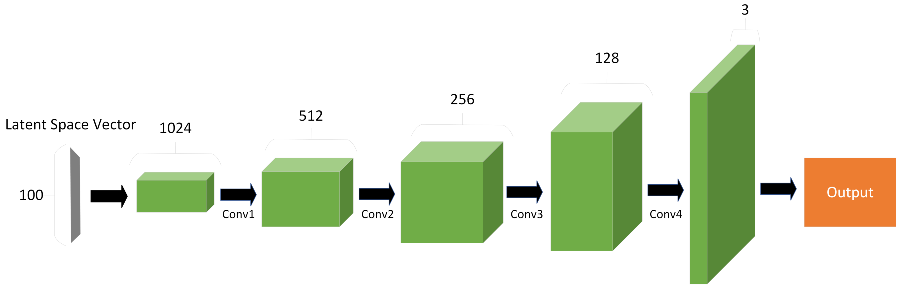


Figure 5: DCGAN generator architecture

Each convolutional-transpose layers consist of batch normalization layers, a ReLU activation function, and the output layer consists of a tanh activation function that structures the generator network.

In Table 2 below, the various layers of the generator are represented in detail.

Layer	Input	Kernel Size	Stride	Padding
ConvTranspose2D	100	4x4	2	-
ConvTranspose2D	512	4x4	2	1
ConvTranspose2D	256	4x4	2	1
ConvTranspose2D	128	4x4	2	1
ConvTranspose2D	64	4x4	1	1
Tanh	-	-	-	-

Table 2: Description of generator architecture and layers.

5.7.2.2 *Discriminator*

The discriminator works as a binary classification network, and the network takes 3x64x64 real images or fake images as input and gives the images a label as output (real or fake). Compared to the generator, the discriminator consists of strided convolutional layers. Each layer has batch normalization layers, a Leaky ReLU activation function, and the output layer consists of a sigmoid activation function, described in Table 3 below.

Radford and others' study mention that using strided convolution layers instead of the pooling layer is proposed for downsampling. This is because it allows the network to develop its own pooling function. Additionally, healthy gradient flow is promoted by using batch

Layer	Input	Kernel Size	Stride	Padding
Conv2D	64	4x4	2	-
Conv2D	64	4x4	2	1
Conv2D	128	4x4	2	1
Conv2D	256	4x4	2	1
Conv2D	512	4x4	1	1
Sigmoid	-	-	-	-

Table 3: Description of the discriminator architecture and layers.

normalization and leaky ReLU function, which is essential for the discriminator and the generator's learning process [41].

5.7.2.3 *Training*

When the structure of the networks is implemented, the random weight function is initialized, which takes the models as input and applies random weight to the layers. The learning process of the networks can be controlled using the loss function and optimizers. As mentioned previously, the BCEloss function is used for both networks and is defined in PyTorch as Equation 1 and Equation 2 below [19]:

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top \tag{1}$$

$$l_n = -w_n \left[y_n \cdot \log x_n + (1 - y_n) \cdot \log(1 - x_n)\right] \tag{2}$$

The objective of the function is to provide the calculation of both log components. With the "y" input, the part of the BCE equation applied can be chosen. This is achieved during the training loop. However, it is critical to comprehend how the component that needs to be computed is selected when changing "y". The definition of real and fake labels are defined for the calculation of the network's losses [41]. The latent vector is defined as fixed noise and is continuously an input in the generator during the training loop. It is through the noise vector an image will be developed during the iterations of the epochs. The training of the networks is divided into two parts, update of the discriminator network and generator network. According to Goodfellow and others, the goal for the discriminator is to update by ascending the stochastic gradient [16], meaning that the objective of the discriminator is to maximize the probability by labeling each image corresponding to real or fake (1 or 0). The discriminators' loss and the gradient are calculated, and an optimizer step is performed. The generator aims to maximize the log function instead of minimizing it to provide sufficient gradient and generate fake images to trick the discriminator. This is achieved during the training loops when the discriminator labels the generator output, the generator loss is calculated using the real images, the gradient is calculated, and the optimizer step updates the generator parameters. During the training, multiple generated images were saved after a specific amount of epochs and iterations to allow the generator to learn how to generate synthetic images. This is decided while printing the loss functions of the networks to determine when the synthetic images should be saved. The images were saved in folders corresponding to their labels.

5.8 PRE-TRAINED MODELS

The various pre-trained CNN models are implemented after the images are converted to spectrogram and data augmentation techniques are applied. As mentioned in Section 4.4.1, the documentation in PyTorch is followed where multiple pre-trained CNN models are implemented effectively [20].

Firstly, the necessary libraries, partly consisting of the pre-trained models, are implemented. The important step during this section is to initialize the various pre-trained CNN models, fine-tune the model and adapt it to the new task [64], update the different parameters with a defined optimizer, and finally train and validate the models. Six datasets were used in the experiments, STFT and CWT spectrogram datasets without data augmentation, STFT and CWT spectrogram datasets with noise data augmentation, and STFT and CWT spectrograms datasets with DCGAN data augmentation. The datasets are stored in separate folders, each subfolder representing a specific class. The images corresponding to each class are stored in their respective subfolders. The data transformers and loaders are initialized, and the dataset is created by transforming the images. The transformation of the images includes random resized cropping, random horizontal flipping, and hard-coded normalization.

Four pre-trained models are selected for training: AlexNet, DenseNet, VGG16, and InceptionV3. Each model is initialized with pre-trained weights obtained from the torchvision library. The input images fed into each model is upscaled to 224, while InceptionV3 upscales the input images to 299. The models' architectures are modified to suit our classification task using the function set_parameter_requires_grad. This function freezes the layers in the pre-trained models and unfreezes the last layer for fine-tuning and updating the parameters for the last layer. Each model's last fully connected layer is replaced with a new linear layer that matches the number of classes in each dataset.

After initializing the model structure correctly, the final step in fine-tuning involves creating an optimizer that selectively updates the desired parameters. It is important to note that after loading the pre-trained models, the .requires_grad attribute of all parameters is set manually to False. However, the parameters of the reinitialized layer automatically have .requires_grad equal to True. As a result, only the parameters with .requires_grad equal to True need to be optimized. To achieve this, a list of such parameters is compiled and provided as input to the constructor of the Stochastic Gradient Descent (SGD) algorithm. SGD is known for performing well with small datasets or with few training examples. SGD allows for more precise control over the learning rate and momentum parameters, facilitating the fine-tuning of the optimization process.

The training and validation process is performed for each model separately on each dataset. This is done using the train_model function. The function considers if the model to be trained is the InceptionV3. As the InceptionV3 architecture includes an auxiliary output, and the overall loss respects both the auxiliary output and the final output. The hyperparameters chosen are a batch size of 32, and 100 epochs. Two different learning rates that are tested during training, 0.001 and 0.0002. This is done to identify the one that produces the fastest convergence and the best validation accuracy. During training, the models are set to training mode to enable gradient computation and weight updates. The models are then evaluated on the validation set, where no gradients are computed, and no weight updates are performed. The training and validation iterations over the datasets, calculate the model's outputs and the loss using the specified criterion, and update the model's weights based on the gradients obtained during backpropagation. The Cross-entropy loss function is used, which is well-suited for multi-class classification tasks due to its ability to measure the dissimilarity between the predicted probabilities (network outputs) and the true labels. The running loss and correct predictions are then collected to compute the average loss and accuracy for each epoch. After each epoch, the validation accuracy is compared with the best accuracy achieved. The model weights are then saved as the best-performing model if the current accuracy outperforms the prior best accuracy obtained.

6 RESULTS

6.1 BASELINE CLASSIFIERS

After the baseline signal processing and pre-processing of the data, the classifiers were trained on the limited tabular dataset. The data was not augmented for the tabular data in Experiment 1 because of the comparison with the pre-trained models without data augmentation. The baseline methods were evaluated by training the models with the training data and acquiring accuracy from the validation data. In Table 4, the performance and evaluation of the Decision Tree, Random Forest, and SVM is shown.

Table 4: Evaluation of the baseline classifiers for Experiment 1

Classifier	Accuracy
Decision Tree	47.7 %
Random Forest	**56.8%**
SVM	50.0 %

The baseline models' run time was not considered because they ran significantly faster than the pre-trained models. As shown in Table 4, the random forest classifier achieves the highest accuracy on the limited tabular data set. However, the achieved accuracy is neither beneficial nor sufficient to differentiate between the multiple classes. This signifies that a comparison with the state-of-the-art methodologies is necessary to determine whether better accuracy can be achieved on limited or augmented image data.

6.2 HARDWARE COMPONENT

Due to the computationally heavy algorithms and models executed during this thesis, the choice of hardware components is crucial. To save time, the hardware differs depending on the experiment part to efficiently run the various tasks in parallel. For converting the raw data to spectrogram images and running the pre-trained models, mentioned in Section 5.4.2 and Section 5.8, a 12:e gen. Intel Core i5-12400F-processor with NVIDIA GeForce RTX 3060 - 12 GB GDDR6X mainly because of these tasks consisting of high computational complexity. The execution time for the spectrogram methods is mentioned in Section 6.3.1, and the execution time for the pre-trained models are presented in Table 5 and Table 6. For training the DCGAN, mentioned in Section 5.7.2, a Macbook Pro with an Apple M1 Pro chip DCGAN is used. The training to produce synthetic images for each class (0-6 classes) took approximately 2 hours. However, the execution time during hyperparameter tuning for DCGAN and the pre-trained CNN models was excluded.

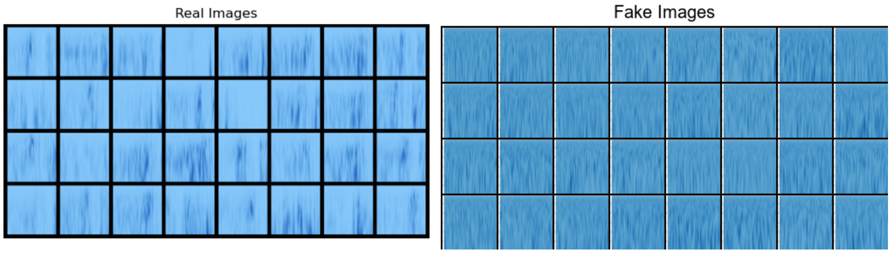
6.3 SPECTROGRAM AND DATA AUGMENTATION

6.3.1 *CWT and STFT Spectrograms*

The steps mentioned in Section 5.4.2 was followed, successfully converting the EMG signal to STFT and CWT spectrogram images. Converting the raw signals to spectrograms is computationally efficient for STFT compared to CWT. The execution of noise augmentation with different noise factors to upsample the dataset does not require significant computational time. The conversion of noise augmented signals to spectrogram images proved to be a time-intensive task in this study. Specifically, the STFT method required approximately 3.5 hours to complete the conversion process. On the other hand, the CWT approach took over 3 days to convert the augmented signals to spectrogram images. This is to achieve the same number of augmented images for training the pre-trained CNN models.

6.3.2 *Traditional- and Intelligent Data Augmentation*

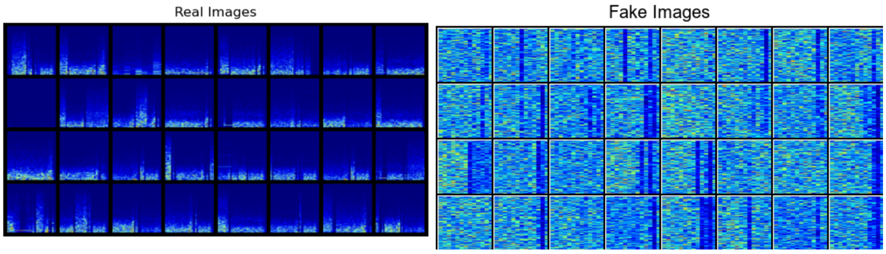
Before comparing the baseline classifiers and the pre-trained models, all the data augmentation methods were implemented to run experiments 2-7 simultaneously. The data augmentation techniques differ in the implementation and generation of images. While the traditional augmentation technique adds noise to the signal, which changes the image's appearance, the DCGAN produces similar images to the real images. In figures 6, 7, 8, and 9 below, a comparison of real and generated images from the different augmentation methods is shown.



(a) Real CWT spectrogram images.

(b) Noise augmented CWT spectrogram images.

Figure 6: Example of real (a) and noise augmented (b) CWT images with a combination of multiple noise factors.

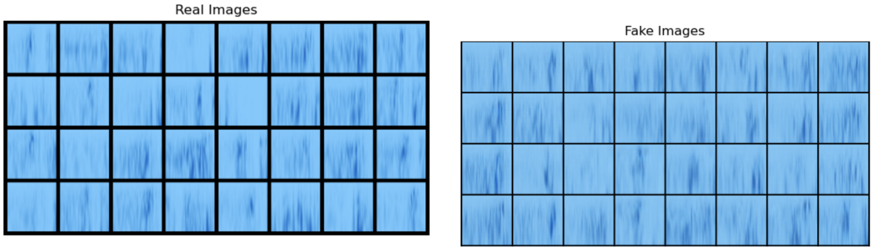


(a) Real STFT spectrogram images.

(b) Noise augmented STFT spectrogram images.

Figure 7: Example of real (a) and noise augmented (b) STFT images with a combination of multiple noise factors.

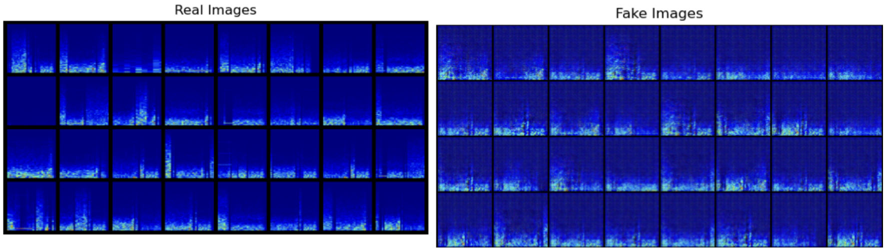
The noise augmentation method, in Figure 6 and Figure 7, adds noise of different levels to the frequency content of the signals, which explains the signal content visualized in various colors in the noise augmented spectrogram images. However, the noise augmented CWT images have more resemblance to the real CWT images compared to the STFT images.



(a) Real CWT spectrogram images.

(b) Fake generated CWT spectrogram images.

Figure 8: Example of real (a) and fake generated (b) CWT images using DC-GAN.



(a) Real STFT spectrogram images.

(b) Fake generated STFT spectrogram images.

Figure 9: Example of real (a) and fake generated (b) STFT images using DC-GAN.

The evaluation of the generated images from DCGAN was done through visualized evaluation. From Figure 8 and Figure 9, the DC-GAN generates similar images as the real images visually. The traditional- and intelligent augmented images are then evaluated by the performance of the various pre-trained CNN models trained on these augmented images. The augmented images from the various methodologies were stored in different directories where each folder contained the augmented images that belonged to a specific label to ensure that the generated images were set to their corresponding label. The various data augmentation methods produced the same number of augmented data, approximately 4900 labeled spectrogram images.

6.4 PRE-TRAINED CNN MODELS

When all the parts of experiments 2-7 were done, the various pre-trained CNN models were initialized for each task. This part of the thesis took the most time because of the number of images to be classified. Various learning rates were tested during this part. However, the models performed best on a learning rate set to 0.001. Table 5, representing the STFT spectrogram images, and Table 6, representing the CWT spectrogram images, the performance and results of the various models with and without augmentation are shown along with running time and accuracy. Each CNN classifier's accuracy reveals whether it can differentiate between the images successfully. The pre-trained CNN models that achieve an accuracy level of over 70 % are more reliable compared, according to Sulaiman and Larsson, if the models only have an accuracy of 50 % or lower [54]. By analyzing the tables, it's clear that the models perform best on CWT spectrogram images compared to the STFT spectrogram images. Some models perform equally or poorly compared to the baseline classifiers in Table 4. However, the accuracy acquired from the baseline classifiers and the various pre-trained CNN models with noise augmentation and without augmentation is unreliable. On the other hand, intelligent data augmentation significantly increases the accuracy of the CNN models and can differentiate greater between the spectrogram images.

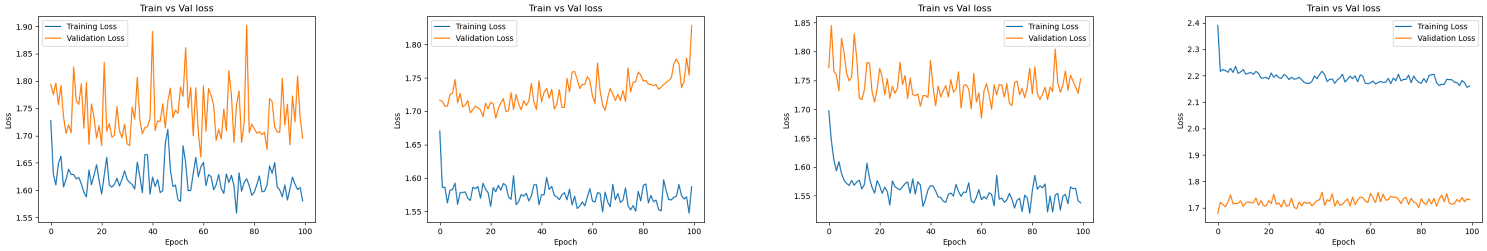
Evaluation of Experiments 2, 4, and 6 for STFT spectrograms				
Pre-Trained model	Data Augmented	Learning Rate	Run Time	Accuracy
AlexNet	-	0.001	9 min	42.7 %
AlexNet	Noise	0.001	33 min	52.7 %
AlexNet	DCGAN	0.001	26 min	71.8 %
VGG16	-	0.001	14 min	42.1 %
VGG16	Noise	0.001	54 min	51.2%
VGG16	DCGAN	0.001	58 min	**81** %
DenseNet	-	0.001	11 min	42.7 %
DenseNet	Noise	0.001	42 min	52.9 %
DenseNet	DCGAN	0.001	44 min	71.9 %
InceptionV3	-	0.001	12 min	41 %
InceptionV3	Noise	0.001	47 min	52.8 %
InceptionV3	DCGAN	0.001	46 min	68.3 %

Table 5: Data augmentation, run time, and accuracy of the different pre-trained models for STFT spectrograms.

Evaluation of Experiments 3, 5, and 7 for CWT spectrograms				
Pre-Trained model	Data Augmented	Learning Rate	Run Time	Accuracy
AlexNet	-	0.001	9 min	43.4 %
AlexNet	Noise	0.001	24 min	50.8 %
AlexNet	DCGAN	0.001	29 min	84.5 %
VGG16	-	0.001	13 min	43.3 %
VGG16	Noise	0.001	43 min	51 %
VGG16	DCGAN	0.001	59 min	**89.9** %
DenseNet	-	0.001	11 min	44.6 %
DenseNet	Noise	0.001	32 min	50.7 %
DenseNet	DCGAN	0.001	43 min	**89.9** %
InceptionV3	-	0.001	12 min	42.7 %
InceptionV3	Noise	0.001	36 min	50.7 %
InceptionV3	DCGAN	0.001	48 min	82.8 %

Table 6: Data augmentation, run time, and accuracy of the different pre-trained models for CWT spectrograms.

In figures 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, and 21 the learning curves and the confusion matrices of the various models with and without data augmentation is shown.



(a) AlexNet (b) VGG16 (c) DenseNet (d) InceptionV3

Figure 10: Training and validation learning curve for STFT without data augmentation.

$$\begin{bmatrix} \mathbf{10} & 3 & 0 & 10 & 0 & 1 & 0 \\ 1 & \mathbf{18} & 0 & 6 & 0 & 0 & 0 \\ 7 & 9 & \mathbf{0} & 12 & 0 & 0 & 0 \\ 2 & 4 & 0 & \mathbf{17} & 0 & 4 & 0 \\ 4 & 3 & 0 & 5 & \mathbf{0} & 8 & 4 \\ 0 & 1 & 0 & 3 & 0 & \mathbf{11} & 3 \\ 0 & 2 & 0 & 2 & 0 & 1 & \mathbf{15} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{15} & 1 & 4 & 4 & 0 & 0 & 0 \\ 4 & \mathbf{9} & 5 & 2 & 5 & 0 & 0 \\ 2 & 0 & \mathbf{25} & 0 & 1 & 0 & 0 \\ 2 & 0 & 22 & \mathbf{0} & 1 & 2 & 0 \\ 0 & 3 & 12 & 2 & \mathbf{4} & 1 & 2 \\ 1 & 0 & 6 & 0 & 0 & \mathbf{10} & 1 \\ 0 & 1 & 4 & 1 & 2 & 5 & \mathbf{7} \end{bmatrix}$$

(b) VGG16

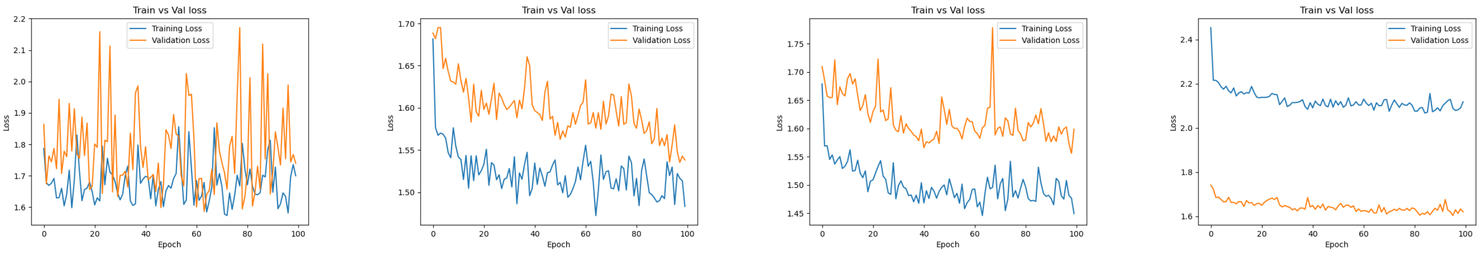
$$\begin{bmatrix} \mathbf{11} & 1 & 6 & 6 & 0 & 0 & 0 \\ 0 & \mathbf{12} & 7 & 5 & 1 & 0 & 0 \\ 1 & 5 & \mathbf{12} & 9 & 1 & 0 & 0 \\ 1 & 6 & 7 & \mathbf{12} & 0 & 0 & 1 \\ 0 & 1 & 5 & 7 & \mathbf{8} & 0 & 3 \\ 0 & 6 & 2 & 1 & 1 & \mathbf{4} & 4 \\ 0 & 3 & 0 & 3 & 2 & 0 & \mathbf{12} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{20} & 2 & 2 & 0 & 0 & 0 & 0 \\ 12 & \mathbf{11} & 0 & 2 & 0 & 0 & 0 \\ 10 & 2 & \mathbf{13} & 0 & 3 & 0 & 0 \\ 11 & 2 & 14 & \mathbf{0} & 0 & 0 & 0 \\ 8 & 1 & 1 & 0 & \mathbf{12} & 2 & 0 \\ 6 & 1 & 2 & 0 & 1 & \mathbf{6} & 2 \\ 4 & 2 & 0 & 0 & 4 & 4 & \mathbf{6} \end{bmatrix}$$

(d) InceptionV3

Figure 11: Confusion matrix for STFT without augmentation.



(a) AlexNet

(b) VGG16

(c) DenseNet

(d) InceptionV3

Figure 12: Training and validation learning curve for CWT without data augmentation.

$$\begin{bmatrix} \mathbf{17} & 5 & 0 & 0 & 2 & 0 & 0 \\ 8 & \mathbf{15} & 0 & 0 & 2 & 0 & 0 \\ 7 & 2 & \mathbf{5} & 3 & 11 & 0 & 0 \\ 8 & 6 & 5 & \mathbf{4} & 4 & 0 & 0 \\ 1 & 3 & 0 & 1 & \mathbf{18} & 1 & 0 \\ 2 & 8 & 0 & 0 & 4 & \mathbf{3} & 0 \\ 2 & 4 & 0 & 0 & 3 & 2 & \mathbf{10} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{10} & 8 & 5 & 1 & 0 & 0 & 0 \\ 7 & \mathbf{14} & 3 & 1 & 0 & 0 & 0 \\ 8 & 5 & \mathbf{11} & 4 & 0 & 0 & 0 \\ 9 & 3 & 3 & \mathbf{10} & 0 & 0 & 2 \\ 2 & 3 & 5 & 2 & \mathbf{11} & 1 & 0 \\ 4 & 1 & 2 & 2 & 0 & \mathbf{7} & 2 \\ 0 & 2 & 0 & 4 & 2 & 3 & \mathbf{9} \end{bmatrix}$$

(b) VGG16

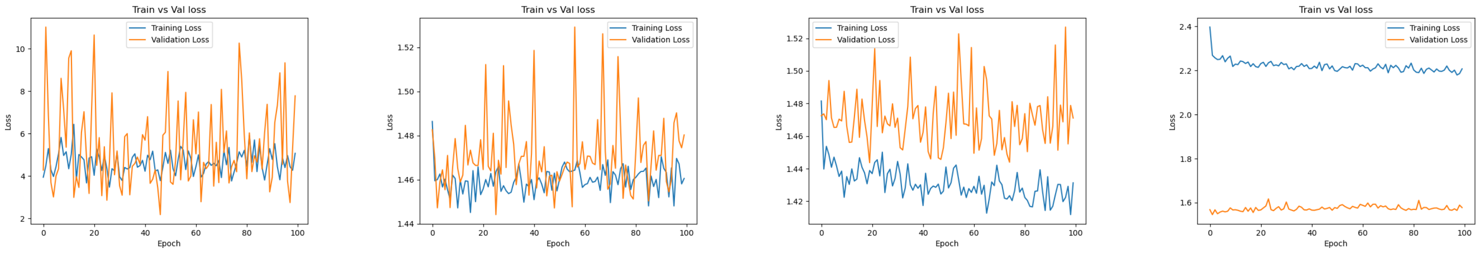
$$\begin{bmatrix} \mathbf{12} & 3 & 2 & 4 & 3 & 0 & 0 \\ 2 & \mathbf{11} & 7 & 3 & 2 & 0 & 0 \\ 3 & 1 & \mathbf{12} & 6 & 6 & 0 & 0 \\ 2 & 3 & 9 & \mathbf{11} & 2 & 0 & 0 \\ 2 & 0 & 2 & 8 & \mathbf{12} & 0 & 0 \\ 0 & 0 & 3 & 4 & 3 & \mathbf{8} & 0 \\ 0 & 0 & 0 & 6 & 3 & 3 & \mathbf{8} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{15} & 1 & 2 & 5 & 1 & 0 & 0 \\ 8 & \mathbf{11} & 2 & 3 & 1 & 0 & 0 \\ 7 & 4 & \mathbf{12} & 3 & 2 & 0 & 0 \\ 6 & 6 & 4 & \mathbf{11} & 0 & 0 & 0 \\ 4 & 4 & 4 & 0 & \mathbf{12} & 0 & 0 \\ 4 & 10 & 2 & 2 & 0 & \mathbf{0} & 0 \\ 2 & 4 & 2 & 2 & 0 & 0 & \mathbf{10} \end{bmatrix}$$

(d) InceptionV3

Figure 13: Confusion matrix for CWT without augmentation.



(a) AlexNet

(b) VGG16

(c) DenseNet

(d) InceptionV3

Figure 14: Training and validation learning curve for STFT with noise augmentation.

$$\begin{bmatrix} \mathbf{118} & 26 & 14 & 0 & 10 & 0 & 8 \\ 10 & \mathbf{82} & 14 & 0 & 32 & 0 & 22 \\ 10 & 14 & \mathbf{77} & 0 & 31 & 0 & 24 \\ 17 & 38 & 16 & \mathbf{0} & 43 & 2 & 46 \\ 4 & 17 & 4 & 0 & \mathbf{110} & 0 & 20 \\ 5 & 16 & 4 & 0 & 21 & \mathbf{86} & 37 \\ 5 & 8 & 1 & 0 & 17 & 0 & \mathbf{125} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{106} & 5 & 27 & 15 & 11 & 5 & 7 \\ 49 & \mathbf{63} & 21 & 10 & 7 & 2 & 8 \\ 38 & 2 & \mathbf{76} & 13 & 6 & 7 & 14 \\ 32 & 12 & 33 & \mathbf{67} & 3 & 4 & 11 \\ 35 & 8 & 14 & 5 & \mathbf{64} & 9 & 20 \\ 30 & 5 & 11 & 1 & 3 & \mathbf{94} & 25 \\ 24 & 4 & 8 & 1 & 0 & 8 & \mathbf{111} \end{bmatrix}$$

(b) VGG16

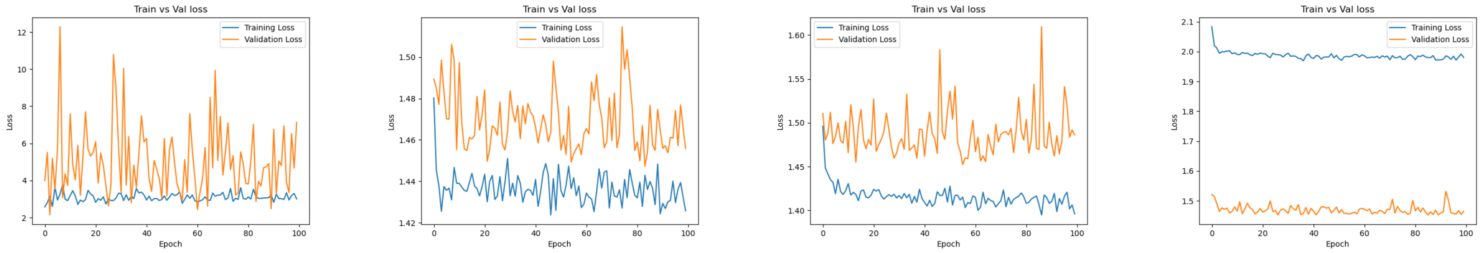
$$\begin{bmatrix} \mathbf{103} & 3 & 29 & 13 & 21 & 4 & 3 \\ 33 & \mathbf{73} & 15 & 21 & 16 & 1 & 1 \\ 34 & 11 & \mathbf{79} & 19 & 10 & 3 & 0 \\ 32 & 1 & 23 & \mathbf{80} & 21 & 3 & 2 \\ 36 & 1 & 13 & 15 & \mathbf{89} & 1 & 0 \\ 21 & 0 & 17 & 5 & 18 & \mathbf{87} & 21 \\ 10 & 0 & 9 & 16 & 18 & 13 & \mathbf{90} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{93} & 14 & 22 & 6 & 20 & 17 & 4 \\ 37 & \mathbf{71} & 15 & 9 & 19 & 5 & 4 \\ 26 & 6 & \mathbf{73} & 28 & 9 & 11 & 3 \\ 26 & 4 & 21 & \mathbf{78} & 10 & 22 & 1 \\ 33 & 5 & 11 & 2 & \mathbf{83} & 15 & 6 \\ 18 & 0 & 3 & 5 & 7 & \mathbf{124} & 12 \\ 29 & 0 & 2 & 13 & 7 & 28 & \mathbf{77} \end{bmatrix}$$

(d) InceptionV3

Figure 15: Confusion matrix for STFT with noise augmentation.



(a) AlexNet (b) VGG16 (c) DenseNet (d) InceptionV3

Figure 16: Training and validation learning curve for CWT with noise augmentation.

$$\begin{bmatrix} \mathbf{165} & 0 & 2 & 3 & 1 & 2 & 3 \\ 82 & \mathbf{50} & 2 & 1 & 5 & 2 & 18 \\ 46 & 2 & \mathbf{82} & 1 & 18 & 3 & 4 \\ 76 & 4 & 3 & \mathbf{58} & 4 & 4 & 13 \\ 77 & 0 & 2 & 1 & \mathbf{73} & 0 & 2 \\ 54 & 1 & 0 & 0 & 10 & \mathbf{60} & 44 \\ 62 & 0 & 4 & 0 & 2 & 0 & \mathbf{88} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{74} & 26 & 26 & 7 & 16 & 8 & 19 \\ 10 & \mathbf{78} & 26 & 11 & 5 & 10 & 20 \\ 19 & 15 & \mathbf{61} & 21 & 13 & 5 & 22 \\ 22 & 10 & 22 & \mathbf{68} & 5 & 11 & 24 \\ 17 & 7 & 21 & 35 & \mathbf{59} & 5 & 11 \\ 2 & 7 & 6 & 18 & 8 & \mathbf{119} & 9 \\ 7 & 10 & 6 & 8 & 1 & 5 & \mathbf{119} \end{bmatrix}$$

(b) VGG16

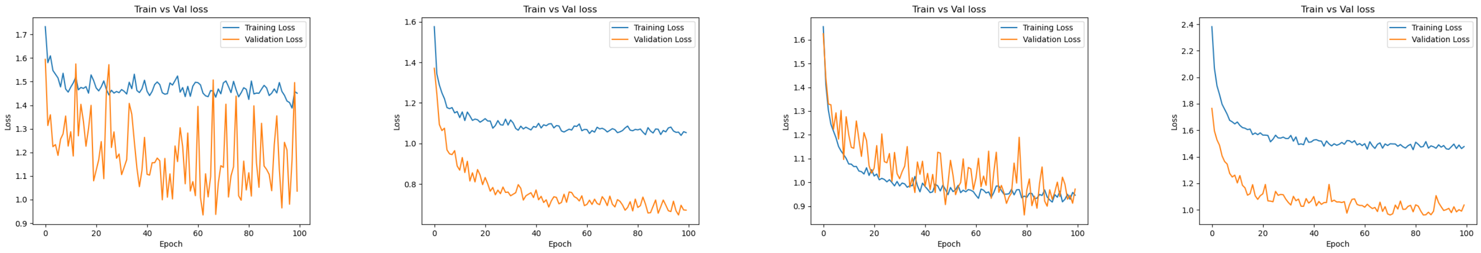
$$\begin{bmatrix} \mathbf{93} & 15 & 1 & 3 & 39 & 13 & 12 \\ 58 & \mathbf{47} & 0 & 6 & 21 & 17 & 11 \\ 10 & 15 & \mathbf{53} & 11 & 43 & 19 & 5 \\ 44 & 11 & 4 & \mathbf{41} & 24 & 26 & 12 \\ 30 & 2 & 8 & 2 & \mathbf{101} & 3 & 9 \\ 24 & 4 & 1 & 1 & 2 & \mathbf{137} & 0 \\ 11 & 6 & 0 & 1 & 12 & 23 & \mathbf{103} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{123} & 18 & 0 & 2 & 18 & 10 & 5 \\ 63 & \mathbf{59} & 0 & 5 & 13 & 16 & 4 \\ 87 & 17 & \mathbf{0} & 8 & 23 & 18 & 3 \\ 32 & 13 & 0 & \mathbf{70} & 24 & 16 & 7 \\ 39 & 7 & 0 & 1 & \mathbf{93} & 12 & 3 \\ 30 & 4 & 0 & 2 & 2 & \mathbf{127} & 4 \\ 29 & 3 & 0 & 4 & 6 & 11 & \mathbf{103} \end{bmatrix}$$

(d) InceptionV3

Figure 17: Confusion matrix for CWT with noise augmentation.



(a) AlexNet (b) VGG16 (c) DenseNet (d) InceptionV3

Figure 18: Training and validation learning curve for STFT with DCGAN augmentation.

$$\begin{bmatrix} \mathbf{75} & 8 & 59 & 6 & 1 & 14 & 13 \\ 35 & \mathbf{66} & 31 & 6 & 0 & 8 & 14 \\ 7 & 0 & \mathbf{146} & 2 & 0 & 1 & 0 \\ 9 & 1 & 18 & \mathbf{127} & 0 & 7 & 0 \\ 2 & 0 & 15 & 0 & \mathbf{130} & 8 & 0 \\ 11 & 0 & 5 & 0 & 2 & \mathbf{151} & 0 \\ 4 & 1 & 13 & 8 & 0 & 10 & \mathbf{120} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{156} & 0 & 13 & 2 & 2 & 3 & 0 \\ 51 & \mathbf{94} & 3 & 0 & 0 & 1 & 11 \\ 21 & 0 & \mathbf{131} & 1 & 2 & 0 & 1 \\ 24 & 0 & 8 & \mathbf{125} & 1 & 2 & 2 \\ 9 & 0 & 3 & 0 & \mathbf{142} & 1 & 0 \\ 7 & 0 & 6 & 0 & 1 & \mathbf{137} & 18 \\ 5 & 0 & 6 & 0 & 8 & 7 & \mathbf{130} \end{bmatrix}$$

(b) VGG16

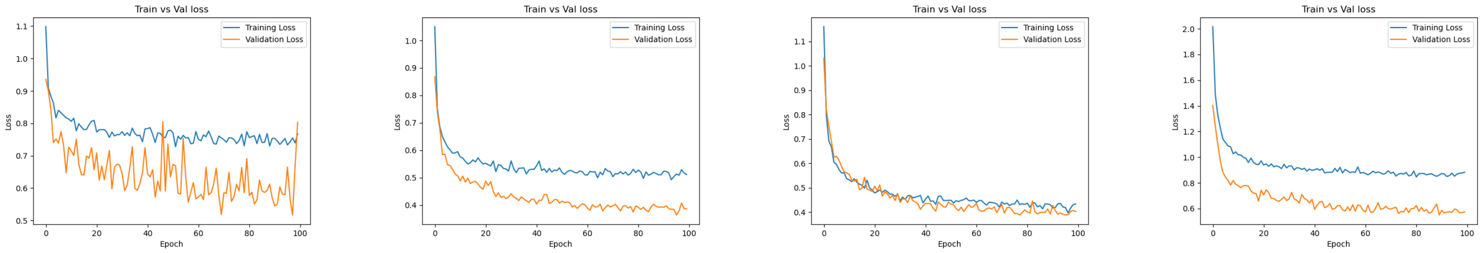
$$\begin{bmatrix} \mathbf{125} & 12 & 17 & 4 & 6 & 12 & 0 \\ 26 & \mathbf{113} & 8 & 4 & 1 & 7 & 1 \\ 22 & 2 & \mathbf{119} & 2 & 8 & 2 & 1 \\ 19 & 6 & 14 & \mathbf{120} & 1 & 2 & 0 \\ 6 & 3 & 4 & 1 & \mathbf{138} & 1 & 2 \\ 27 & 26 & 9 & 0 & 19 & \mathbf{84} & 4 \\ 11 & 3 & 7 & 0 & 10 & 11 & \mathbf{114} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{152} & 1 & 10 & 10 & 1 & 1 & 1 \\ 107 & \mathbf{28} & 4 & 7 & 1 & 0 & 13 \\ 20 & 1 & \mathbf{119} & 6 & 7 & 1 & 2 \\ 22 & 0 & 5 & \mathbf{134} & 1 & 0 & 0 \\ 8 & 0 & 5 & 6 & \mathbf{127} & 6 & 3 \\ 20 & 0 & 13 & 5 & 2 & \mathbf{118} & 11 \\ 10 & 1 & 33 & 2 & 11 & 3 & \mathbf{96} \end{bmatrix}$$

(d) InceptionV3

Figure 19: Confusion matrix for STFT with DCGAN augmentation.



(a) AlexNet (b) VGG16 (c) DenseNet (d) InceptionV3

Figure 20: Training and validation learning curve for CWT with DCGAN augmentation.

$$\begin{bmatrix} \mathbf{172} & 0 & 3 & 1 & 0 & 0 & 0 \\ 24 & \mathbf{134} & 1 & 1 & 0 & 0 & 0 \\ 23 & 0 & \mathbf{121} & 1 & 0 & 0 & 11 \\ 25 & 8 & 2 & \mathbf{127} & 0 & 0 & 0 \\ 9 & 0 & 12 & 1 & \mathbf{119} & 5 & 9 \\ 9 & 0 & 1 & 0 & 2 & \mathbf{155} & 2 \\ 4 & 0 & 3 & 0 & 10 & 9 & \mathbf{130} \end{bmatrix}$$

(a) AlexNet

$$\begin{bmatrix} \mathbf{170} & 0 & 0 & 6 & 0 & 0 & 0 \\ 18 & \mathbf{138} & 0 & 4 & 0 & 0 & 0 \\ 17 & 1 & \mathbf{135} & 1 & 1 & 1 & 0 \\ 18 & 3 & 1 & \mathbf{140} & 0 & 0 & 0 \\ 11 & 0 & 6 & 0 & \mathbf{130} & 2 & 6 \\ 6 & 2 & 0 & 1 & 0 & \mathbf{160} & 0 \\ 3 & 0 & 0 & 1 & 3 & 2 & \mathbf{147} \end{bmatrix}$$

(b) VGG16

$$\begin{bmatrix} \mathbf{162} & 0 & 10 & 3 & 1 & 0 & 0 \\ 12 & \mathbf{139} & 6 & 3 & 0 & 0 & 0 \\ 15 & 1 & \mathbf{137} & 2 & 1 & 0 & 0 \\ 13 & 1 & 5 & \mathbf{141} & 0 & 0 & 0 \\ 7 & 3 & 5 & 1 & \mathbf{135} & 0 & 4 \\ 5 & 0 & 2 & 1 & 3 & \mathbf{155} & 2 \\ 4 & 1 & 0 & 0 & 3 & 0 & \mathbf{148} \end{bmatrix}$$

(c) DenseNet

$$\begin{bmatrix} \mathbf{163} & 2 & 2 & 7 & 2 & 0 & 0 \\ 36 & \mathbf{110} & 3 & 10 & 0 & 0 & 1 \\ 20 & 1 & \mathbf{119} & 4 & 5 & 5 & 2 \\ 28 & 3 & 0 & \mathbf{131} & 0 & 0 & 0 \\ 8 & 0 & 10 & 1 & \mathbf{127} & 4 & 8 \\ 6 & 1 & 0 & 2 & 0 & \mathbf{160} & 0 \\ 4 & 0 & 1 & 2 & 13 & 4 & \mathbf{132} \end{bmatrix}$$

(d) InceptionV3

Figure 21: Confusion matrix for CWT with DCGAN augmentation.

The confusion matrices present an overview of how the pre-trained CNN models perform where true positives, true negatives, false positives, and false negatives are illustrated, showing the types of errors. The learning curve plots show the performance of the various pre-trained CNN models depending on learning rate, data size, spectrogram technique, and data augmentation method.

7 DISCUSSION

The experiments in Figure 3 were successfully implemented and performed during this thesis. The results indicated that intelligent data augmentation using the DCGAN method outperformed traditional data augmentation using noise for the task of classifying the jump height difference. The images generated by the DCGAN during the augmentation process exhibited similarity to real images, leading to better predictions during the fine-tuning and training of pre-trained CNN models. In contrast, the images generated by adding noise to the signals resulted in high variability and less informative representations, affecting prediction accuracy. This observation highlights the importance of generating high-quality augmented data to improve model performance.

The results of the transfer learning part show that training pre-trained CNN models with CWT spectrograms has a better performance compared to using STFT spectrograms. Despite the increased computational time required for the conversion from signals to CWT spectrograms, the improved performance justified this additional computational cost. The CWT spectrograms captured more detailed and informative frequency representations due to their time-frequency localization properties, enabling the CNN models to extract more discriminative features. In contrast, the STFT spectrograms, despite their computational efficiency, provided less precise frequency information, resulting in decreased prediction accuracy. The results highlight the importance of intelligent data augmentation and the use of CWT spectrograms for enhancing model performance in our specific application domain. The combination of intelligent data augmentation using DCGAN-generated images, which closely resemble real images, and the utilization of CWT spectrograms, with their enhanced frequency representation capabilities, leads to improved predictions by the pre-trained CNN models.

The accuracy of the baseline classifier shown in Table 4 proves that the machine learning models Decision Tree, Random Forest, and SVM do not manage to classify the data to the right label and indicate poor performance of the models with the limited EMG dataset and the need for an automated signal processing that represents the time and frequency content of the signal in an image. This is why representing the signal using spectrograms is beneficial for this task. However, the results acquired from the baseline classifiers are similar or better compared to the pre-trained CNN models, despite noise augmentation and architecture. This indicates that the adaption of transfer learning requires more data samples and an augmentation technique that can produce similar images to be beneficial. As shown in Table 5 and Table 6, reliable accuracy was achieved when adapting the intelligent data augmentation technique, DCGAN, which implies that DCGAN augmented images is easier to learn. On the other hand, the CNN models with noise augmented data acquire an accuracy similar to the baseline classifier. Nonetheless, the accuracy is unreliable enough to be considered a well-performing model.

According to our experiments, our research questions could be answered, further discussed in Section 7.7. The pre-trained models do not perform better than the baseline classifier without and with noise augmentation. This is due to the amount of data necessary for CNN models to be beneficial, and the introduction of images with high variability leads to bad accuracy. The only noticeable change in accuracy was achieved from DCGAN. However, the downside of intelligent data augmentation is the vast amount of time necessary to structure and train the models.

7.1 COMPARISON OF THE SPECTROGRAM METHODS

As mentioned, the signals were converted to STFT and CWT spectrogram images. Both methods represent the signal in the time-frequency domain described in Section 4.1.1.3. The frequency represented in the STFT spectrograms is calculated using a linear scale which results in a constant resolution of the frequency, while the CWT uses a logarithmic scale which allows for capturing a better representation of the low and high frequencies. Despite the efficient computation of the STFT method compared to the intensive computation of CWT when converting the EMG signal to spectrogram images, the CWT images have been proven to acquire better results when validating the fine-tuned and trained CNN models.

7.2 LEARNING PLOTS AND CONFUSION MATRICES ANALYSIS

As shown in the various figures and matrices in Section 6.4, the learning plots and confusion matrices were used to analyze how the different models perform and how different test points are predicted to which class. The training learning curve is a measure of the model's learning efficiency derived from the training dataset, and the validation learning curve is used to measure the generalization performance of a model. Analyzing the learning curves of the training loss can reveal an underfit model. If the model is completely unable to learn the training dataset, it may display a flat line or noisy values with a significant loss. The situation described below is typical when the model is unable to adequately handle the complexity of the dataset. These characteristics are presented in the experiments where no data augmentation or noise augmentation method was used, as shown in figures 10, 12, 14, and 16. However, the InceptionV3 model in the mentioned figures and AlexNet with DCGAN augmentation in Figure 18 and Figure 20, the validation dataset is not representative of the classes. Suggesting that there is insufficient data to assess the generalizability of the model. This could happen if there aren't enough instances in the validation dataset compared to the training dataset. A learning curve for training loss that seems to be a good match and a learning curve for validation loss that exhibits noisy movements around the training loss can both be used to identify these characteristics. A validation loss that is less than the training loss may also be used to identify it. In this instance, it suggests that the validation dataset could be simpler to forecast than the training dataset for the model. The models in Figure 18 and Figure 20 where the best accuracy was acquired, the learning plots show a good fit. The learning algorithm aims for a good fit, which may be found between an overfit and an underfit model. A training and validation loss that lowers to a point of stability with a small difference between the two final loss values indicates a good match. Accordingly, there will likely be a discrepancy between the train and validation loss learning curves. This discrepancy is known as the "generalization gap."

The confusion matrices are used to calculate the acquired accuracy for the various models presented in Table 5 and Table 6. It displays the number of accurate and wrong predictions made for each class. It aids in clarifying the classes that models mistake for other classes. The learning curves have a relation with the confusion matrices for each model. The models with the underfitting characteristics, shown in figures 11, 13, 15, and 17, have a confusion matrix which indicates poor performance. In a case of underfitting, the confusion matrix may indicate significant numbers of misclassifications for different classes, reflecting the model's inability to accurately identify instances from any class. However, Figure 19 and Figure 21 consist of the highest true positives and true negatives prediction for each class while having the lowest number of false positives and false negatives, which explains the high accuracy acquired by these models.

7.3 IMPROVEMENT OF ACCURACY USING DATA AUGMENTATION

There are various factors that can affect the huge accuracy improvement of the models. One of the most significant factors was the quality of the images produced by the DCGAN. Noise augmentation only adds random variations to the existing data, but it does not provide new information or different data structures. While DCGANs provide new samples that closely match the original data after learning the underlying data distribution. The model learns more about the semantic features and data structure through this approach. As a result, the model improves its ability to identify crucial patterns and its ability to distinguish between various classes [41]. The images produced from our DCGAN were visually evaluated and by running the pre-trained CNN models on the synthetic images. This proved that our DCGAN had an excellent performance in producing synthetic images that signify the real spectrogram images. Due to the minimal diversity of the samples, the initial dataset used to train the pre-trained CNN models was small, which might cause poor generalization. The effective size of the dataset was greatly expanded by data augmentation. The pre-trained CNN model was given a wider variety of examples in this enhanced dataset, which helped it better understand the underlying patterns and correlations between various classes. Pre-trained CNN models are effective feature extractors. However, the features may not be completely explored if the models are fine-tuned on a limited dataset [41]. While noise augmentation can help a model be more resilient and generalize better, it may not fully utilize the CNN model's feature learning potential. On the other hand, DCGAN-generated synthetic data exposes the models to additional variations and complicated data patterns, allowing it to enhance and expand its gained characteristics. Improved discriminative abilities followed, which improved accuracy on unobserved data [41]. The uncertainty present in DCGAN-generated synthetic data could be utilized as a form of regularisation during training. Regularisation encourages the model to emphasize more generalizable features rather than memorizing the training data, which helps prevent overfitting. The model improved in terms of the ability to generalize to new and unseen data [45].

7.4 COMPARISON OF TRADITIONAL- AND INTELLIGENT DATA AUGMENTATION

The data augmentation methods were intended to produce different types of images. The DCGAN was implemented to produce a fake image that does not exist while resembling the real images, while noise was intended to add noise to the frequency component of the real signal in which the dataset is upsampled. The images produced by the DCGAN, shown in Figure 8 and Figure 9, were visually evaluated and evaluated by implementing the pre-trained CNN models. The noise augmented images, shown in Figure 6 and Figure 7, show that high variability is introduced to the images leading to poor results. Even if the produced images from DCGAN can fool the human eye, the CNN models have been trained on millions of images and manage to extract more information from an image compared to a human. When comparing our result with the literature, it shows that multiple articles achieve a relatively same difference in accuracy when implementing traditional- and intelligent data augmentation [6][39]. Chen and others state that the implementation of Gaussian Noise results in particles of various sizes interfering with noise generated pixels. The variability pixel points alter the distribution of the texture, this lead to a negative result in feature learning, extraction of features, and achieving a reliable accuracy [6]. Therefore, the noise augmentation methodology is not suitable for our task of increasing data samples. This signifies the DCGAN produced images of high quality, significantly enlarged the dataset, and introduced diversity, allowing the CNN models to classify the images with various labels. As mentioned earlier, the DCGAN performs significantly better compared to noise. Thus, intelligent data augmentation techniques should be preferred over traditional data augmentation techniques. However, the implementation of DCGAN is much more complicated and needs two well-built networks, which require hyperparameter tuning until high quality images are generated. A model's hyperparameter tuning and training take significant time, but the results are redeemed and validated.

7.5 COMPARISON OF THE PRE-TRAINED CNN MODELS

The models that should achieve the highest and most reliable accuracy were mentioned in Section 3.4 and Section 4.4.1. The results of this thesis proved that pre-trained DenseNet and VGG16 models were the best performed CNN models when running the different experiments. The deep architecture of DenseNet and VGG16 enables them to identify more complex patterns and representations in the data. These models can learn complex characteristics from the input data due to many layers and parameters, allowing them to capture more details from the input data [18][50]. This has been proven to be beneficial for our tasks which required capturing more details from the spectrograms used.

7.6 DATASET

The data samples and the processing of the signals using the company's MatLab script during this thesis were time-consuming. The script did not allow the processing of all data samples due to most faulty samples being deleted. Hence, using augmentation techniques and transfer learning was crucial during this thesis. When the data samples were gathered, the labeling of the jump exercises was computed by calculating the height difference that allowed a specific target variable. The dataset was limited, creating a challenge in training machine learning algorithms. Acquiring new EMG measurements is a time-consuming process, and the proven results from the intelligent data augmentation DCGAN by generating augmented CWT spectrogram images show that adapting and fine-tuning pre-trained CNN models can solve the problem of the limited data to utilize machine learning techniques to predict muscle fatigue. This combination of state-of-the-art techniques looks promising to be adapted for different tasks.

7.7 ADRESSING THE RESEARCH QUESTIONS

Can state-of-the-art methods, such as transfer learning, STFT and CWT spectrogram techniques, and traditional- and intelligent data augmentation methods, be combined and introduced to the EMG field, if they can, how do they perform compared to each other and to the baseline methods?
The initial assumption was that it would be challenging to introduce state-of-the-art methods to the EMG field due to the lack of data compared to using baseline classifiers, such as Decision Tree, Random Forest, and SVM. This was the case, the pre-trained CNN models did not outperform the baseline classifier without additional assistance to upsample the data. However, neither the baseline classifier nor the CNN models achieved reliable accuracy on the limited dataset.

How do various pre-trained models perform when using traditional- and intelligent data augmentation techniques on time-series data?
The traditional- and intelligent data augmentation techniques affected the CNN models differently. The idea is that the models will perform better with augmented data. However, it was proven that intelligent data augmentation outperforms traditional data augmentation by a margin. On the other hand, the implementation of the data augmentation methods differs. While noise is less computationally heavy compared to the implementation of DCGAN, the DCGAN allows the pre-

trained CNN models to generalize the unseen data well and achieve astonishing results.

How do the pre-trained models perform on different spectrogram methods? As discussed in Section 7.1, spectrogram technology allows the process of raw EMG signals to images. The images contain various types of information, depending on what spectrogram technique is used. The original hypothesis was that the CNN models would perform better when the spectrogram methods STFT and CWT were utilized. This turns out to be the case. The accuracy of the models increased. However, CWT outperformed STFT spectrogram because the method store and contains more information by capturing a high resolution of low and high frequencies.

7.8 FUTURE WORKS

Because the EMG and methodology field is broad, multiple works and methods can be implemented to acquire more results. During this thesis, more experiments could be implemented to compare various state-of-the-art methods. Still, it was not possible due to the constraint imposed by the timeframe. To fully extend our work, the necessary steps in subsections 7.8.1, 7.8.2, 7.8.3, 7.8.4, and 7.8.5 needs to be done.

7.8.1 *The choice of target variable*

For this thesis, the chosen target variable was the difference in jump heights of the subjects when performing the squat jump exercise. This target variable contained specific information about the tired- and newly activated muscles explained in the jump height of each subject. The values were achieved before and after the intensive session and 15 minutes after the muscles started to recover. Interesting research could be done on combining this target variable with another general target variable, such as the Maximum Voluntary Contraction (MVC), which includes more general information about the muscle contractions during the training and indicates the fatigue in the muscle. Combining these two target variables could be an interesting research question to explore if this information could improve the accuracy of the pre-trained CNN models.

7.8.2 *Choice of datasets*

The dataset chosen for this task was limited, allowing us to try various state-of-the-art methods to determine the performance of their introduction within the EMG field. The conducted experiments were successful, and the methodologies were compared. On the other hand, implementing a similar dataset would further confirm our contribution and novelty. Thus, the implementation of another dataset would be necessary to evaluate the achieved result, but this could not be accomplished due to limited hardware and timeframe.

7.8.3 *Hyperparameter tuning*

Due to the limited hardware, the hyperparameter tuning could not be done to a full extent. Hyperparameter tuning is crucial when training deep learning models, meaning that the DCGAN could generate even better images to acquire a better accuracy or that the pre-trained CNN models that did not achieve a reliable accuracy could acquire one. The result of these changes could have beneficial or negative outcomes; thus, more hyperparameter tuning could be necessary for this task.

7.8.4 *Other spectrogram methods*

The spectrogram methodologies were proved to work well with EMG signals were implemented. The implementation of more spectrogram methods could change the outcome of the results, the Mel spectrogram has been shown to perform well with a combination of pre-trained CNN models, but Mel spectrograms are commonly used for voice or sound tasks [34] [46]. However, comparing the result with the already implemented spectrograms within the EMG field would be interesting to see if a potential improvement could be made.

7.8.5 *Transfer learning methodologies*

During this thesis, some of the most common pre-trained CNN models were implemented. However, pre-trained models, such as GoogleNet, ResNet, and SqueezeNet, could have potential in the EMG field. GoogleNet is one of the CNN models that perform well on spectrogram images. An investigation of the mentioned CNN models could be further explored to compare with the CNN models achieving high accuracy during this thesis.

8 CONCLUSION

The goal of surveying and exploring state-of-the-art methods and systematically, theoretically, and practically testing the applicability and performance of more recent machine learning methods on an existing EMG to muscle fatigue pipeline was achieved. A comparison of the baseline methods and state-of-the-art methods was made, and reliable accuracy was achieved by conducting seven different experiments. The suggested future works and implementation are raised in Chapter 7.

The EMG signals were converted to STFT and CWT spectrograms. CWT spectrograms are more computationally heavy than STFT. Still, CWT contains more information by capturing a high resolution of low and high frequencies allowing the pre-trained CNN models from mentioned literature to perform better with CWT images as input. The pre-trained CNN models do not perform well without or with noise augmentation compared to the baseline classifier on limited data. Regardless, none of the models managed to achieve reliable accuracy. However, the implementation and utilization of intelligent data augmentation resulted in an astounding performance for the pre-trained models. DenseNet and VGG16 reached the highest overall accuracy of 89.9 % on augmented CWT images. The well-performed CNN models are qualified to learn features that can generalize unseen data. The DCGAN performed well when generating images to resemble the authentic ones but is more computationally heavy and takes significant time to hyperparameter tune and train.

A APPENDIX

BIBLIOGRAPHY

[1] Know your body excel your game. https://innowearable.se/. Accessed: 2022-11-20.

[2] Mohamed R Al-Mulla, Francisco Sepulveda, and Martin Colley. A review of non-invasive techniques to detect and predict localised muscle fatigue. *Sensors*, 11(4):3545–3594, 2011.

[3] Behnaz Bahmei, Elina Birmingham, and Siamak Arzanpour. Cnn-rnn and data augmentation using deep convolutional generative adversarial network for environmental sound classification. *IEEE Signal Processing Letters*, 29:682–686, 2022.

[4] Mehmet Rahmi Canal. Comparison of wavelet and short time fourier transform methods in the analysis of emg signals. *Journal of medical systems*, 34:91–94, 2010.

[5] Aggelina Chatziagapi, Georgios Paraskevopoulos, Dimitris Sgouropoulos, Georgios Pantazopoulos, Malvina Nikandrou, Theodoros Giannakopoulos, Athanasios Katsamanis, Alexandros Potamianos, and Shrikanth Narayanan. Data augmentation using gans for speech emotion recognition. In *Interspeech*, pages 171–175, 2019.

[6] Ning Chen, Zijin Xu, Zhuo Liu, Yihan Chen, Yinghao Miao, Qiuhan Li, Yue Hou, and Linbing Wang. Data augmentation and intelligent recognition in pavement texture using a deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(12): 25427–25436, 2022.

[7] Rubana H Chowdhury, Mamun BI Reaz, Mohd Alauddin Bin Mohd Ali, Ashrif AA Bakar, Kalaivani Chellappan, and Tae G Chang. Surface electromyography signal processing and classification techniques. *Sensors*, 13(9):12431–12466, 2013.

[8] Christina Christodoulakis, Eric B Munson, Moshe Gabel, Angela Demke Brown, and Renée J Miller. Pytheas: pattern-based table discovery in csv files. *Proceedings of the VLDB Endowment*, 13(12):2075–2089, 2020.

[9] Mario Cifrek, Vladimir Medved, Stanko Tonković, and Saša Ostojić. Surface emg based muscle fatigue evaluation in biomechanics. *Clinical biomechanics*, 24(4):327–340, 2009.

[10] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette,

François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE transactions on neural systems and rehabilitation engineering*, 27(4):760–771, 2019.

[11] Fatih Demir, Varun Bajaj, Melih C Ince, Sachin Taran, and Abdulkadir Şengür. Surface emg signals and deep transfer learning-based physical action classification. *Neural Computing and Applications*, 31:8455–8462, 2019.

[12] Na Duan, Li-Zheng Liu, Xian-Jia Yu, Qingqing Li, and Shih-Ching Yeh. Classification of multichannel surface-electromyography signals based on convolutional neural networks. *Journal of Industrial Information Integration*, 15:201–206, 2019.

[13] Roger M Enoka and Jacques Duchateau. Muscle fatigue: what, why and how it influences muscle function. *The Journal of physiology*, 586(1):11–23, 2008.

[14] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *2016 IEEE international conference on image processing (ICIP)*, pages 3688–3692. Ieee, 2016.

[15] Ebrahim Ghaderpour and Spiros D Pagiatakis. Least-squares wavelet analysis of unequally spaced and non-stationary time series and its applications. *Mathematical Geosciences*, 49(7):819–844, 2017.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[17] Dawei Huang and Badong Chen. Surface emg decoding for hand gestures based on spectrogram and cnn-lstm. In *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, pages 123–126. IEEE, 2019.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[19] Nathan Inkawhich. Dcgan tutorial. `https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html`, . Accessed: 2023-04-20.

[20] Nathan Inkawhich. Finetuning torchvision models. https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html, . Accessed: 2023-05-12.

[21] Fauzani N Jamaluddin, Siti A Ahmad, Samsul Bahari Mohd Noor, Wan Zuha Wan Hassan, and Y Azhar. Features selection for bayes classification of prolonged fatigue on rectus femoris muscle. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2506–2509. IEEE, 2017.

[22] Lorenz Kahl and Ulrich G Hofmann. Comparison of algorithms to quantify muscle fatigue in upper limb muscles based on semg signals. *Medical Engineering & Physics*, 38(11):1260–1269, 2016.

[23] Stefan Karlsson and Björn Gerdle. Mean frequency and signal amplitude of the surface emg of the quadriceps muscles increase with increasing torqueâa study using the continuous wavelet transform. *Journal of electromyography and kinesiology*, 11(2):131–140, 2001.

[24] PA Karthick, Diptasree Maitra Ghosh, and S Ramakrishnan. Surface electromyography based muscle fatigue detection using high-resolution time-frequency methods and machine learning algorithms. *Computer methods and programs in biomedicine*, 154:45–56, 2018.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Sotiris B Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. Data preprocessing for supervised leaning. *International journal of computer science*, 1(2):111–117, 2006.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[28] Thomas A Lampert and Simon EM OKeefe. A survey of spectrogram track detection algorithms. *Applied acoustics*, 71(2):87–100, 2010.

[29] GN Levine and GJ Balady. The benefits and risks of exercise training: the exercise prescription. *Advances in internal medicine*, 38:57–79, 1993.

[30] Gongfa Li, Jiahan Li, Zhaojie Ju, Ying Sun, and Jianyi Kong. A novel feature extraction method for machine learning based on surface electromyography from healthy brain. *Neural Computing and Applications*, 31:9013–9022, 2019.

[31] Tan Jo Lynn and Ahmad Zuri bin Sha'ameri. Comparison between the performance of spectrogram and multi-window spectrogram in digital modulated communication signals. In *2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pages 97–101. IEEE, 2007.

[32] Kiran Marri and Ramakrishnan Swaminathan. Classification of muscle fatigue using surface electromyography signals and multifractals. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 669–674. IEEE, 2015.

[33] Starting Matlab. Matlab. *The MathWorks, Natick, MA*, 2012.

[34] Jash Mehta, Deep Gandhi, Govind Thakur, and Pratik Kanani. Music genre classification using transfer learning on log-based mel spectrogram. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1101–1107. IEEE, 2021.

[35] V Mendez, C Lhoste, and S Micera. Emg data augmentation for grasp classification using generative adversarial networks. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3619–3622. IEEE, 2022.

[36] Kerry R Mills. The basics of electromyography. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl 2):ii32–ii35, 2005.

[37] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak S Turaga. Learning feature engineering for classification. In *Ijcai*, volume 17, pages 2529–2535, 2017.

[38] Hiroyuki Nodera, Yusuke Osaki, Hiroki Yamazaki, Atsuko Mori, Yuishin Izumi, and Ryuji Kaji. Deep learning for waveform identification of resting needle electromyography signals. *Clinical Neurophysiology*, 130(5):617–623, 2019.

[39] J Arun Pandian, G Geetharamani, and B Annette. Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques. In *2019 IEEE 9th international conference on advanced computing (IACC)*, pages 199–204. IEEE, 2019.

[40] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384*, 2018.

[41] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[42] Shivarudhrappa Raghu, Natarajan Sriraam, Yasin Temel, Shyam Vasudeva Rao, and Pieter L Kubben. Eeg based multi-class seizure type classification using convolutional neural network and transfer learning. *Neural Networks*, 124:202–212, 2020.

[43] Mamun Bin Ibne Reaz, M Sazzad Hussain, and Faisal Mohd-Yasin. Techniques of emg signal analysis: detection, processing, classification and applications. *Biological procedures online*, 8:11–35, 2006.

[44] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10), 2020.

[45] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[46] SR Mani Sekhar, Gaurav Kashyap, Akshay Bhansali, Kushan Singh, et al. Dysarthric-speech detection using transfer learning with convolutional neural networks. *ICT Express*, 8(1):61–64, 2022.

[47] Ashish Sharma, Vishal Madaan, and Frederick D Petty. Exercise for mental health. *Primary care companion to the Journal of clinical psychiatry*, 8(2):106, 2006.

[48] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1): 1–48, 2019.

[49] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34, 2021.

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

[51] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[52] Christopher Spiewak, M Islam, A Zaman, Mohammad Habibur Rahman, et al. A comprehensive study on emg feature extraction and classifiers. *Open Access Journal of Biomedical Engineering and Biosciences*, 1(1):1–10, 2018.

[53] Abdulhamit Subasi and M Kemal Kiymik. Muscle fatigue detection in emg using time–frequency methods, ica and neural networks. *Journal of medical systems*, 34(4):777–785, 2010.

[54] Leif Sulaiman and Sebastian Larsson. Genre style transfer: Symbolic genre style transfer utilising gan with additional genre-enforcing discriminators, 2022.

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

[56] Sergio Fuentes del Toro, Silvia Santos-Cuadros, Ester Olmeda, Carolina Álvarez-Caldas, Vicente Díaz, and José Luís San Román. Is the use of a low-cost semg sensor valid to measure muscle fatigue? *Sensors*, 19(14):3204, 2019.

[57] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[58] Panagiotis Tsinganos, Bruno Cornelis, Jan Cornelis, Bart Jansen, and Athanassios Skodras. Data augmentation of surface electromyography for hand gesture recognition. *Sensors*, 20(17):4892, 2020.

[59] David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[60] Jing-jing Wan, Zhen Qin, Peng-yuan Wang, Yang Sun, and Xia Liu. Muscle fatigue: general understanding and treatment. *Experimental & molecular medicine*, 49(10):e384–e384, 2017.

[61] Weiqi Wang. Intelligent scanning detection system of muscle exercise fatigue based on surface electromyography. *Scanning*, 2022, 2022.

[62] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[63] Bilgi Görkem Yazgaç and Mürvet Kırcı. Fractional-order calculus-based data augmentation methods for environmental sound classification with deep learning. *Fractal and Fractional*, 6 (10):555, 2022.

[64] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[65] Hayder A Yousif, Ammar Zakaria, Norasmadi Abdul Rahim, Ahmad Faizal Bin Salleh, Mustafa Mahmood, Khudhur A Alfarhan, Latifah Munirah Kamarudin, Syed Muhammad Mamduh, Ali Majid Hasan, and Moaid K Hussain. Assessment of muscles fatigue based on surface emg signals using machine learning and statistical approaches: a review. In *IOP conference series: materials science and engineering*, volume 705, page 012010. IOP Publishing, 2019.

[66] TNS Tengku Zawawi, Abdul Rahim Abdullah, Ezreen Farina Shair, Isa Halim, and O Rawaida. Electromyography signal analysis using spectrogram. In *2013 IEEE Student Conference on Research and Developement*, pages 319–324. IEEE, 2013.

[67] Kai Zhang, Guanghua Xu, Zezhen Han, Kaiquan Ma, Xiaowei Zheng, Longting Chen, Nan Duan, and Sicong Zhang. Data augmentation for motor imagery signal classification based on a hybrid neural network. *Sensors*, 20(16):4485, 2020.

[68] Yongqing Zhang, Siyu Chen, Wenpeng Cao, Peng Guo, Dongrui Gao, Manqing Wang, Jiliu Zhou, and Ting Wang. Mffnet: Multi-dimensional feature fusion network based on attention mechanism for semg analysis to detect muscle fatigue. *Expert Systems with Applications*, 185:115639, 2021.

[69] Yang Zhou, Chaoyang Chen, Mark Cheng, Yousef Alshahrani, Sreten Franovic, Emily Lau, Guanghua Xu, Guoxin Ni, John M Cavanaugh, Stephanie Muh, et al. Comparison of machine learning methods in semg signal processing for shoulder motion recognition. *Biomedical Signal Processing and Control*, 68:102577, 2021.

[70] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

HALMSTAD
UNIVERSITY

PO Box 823, SE-301 18 Halmstad
Phone: +35 46 16 71 00
E-mail: registrator@hh.se
www.hh.se