BACHELOR

Spiking Neural Networks in Sign Language Recognition

van der Werf, Siebren R.

Award date:
2023

Department of Mathematics and Computer Science
Interconnected Resource-aware Intelligent Systems

Spiking Neural Networks in Sign Language Recognition

Bachelor's Thesis

Siebren Robert-Jan van der Werf

Supervisors:

Prof. Dr. Ir N. Meratnia
MSc. B.R.D. van Berlo

Final Version

Eindhoven, January 2023

Abstract

The recognition of human activity has become increasingly important in recent years, and using WiFi Channel State Information (CSI) data and machine learning techniques to recognize activities performed by humans is promising of research. In this study, we focused on recognizing sign language gestures performed by different humans, which presents a challenge due to the domain shift this dataset then offers. Spiking Neural Networks are a type of neural network that more closely simulate the human brain and uses binary spikes as input data. Through experimentation with different building blocks of a Spiking Neural Network and subsequent analysis, we found that the Spiking Neural Network was able to recognize patterns and classify a number of different sign language gestures. However, the highest accuracy achieved was not consistent with levels reported in previous research utilizing the same sign language gesture dataset. We conducted an optimal hyperparameter search on a smaller subset of the dataset, but when these parameters were applied to a larger dataset they did not yield the same results. Additionally, our results revealed a noteworthy improvement in the network's performance on the MNIST dataset in comparison to the SignFi dataset, suggesting the need for optimization and modification for complex signals. Overall, this research emphasizes the importance of selecting appropriate hyperparameters and the need for further investigation into alternative encoding methods and other building blocks to improve performance.

Preface

This Bachelor's thesis is the result of my Bachelor End Project at the Department of Mathematics and Computer Science of my Bachelor Data Science at the Eindhoven University of Technology. It serves as a demonstration of the skills and knowledge I have acquired throughout my Bachelor's degree in Data Science. The research presented in this thesis is the result of my own independent work, and covers the process of developing and formulating research questions, analyzing the problem at hand, and making a justified choice for methods and techniques.

I would like to thank my supervisors Prof. Dr. ir. N. Meratnia and MSc. B.R.D. van Berlo for the guidance they gave me during the 5 months, but also letting me take my own responsibilities for this thesis and learning the process of how research is conducted.

Siebren van der Werf

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human activity recognition (HAR) is a rapidly growing field that has attracted significant attention in recent years. With the increase of sensors in our environment like the CCTV cameras to the smartphones that everyone has. There is a wealth of data available for tracking and recognizing human behavior. However, the use of this data is limited by privacy laws, as most of this data can be personally identifiable and therefore cannot be used without the proper consent and regulations. To overcome this limitation, researchers have been exploring the use of alternative data sources that can recognize human activity without violating privacy laws.

One such alternative data source is Wi-Fi radio waves. Wi-Fi radio waves are collected through Wi-Fi receivers measuring the disturbance through different wavelengths and frequencies, known as Wi-Fi CSI data. This data can provide valuable information about human activity, such as body movements and actions, without requiring personal identification. Moreover, this data can also be used to recognize more precise movements, like hand gestures, which are difficult to recognize. Wi-Fi CSI data has the potential to be used in a wide range of applications, from smart homes and healthcare to gaming and entertainment.

However, the use of Wi-Fi CSI data for human activity recognition is not without its challenges. The use of Wi-Fi CSI data is limited by the fact that the data collected through the same action can vary depending on external factors such as the position of the action or the size of the person performing the action. This disturbance can be seen as noise that can negatively impact the accuracy of deep learning models trained on this data. This disturbance or noise is one of the main challenges with using Wi-Fi CSI data for human activity recognition and is called the domain shift problem.

One of the key areas of research in the use of Wi-Fi CSI data for human activity recognition is in the development of robust and accurate algorithms for processing this data. Researchers have been exploring various approaches to improve the performance of deep learning models trained on Wi-Fi CSI data, including the use of domain adaptation techniques and transfer learning. Additionally, researchers have been exploring the use of different types of neural networks, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to process Wi-Fi CSI data.

Another approach to addressing the domain shift problem in Wi-Fi CSI data is the use of flexible algorithms that can efficiently process large amounts of information. One promising method is the use of Spiking Neural Networks. Spiking Neural Networks are also called the third generation of neural networks and are a type of artificial neural network that are more like the brain in a human processes all the information it receives. They have a time-dependent feature with spikes carrying the information to the neuron instead of weights. Since the human brain is highly adaptable to changes in domains, it is an interesting hypothesis that spiking neural networks may also have this adaptability.

Overall, the use of Wi-Fi CSI data for human activity recognition is a promising area of research that has the potential to provide valuable insights into human behavior without violating privacy laws. However, the domain shift problem remains a significant challenge that requires further research to overcome. The use of Spiking Neural Networks, domain adaptation techniques, transfer learning, and multimodal learning are some of the promising approaches that can be used to address this challenge.

1.1 Research Questions

1. Can spiking neural networks accurately classify different sign language gestures based on SignFi data?
2. What are the optimal parameters for the spiking neural network when applied to the SignFi datset?
3. How does the performance of the spiking neural network on the MNIST dataset compare to its performance on the SignFi dataset?

1.2 Project Structure

This thesis is structured as follows: in Chapter 2, ”Related Work”, an overview of the most important topics related to this research is presented. This includes literature on the domain shift problem, which motivated the start of this research, literature on WiFi CSI-based recognition, which serves as a comparison for the results and current literature on Spiking Neural Networks. Chapter 3, ”Preliminaries”, provides information on WiFi CSI data and the Doppler Frequency Spectrum, as well as the basic concepts of Spiking Neural Networks and the parameters that can be adjusted. In Chapter 4, ”Methodology”, the research methodology is explained, including the datasets used and the preprocessing steps taken. The different parameters that will be altered in the experiments are also highlighted. The results of the research are presented in Chapter 5, ”Results”, which includes an overview of the results in the form of four tables. Also a follow-up experiment with the optimal parameter combination on a larger dataset is presented. The limitations and potential areas for future work are discussed in Chapter 6, ”Discussion”. Finally, the conclusions of the thesis are presented in Chapter 7, ”Conclusions”.

Chapter 2

Related Work

The domain-shift problem, where models trained on one dataset struggle to generalize to new data from a different domain, has been studied in the medical field for quite some time. One of these studies is the use of images of frontal-view chest radiographs from 32,717 different patients from 4 different databases from hospitals [Petersen et al., 2020]. In this study, the patients were labeled with 8 overlapping findings and a 3d convolutional neural network was used to detect the different cancer classifiers. However, when the dataset was split into train and test sets and the domain-shift of the different hospitals was applied. The average dropping in accuracy of correctly labeling the findings could be as high as 80 percent to 70 percent, which is a drop of 10 percent.

However the problem of domain-shift sensitive data came alight in the research of using WiFi-CSI data for recognition of activities. The field of gesture recognition using wireless signals has seen a number of developments in recent years. In [Li et al., 2016] presented one of the first efforts in this area by detecting small hand movements (finger-grained movements) using wireless router and two antennas. They preprocessed Wi-Fi CSI data by removing outliers, applying a low-pass filter and weighted moving average, and using a discrete wavelet transformation to calculate distances among features. These features were then used in a k-Nearest-Neighbors classifier to classify the 9 finger-grained-movements.

Another study [Albadawy et al., 2018] also shows a clear example of the domain-shift problem in the medical field. In this study images of brain tumors were split depending on the institution they were made. Three CNN's were trained with the training data of one institution but the testing data differed, with one using data from the same institution, one using data from the other institution, and one using data from both institutions. The results showed an average difference of around 5 percent accuracy.

Building on this work,[Ma et al., 2018] expanded the sample size to 276 gestures, each with 20 instances in two different environments. Their system, called SignFi, used signal processing and a 9-layer convolution neural network and achieved an accuracy of 94.81 per cent. This dataset was also used in this thesis.

[Zhou and Wang, 2019] also worked on gesture recognition using wireless signals. They created a dataset of 10 hand gestures performed in different environments, preprocessed the data and transformed it into images using the phase amplitude of the CSI data. They trained a Convolution Neural Network on these images, called DeepNum, which achieved an accuracy of 98 per cent.

[Zheng et al., 2019] created a large dataset of hand gestures that were performed in various locations and orientations and by different people, in order to enable extensive cross-environment testing. This dataset, called Widar, is publicly available and has been used by other researchers as a benchmark dataset. They also developed their own gesture recognition system, called Widar3, which builds on their previous efforts. To create their system, they first extracted the Doppler frequency shift profile from the Wi-Fi CSI data, and then used it to create a body-coordinate velocity profile, which is less dependent on the specific environment. Their system achieved high recognition accuracy, both within the same environment (92.7%) and across different environments (82.6%-92.4%).

In [Brinke and Meratnia, 2019] created their own dataset of participants performing different activities such as clapping, walking, and jumping in a room on different days. They trained a Convolution Neural Network on this data and investigated the possibility of applying transfer learning on raw CSI data. The model achieved a high performance of 90 per cent for a single participant but was sensitive to different participants.

The paper from [Hu et al., 2021] presents WiGR, a Wi-Fi-based gesture recognition system that utilizes a lightweight few-shot learning network to address the domain shift problem. The system can learn a transferable similarity evaluation ability from the training set and apply it to new domains. The experiments on the CSIDA and SignFi datasets show that WiGR can reach high cross-domain accuracy (89.3% - 94.8%) with a reduction in parameters and calculations. The results demonstrate WiGR to be a lightweight and practical gesture recognition system compared to state-of-the-art methods, with excellent recognition performance using only a few samples.

In [Gu et al., 2022] presents WiGRUNT, a device-free and non-contact solution that utilizes the widely available WiFi infrastructure. It is based on an attention-based ResNet backbone that dynamically focuses on the informative clues of a gesture over the spatial and temporal dimension and examines their inherent sequential correlations for cross-domain gesture recognition. WiGRUNT has been evaluated on the open Widar3 dataset and has achieved the highest performance in-domain or cross-domain compared to its state-of-the-art competitors.

The paper [Vreeken et al., 2003] provides an overview of the origins of spiking neural networks and explains the rationale for using a neuromorphic-style artificial neural network. It describes the various types of neurons that make up a spiking neural network and the ways in which the properties of the neurons and synapses can be adjusted to achieve different results. The paper also highlights the advantages and limitations of using this type of network, and the current state of research and development in this field. With the paper of [Auge et al., 2021] conducting a big survey that explores the different options of input encoding with Spiking Neural Networks. This paper summarizes the different schemes presented and provides the theoretical foundations and applicatications of these encoding schemes. It provides a comprehensive understanding of the different ways in which the spiking signals can be encoded and how these encodings can be used in different applications, providing a useful reference for researchers and practitioners working in this field.

Chapter 3

Preliminaries

3.1 Wi-Fi Channel State Information

Wi-Fi CSI data is data that is generated through Wi-Fi signals from transmitter to receiver. The CSI data provides information that is taking into account the effect of signal diffraction, multipath fading and attenuation. For a multi-path channel with N paths, the CSI values for each transmit and receive antenna pair are determined for each sub-carrier is: [Ma et al., 2018]

$$h = \sum_{n}^{N} a_n e^{-\frac{j2\pi d_n}{\lambda} + j\phi_n} \tag{3.1}$$

where a_n is the attenuation coefficient, d_n is the propagation distance from the transmitter to the receiver, ϕ_n is the phase shift along the nth path, and λ is the carrier wavelength. h is a complex number which may be expressed in terms of its magnitude and phase. If there is a change in either the magnitude or phase of at least one of the paths, the CSI value will be altered.

3.1.1 Doppler Frequency Spectrum

The Doppler Frequency Spectrum (DFS) [van Berlo et al., 2022], [Niu et al., 2021] is a measure of frequency shifts caused by the motion of the source or receiver of a wave. It is used in a variety of applications, including radar, remote sensing, and medical imaging. The DFS can reveal the direction and speed of an object relative to the sensor. When applied to WiFi channel state information it can then thus reveal the direction and speed of different objects relative to the WiFi access point or router. This can be used to detect and analyze user behavior, such as whether a user is walking or running, or more precise, which gestures an user is making.

3.2 Domain Shift Problem

The domain shift problem is an issue in machine learning, which means that the distribution of the training data and the test data are different. Which leads to poor performance of a model when applied to unseen data. This problem is well seen when the model is not able to learn enough from the training data to recognize the characteristics of the the different labeled data.

One solution to address this problem is to use machine learning techniques that are flexible and can adapt to highly varying data. These techniques can be robust to changes in the distribution of the data, thus allowing the model to generalize better to unseen data. Examples of these kind of techniques are deep learning methods or even Bayesian Neural Networks [Wen et al., 2019]. These techniques have been shown to be effective in dealing with changes in data distribution and have been successfully applied in various domains.

3.3 Spiking Neural Networks

Spiking Neural Networks are also knows as Neuromorphic Networks and are a type of neural network that operates on binary spiking events. The Spiking Neural Network is mostly designed in such a way that it has a high similarity to the information processing solutions a biological human brain has. This is called a Neuromorphic design. The human brain processes information by sending electric impulses between the neurons. These are called spikes, and are also present in the Spiking Neural Network. The Spiking Neural Network mimics this process by converting the information that is normally handed to a Artificial Neural Network to binary spikes that en hold all the information.

3.3.1 Encoding Methods

Encoding [Auge et al., 2021] is the process of transforming data into a specific format or representation that is suitable for a particular application or system. Encoding for a Spiking Neural Network involves converting analog temporal signals into spike events for input to the Spiking Neural Network. The goal of encoding is to preserve the relevant information content of the signal while compressing the data size, allowing for faster processing and better information preservation. The process of encoding converts analog signals commonly used for Artificial Neural Networks (ANNs) into spike events for input to the Spiking Neural Network. The use of encoding is an essential step in the implementation of Spiking Neural Networks, as it enables the preservation of relevant information from the input signal while also compressing the data size.
There are various forms of encoding techniques, which can be broadly categorized into two main categories: rate-based and temporal encoding schemes.

- **Rate-based encoding tranformation:** This transformation averages the spike activity over time or populations and do not rely on the precise timing of every single spike event. They are robust against fluctuations and noise, and have the advantage of being simple. Examples of rate-based encoding schemes include Poisson Encoding and Population Encoding.
- **Temporal encoding transformation:** This tranformation relies on the precise timing of every single spike and can achieve higher information densities and efficiencies. Examples of temporal encoding schemes include Time-to-first-spike Encoding and Latency Coding.

3.3.2 Neurons

Neurons in a Spiking Neural Network function like the model of the electrical activity of biological neurons. The neurons gets a train of spikes or non-spikes over time and is raised per spike in action potential. When a neuron's potential is reached, it must translate this potential into a binary output, also knows as a spike, to the next neuron.

The mathematical model [Lee et al., 2020] that can be used in the Leaky Integrate-and-Fire (LIF) model. This model describes the evolution of the membrane potential, $V(t)$, over time, t, as the sum of a leak current, I_{leak}, and an input current, $I_{in}(t)$, which is the sum of all the synaptic inputs received by the neuron. The membrane potential is updated over time according to the following differential equation:

$$C\frac{dV(t)}{dt} = -g_{leak}(V(t) - V_{leak}) + I_{in}(t) \tag{3.2}$$

The input current, $I_{in}(t)$, is the sum of all the synaptic inputs received by the neuron and can be modeled as a Poisson process with a time-dependent rate function, $\lambda(t)$. When the membrane potential reaches a threshold value, V_{th}, the neuron generates a spike, and the membrane potential is reset to a value V_{reset}.

$$if \quad V(t) > V_{th} \quad \text{then} \quad V(t) = V_{reset} \tag{3.3}$$

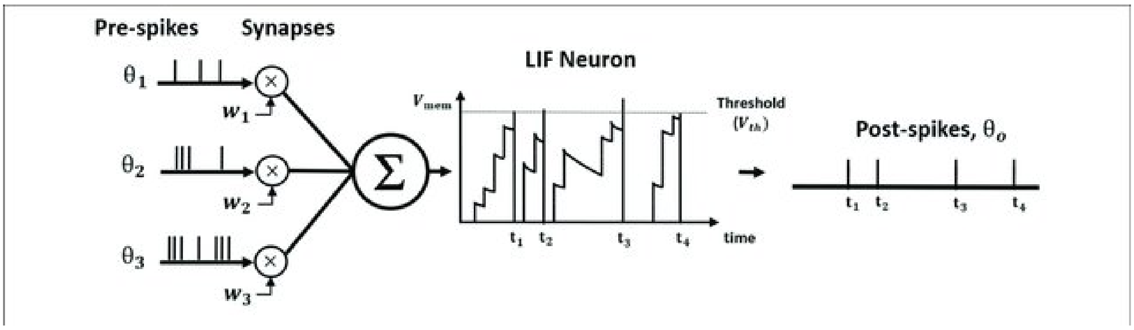


Figure 3.1: The illustration of Leaky Integrate and Fire (LIF) neuron dynamics. The pre-spikes are modulated by the synaptic weight to be integrated as the current influx in the membrane potential that decays exponentially. Whenever the membrane potential crosses the firing threshold, the post-neuron fires a post-spike and resets the membrane potential. *Image and text used from*: [Lee et al., 2020]

3.3.3 Loss function and Optimizer

The primary objective of a neural network is to minimize the loss function by iteratively updating the weights in the network. The loss function serves as a metric to evaluate the performance of the network in correctly classifying or predicting the data. During training of the model there are two separate losses calculated. The loss of the training dataset and the loss of the test dataset. The training loss is utilized for optimizing the model. But both losses are necessary to gain insight into whether the model is experiencing over- or underfitting. Different loss functions are required for various tasks, with the commonly used functions being the cross-entropy loss function for classification tasks and the mean squared error for regression tasks.

The process of updating the weights within the network is facilitated by the optimization function. [Vreeken et al., 2003] The optimizer adjusts the model's parameters during the training process by utilizing the value of the loss function calculated for each training batch. In a Spiking Neural Network, this adjustment is achieved by altering the synaptic weight and threshold of the neuron. Some popular optimization techniques include stochastic gradient descent (SGD) and Adaptive Moment Estimation (Adam).

SGD uses the gradient of the loss function to update the model's parameters and move towards an optimum with a constant step size. On the other hand, Adam uses an adaptive step size known as the learning rate of the model, which controls the rate at which the weights are adjusted per training cycle. This hyperparameter is crucial as it determines the speed at which the model is trained and should be carefully tuned for each problem and dataset.

Chapter 4

Methodology

4.1 Datasets

4.1.1 MNIST

The MNIST dataset [Deng, 2012] is a widely-used dataset for machine learning and computer vision tasks, specifically in the field of image recognition. It consists of 60,000 training images and 10,000 testing images of handwritten digits, each of which is 28x28 pixels in size. The pixels of the images are greyscaled and the images are labeled with the corresponding digit from 0 to 9. The MNIST dataset is often used as a benchmark for comparing the performance of different image recognition algorithms.

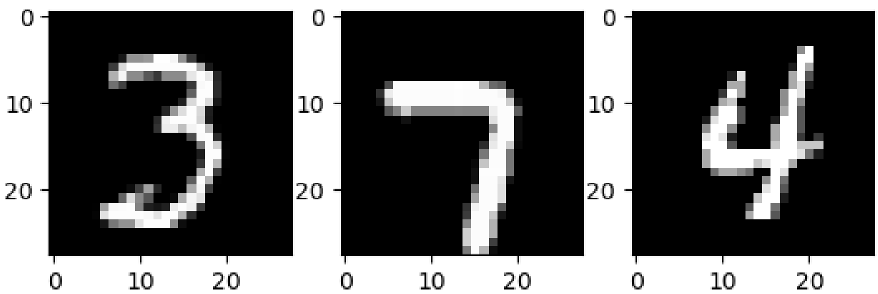


Figure 4.1: Examples of images from the MNIST dataset. The left image represents a handwritten digit '3', the middle image represents a handwritten digit '7', and the right image represents a handwritten digit '4'. The more white the pixel is inside, the higher the value in the pixel. white = 1, black = 0

4.1.2 SignFi

The dataset used in this thesis is derived from the SignFi dataset [Ma et al., 2018] which contains WiFi Channel State Information (CSI) data of various sign language gestures. However, for the purposes of this research, a preprocessed version of the dataset, as described in [Zinys et al., 2021], was utilized. Specifically, the raw WiFi-CSI data was transformed to extract the Doppler shifts of the two different antennas by applying amplitude adjustments, low and high-pass buttersworth filters, and a Short-Time Fourier transform. This resulted in a matrix structure with energy distribution information over both the Doppler frequency and time domains.

The dataset used for the Spiking Neural Network consists of two parts: one part containing 150 sign words that were executed by 5 different individuals, each gesture being executed 10 times, for a total of 7500 inputs. The other part contains 276 gestures performed 30 times by a single individual, with 2/3 of the gestures performed in a lab environment and 1/3 performed in a home environment. This allows for testing the effectiveness of the model in addressing the domain shift problem that may be present in this dataset.

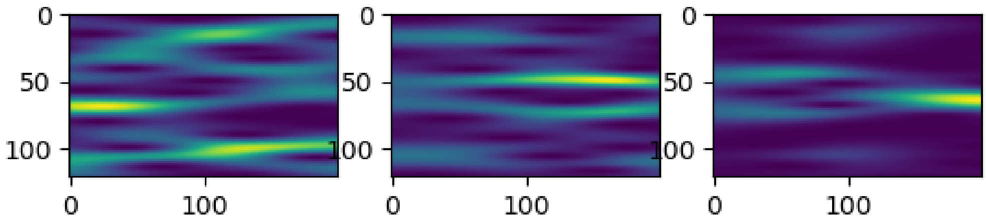


Figure 4.2: Examples of converted images from the SignFi database to DFS images. All the three images represent the word 'go'. The scale is from 0 to 1 and is from dark blue to light green

4.2 Building Blocks

In order to address the research questions, multiple variations of Spiking Neural Networks will be evaluated for their ability to effectively process the challenging WiFi CSI data from the SignFi dataset. The architecture of a Spiking Neural Network can be modified in a variety of ways, and some of the critical building blocks that will be examined are outlined in the table below. These building blocks, including the different input encodings, the number of neurons in the hidden layer, the optimizer used and the learning rate, will be systematically varied in order to explore their impact on the network's performance. The effectiveness of the Spiking Neural Network will be quantitatively assessed through their accuracy in correctly classifying the test data. The assessment will be done by running every variation of combination of the building blocks and running the model for a constant 150 epochs with the datasets presented in 5.1. Furthermore, in this subchapter, the different variations of the building blocks will be explained and justified.

Building Block	**Variation**
Input Encoding	Delta Encoding Rate Encoding
Hidden layer Neurons	Differ between 50, 150, 500 and 1000
Optimizer	Adam SGD
Learning Rate	Differ between 0.1 to 0.00001

Table 4.1: The Building Blocks that are altered in the Spiking Neural Networks

4.2.1 Encoding Methods

Both the SignFi DFS data and the MNIST data can be transformed using various encoding techniques, which are necessary to convert the raw data into a format that is suitable for use as input in a Spiking Neural Network. The encoding techniques that will be used, can be tailored to the specific characteristics of the data to optimize the eventual encoding used as input for the Spiking Neural Network. The encoding schemes that will be used for testing are:

- **delta encoding**: This encoding is the time dependent encoding which executes a spike depending on the difference between the value in the same sub carrier frequency. If the value exceeds a certain threshold, the pixel will be changed to a spike. See figure: 4.3

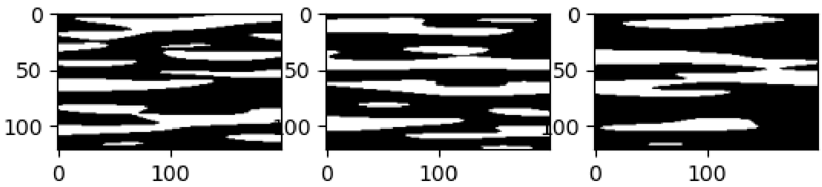


Figure 4.3: The same images converted to spikes using the Delta Encoding of Pytorch. There are only white and black pixels and still white = 1 and black = 0

- **rate encoding**: This encoding is the transformation of the pixels, which have a value between 0 and 1, to a Poisson Probability of a spike event happening. See figure: 4.4

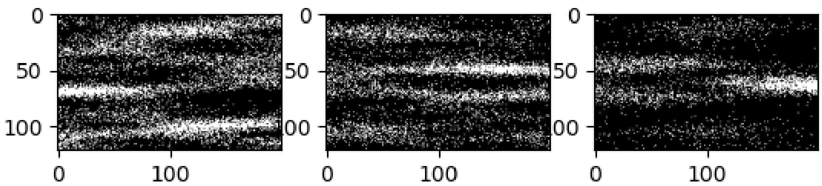


Figure 4.4: The same images converted to spikes using the Rate Encoding of Pytorch. There are only white and black pixels and still white = 1 and black = 0

The Spiking Neural Network will be trained using these encoded data, and the performance of the network can be evaluated based on the chosen encoding method.

4.3 Architecture of the Spiking Neural Network

The architecture of a Spiking Neural Network is crucial to its performance, as it determines the number and organization of neurons within the network. In the case of the SignFi DFS dataset, the preprocessed data is 512x512 pixels in size, and a pooling layer is not applied. Therefore, the input layer of the network must have the same number of neurons as there are pixels in the data. The output layer, on the other hand, will have a number of neurons equal to the number of available classifiers for the dataset's labels. This will result in the MNIST dataset' output layer existing of 10 neurons, one for each digit. And the SignFi DFS data will consist of the number of neurons in the output layer which depends on the specific subset used. This can range from 0 to the maximum of 276.

4.3.1 Hidden Layer

However in the middle layer there is room for flexibility in the number of neurons present. Given the high information density of spikes, the middle layer functions similarly to a convolutional layer in which multiple neurons collectively evaluate an input and fire if a specific pattern is frequently observed in the image. In the experiments this number of neurons will be altered in the search for the effect that this alteration has.

4.4 Optimizers

The choice of optimization algorithms can greatly affect the performance of a model, particularly when working with Spiking Neural Networks. In this study, two optimizers are going to be used to train the Spiking Neural Network: Stochastic Gradient Descent (SGD) and Adaptive Momentum Estimation (Adam). SGD also uses a momentum term which helps the optimizer to overcome the problem of oscillations when the gradients change rapidly. The momentum parameter will be fixed at 0.95 for the experiments. Adam, on the other hand, is an adaptive optimization algorithm that uses the gradient of the loss function to update the model's parameters. It incorporates the concept of adaptive learning rates, which allows the optimizer to adjust the learning rate for each parameter separately based on historical gradient information.

4.4.1 Learning Rate

The learning rate, which determines the step size at which the optimizer makes updates to the model parameters, will also be one of the building blocks. The learning rate for both the SGD and Adam Optimzers will be varied in the range of 1e-6 to 1e-1. When a learning rate increases, the optimizer makes bigger updates to the model parameters, which will lead to faster convergence. But this also has the risk of missing the optimal solution and overshooting and diverging in the wrong direction. However a lower learning rate leads to smaller updates and has a lower risk of diverging in the wrong direction. The influence for all the different learning rates will be assessed based on the accuracy of the spiking neural networks models.

Chapter 5

Results

In this section, we will present the results of the experiments that were conducted using the methodology outlined in Chapter 4. Specifically, we will investigate the effect of altering the various building blocks of the spiking neural network on its performance on the datasets used. The performance of the different models will be evaluated through its accuracy in correctly classifying the test data. Moreover to provide a benchmark comparison, the results of the differentiation of the building blocks will also be tested on the MNIST dataset. Following the examination of all results, the best-performing configurations will be selected for further experimentation.

5.1 Training and Test data split

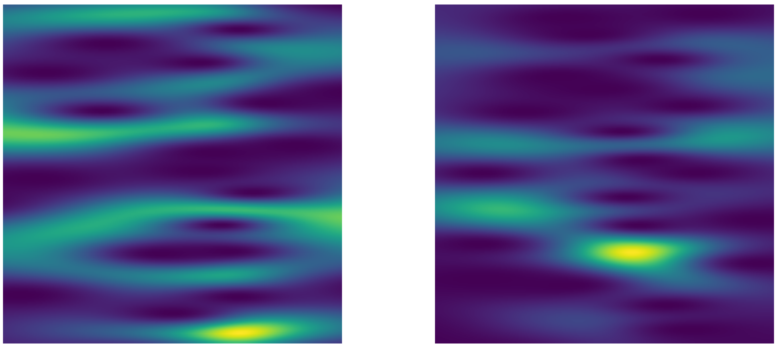
In this subchapter, the various dataset splits that were used in the experiments will be discussed in detail. This will include information on how the data was divided, the rationale behind the chosen split, and any preprocessing that was performed on the data prior and after to the split.

SignFi

In this study, the SignFi dataset, as presented by [Ma et al., 2018], is utilized to train the models. The dataset has been preprocessed to extract the Doppler Frequency Spectrum from raw Wi-Fi CSI data, resulting in a dataset consisting of 7,500 recordings of 121 distinct subcarriers, spanning a duration of 200 milliseconds. This resulted in the input data to have a dimension of: [7500,200,121]. At first the data was divided into different groups based on the users who recorded the gestures performed in the laboratory. This resulted in a remaining of 5520 recordings, 20 of each gesture. Subsequently, the data was further divided into training and test sets to evaluate the performance of the model. The training data was derived from the recordings of four users, which were all from the laboratory dataset. While the test data was derived from the recordings of one user not included in the training set, which was also recorded in the laboratory. This results in a domain-shift for the model to train on.

This domain-shift sensitive dataset was further converted to PNG images of the Doppler Frequency Spectrum per gesture. This dataset has again 5,520 datapoints with images consisting of 512 x 512 pixels and are in RGB. This resulted in a dimension of [3,512,512] for each gesture. Each gesture had 20 Doppler Frequency Spectrum images for training purposes.

For training purposes, a subset of this PNG image dataset was made with the domain-shift sensitivity still intact. This subset consisted of 20 gestures where only 10 of the 20 recordings per gesture were used. Therefor the subset that is used for training consists of 200 recordings of 20 different gestures. The training-test split ratio was 80/20, where 80 per cent of the data was used for training and 20 per cent for testing. This approach allows for a fair evaluation of the performance of the model.



(a) A PNG image of the Doppler Spectrum conversion of the word 'stop'

(b) A PNG image of the Doppler Spectrum conversion of the word 'go'

Figure 5.1: DFS Images

5.2 Experiment

In this subchapter, the results of the experiments conducted to evaluate the effect of altering various building blocks of a Spiking Neural Network on its performance are presented in two different tables. The performance of each Spiking Neural Network was evaluated by measuring its accuracy in correctly classifying the test data from the SignFi dataset. The models were always trained for 150 epochs. Two tables, Table 5.1 and Table 5.2, are provided to summarize the performance of the models.

- **Table 5.1** presents the results of the variations of the building blocks with the constant building block being the Stochastic Gradient Descent optimizer
- **Table 5.2** presents the results of the same variations of building blocks, but with the constant building block being the Adaptive Momentum Estimation optimizer.

Each table provides a comprehensive overview of the effect of altering the building blocks on the performance of the Spiking Neural Networks and allows for easy comparison between the results obtained with different optimizers. The accuracies were calculated by averaging the accuracies of 3 different runs.

Furthermore the tables with the accuracies of the models trained and tested on the MNIST dataset is presented in Table 5.3 and 5.4. The tables have the same format as the previous tables, showing the different variations of the building blocks and the resulting accuracy of the Spiking Neural Networks for each variation. Table 5.3 shows the results with the optimizer being the SGD and Table 5.4 shows the results with the optimizer being the Adam optimizer. These tables provide a benchmark for comparison with the results obtained from the SignFi dataset.

Stochastic Gradient Descent								
	50 Neurons		150 Neurons		500 Neurons		1000 Neurons	
Learning Rate	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding
1e-1	5%	5%	5%	5%	5%	5%	5%	5%
1e-2	7.3%	5%	14.3%	5%	15.6%	10%	14.3%	10%
1e-3	7.7%	6.6%	8.7%	8.3%	11.3%	11.3%	7.5%	12.5%
1e-4	16.6%	10%	20.3%	21.7%	19.6%	12.3%	17.5%	8.3%
1e-5	27.5%	5%	25.3%	18.5%	12.5%	9.3%	27.6%	16.3%
1e-6	20.2%	5%	18.3%	13.6%	10.3%	5.6%	30%	20%

Table 5.1: Accuracy of SignFi subset (20 classifiers) with SGD optimizer

Adaptive Momentum Estimator								
	50 Neurons		150 Neurons		500 Neurons		1000 Neurons	
Learning Rate	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding
1e-1	5%	5%	15.6%	12.3%	26.3%	14.3%	35.6%	10%
1e-2	17.3%	9.6%	27.5%	20%	40%	17.3%	33%	14.6%
1e-3	12.5%	17.5%	15.3%	21.6%	35%	15.5%	33.5%	15%
1e-4	24.6%	15%	15%	20%	36.3%	15.6%	30.3%	12.5%
1e-5	22.5%	5%	30.3	25.3%	37.5%	16.3%	37.5%	25%
1e-6	15.3%	15%	29.6	25.3%	25%	22.9%	30%	35.6%

Table 5.2: Accuracy of SignFi subset (20 classifiers) with Adam optimizer

Stochastic Gradient Descent								
	50 Neurons		150 Neurons		500 Neurons		1000 Neurons	
Learning Rate	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding
1e-1	8%	8%	14%	8%	8%	8%	8%	8%
1e-2	42%	8%	20%	8%	39%	51%	59%	60%
1e-3	45%	49%	42%	49%	45%	49%	48%	53%
1e-4	42%	58%	43%	58%	37%	62%	59%	60%
1e-5	50%	63%	53%	63%	46%	68%	70%	66%
1e-6	65%	57%	62%	57%	54%	69%	71%	73%

Table 5.3: Accuracy of MNIST dataset with SGD optimizer

Adaptive Momentum Estimator								
	50 Neurons		150 Neurons		500 Neurons		1000 Neurons	
Learning Rate	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding	Delta Encoding	Rate Encoding
1e-1	43%	22%	46%	14%	44%	25%	46%	31%
1e-2	63%	72%	66%	78%	74%	74%	73%	77%
1e-3	68%	79%	67%	84%	80%	80%	80%	87%
1e-4	64%	51%	61%	85%	70%	70%	78%	80%
1e-5	54%	62%	75%	69%	82%	82%	82%	88%
1e-6	33%	23%	45%	75%	64%	63%	66%	77%

Table 5.4: Accuracy of MNIST dataset with Adam optimizer

5.2.1 Encodings Methods

In this research, the delta encoding function snntorch.spikegen.delta() and the rate encoding function snntorch.spikegen.rate_conv() were utilized to convert images into a format that can be processed by a Spiking Neural Network. Both these functions are part of the snntorch library which is a PyTorch-based library for training spiking neural networks. The results listed in the Tables 5.1 and 5.2 seem to indicate that the model trained on delta encoded input data demonstrates a higher accuracy compared to the model trained on non-delta encoded input data. The total average for the two different encodings are 13.70% for Rate encoding and 20.64% for Delta encoding. And the best cases for the different encodings can be seen in the following density curves:

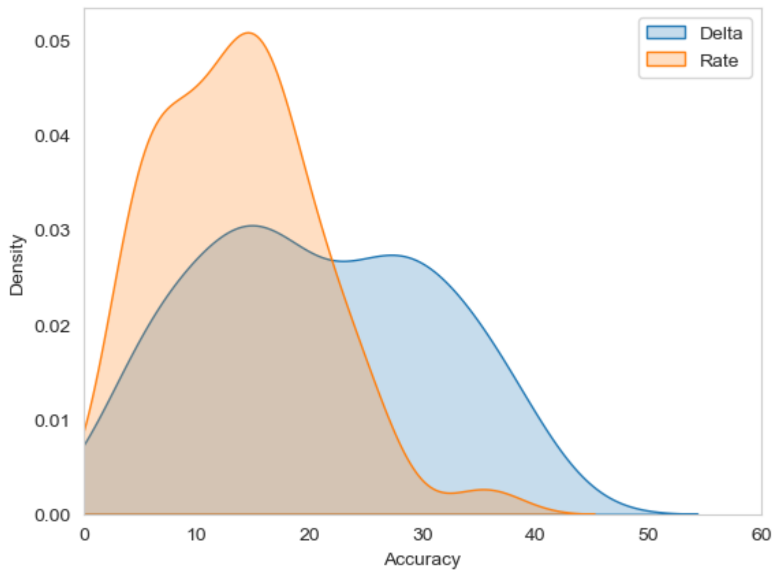


Figure 5.2: A density plot of the accuracies achieved with the different input encodings

5.2.2 Learning Rate

The influence of the learning rate on the accuracy of the Spiking Neural Networks was also evaluated. The results that can be seen in 5.1 and 5.2, show that when using the SGD optimizer, there is a general increase in accuracy with the decrease of the learning rate. However, when using the Adam optimizer, no clear trend is observed in the results for the SignFi dataset. This suggests that the learning rate may have a greater impact on the performance of Spiking Neural Networks when using the SGD optimizer, compared to the Adam optimizer.

Additionally, the same trend is observed in the benchmark test for the MNIST dataset, the increase in accuracy is the same for the SGD optimizer and for the Adam optimizer again no clear trend is observed. This may suggest that the learning rate has a same impact on the MNIST dataset-models as the SignFi dataset-models.

5.2.3 Architecture of the Spiking Neural Network

Furthermore, there also appears to be a correlation between the learning rate and the number of neurons in the hidden layer in relation to the accuracy of the models. As the learning rate of the model decreases, the models tend to perform better with a bigger number of neurons in the hidden layer, and vice versa. This suggests that the optimal number of neurons in the hidden layer and the optimal learning rate may be dependent on each other and may need to be adjusted in conjunction in order to achieve optimal performance of the Spiking Neural Networks.

5.2.4 Optimizers

Based on the results presented in 5.1 and 5.1, there seems to be a clear division in the performance of the optimizers used. The results indicate that the Adam optimizer generally performs better than the SGD optimizer in terms of accuracy. The higher accuracy scores tend to be present on the right side of Table 5.1 and Table 5.2. Additionally, the same trend is observed in the benchmark test for the MNIST dataset, the Adam optimizer generally performs better than the SGD optimizer, this supports the conclusion that Adam is a more suitable optimizer for Spiking Neural Networks in general.

5.3 Following Experiments

Based on the results presented in 5.1 and 5.2 the highest gained accuracies are identified. These highest gained accuracies are further researched by investigating the specific configurations of the building blocks that led to these results. The building blocks that led to the highest accuracy scores, are the ones that are going to be used for testing on a larger subset of the dataset in order to further validate the results and generalize a conclusion. The dataset will be the prior dataset consisting of DFS images of the SignFi dataset, but now the subset will consist of 75 classifiers. This allows us to gain more insight in the ability of the spiking neural network to handle the difficult WiFi CSI data from the SignFi dataset and to create a model that can be used for practical applications. The choice was made to investigate the following building blocks with a larger dataset:

- the Adaptive Momentum Estimation (Adam) optimizer: as it was found to have the highest correlation with accuracy in the experiments
- 500-1000 Neurons: The results suggest that a higher number of neurons in the hidden layer has a positive impact on the accuracy when used in conjunction with the Adam optimizer
- learning rate varying from 1e-1 to 1e-6: The learning rate had no clear effect when using the Adam optimizer.

5.3.1 Following Results

As a result of the experiments, six training runs were conducted, each with a duration of 300 epochs. The performance of the model was evaluated by measuring the accuracy of the test set every 25 epochs. The final accuracy results of the model can be found in Table 5.5. It is worth noting that the only models that demonstrated a high level of accuracy were the Spiking Neural Networks that were trained with a relatively small learning rate.

Learning Rate	**Accuracy**
1e-1	1.33%
1e-2	4.00%
1e-3	2.00%
1e-4	7.33%
1e-5	34.67%
1e-6	24.00%

Table 5.5: Table for accuracy of the Spiking Neural Network (500 neurons in hidden layer, Adam Optimizer, Delta encoding) tested 75 classifiers.

Chapter 6

Discussion

6.1 Limitations

One limitation of this study is the dataset size of the subset, which is significantly smaller than the original dataset tested [Ma et al., 2018]. in 2018. The subset of the dataset used in this study only contained 20 classifiers, which may not have been representative of the full range of gestures in the SignFi dataset. Additionally, the optimal hyperparameters for a smaller dataset may differ from those for a larger dataset, as the added classes may have different recognizable patterns. Which can be seen in the test in which a subset of 75 was used for training with the same hyperparameters.

Another limitation is the limited number of building blocks that were tested in this study. Not all the different building blocks available for a spiking neural network were included in the experiments. The main reason for this is because the number of combinations that could be tested was limited by the time and computational resources available. A single model that ran vor 150 epochs on the bigger dataset of 70 different classifiers took well over 2 hours. However, the use of 150 epochs in the experiments may have been unnecessary. An analysis on the accuracy during training showed that there was little to no improvement in accuracy after the first few 50 epochs, indicating that the training could have been stopped earlier without a significant loss in performance.

6.2 Future Work

The ongoing quest to identify the most effective machine learning algorithm for addressing the domain shift problem requires further investigation. Starting by searching for a efficient algorithm that can correctly classify the domain shift-sensitive dataset SignFi is a good start. The potential utility of third-generation neural networks, known as spiking neural networks, in addressing this issue has yet to be fully determined. Further research is necessary to establish the utility of these networks in solving the domain shift problem. The results of the study revealed what the optimal parameters for a Spiking Neural Network are dependent on the dataset that is trained on. The best accuracy that was achieved was 40.0% (total of 20 gestures) and 34.67% (total of 75 gestures) of the SignFi trained Spiking Neural Network and a 88.0% accuracy for the MNIST trained Spiking Neural Network. Both these optimal accuracies are achieved by a different combination of the building blocks. This indicates that the results highlight the importance of considering the characteristics of the dataset when selecting the parameters for a Spiking Neural Network. Considering that the Spiking Neural Network did achieve an reasonable accuracy, it seems to be able to recognize certain patterns in the data that it can use to classify the complex WiFi CSI data. Therefor a further research for Spiking Neural Networks is still needed, as there are more alterations that can be done like:

- **Different Optimizers and their parameters**: A significant difference in performance was observed between the two optimizers used during the training of the Spiking Neural Network. This highlights the importance of selecting the appropriate optimizer when working with this type of network. This could be further investigated by experimenting with different parameters within the optimizers and even investigating different optimizers to see if they have an impact on the performance of the network.

- **Encoding Methods**: The present study has limited its scope to the investigation of only two encoding methods, namely delta encoding and rate encoding. However, it is important to note that other encoding methods exist and that these encoding methods potentially lead to an improvement in the performance of the Spiking Neural Network and the classification of the SignFi dataset. Further research into alternative encoding methods is necessary in order to fully understand the impact of encoding on the performance of the Spiking Neural Network.

- **Network Architecture**: The experiments only explored a single architecture for the Spiking Neural Network, namely a 3 layered network with the only alteration being the number of neurons in the hidden layer. The architecture was as simple as possible to limit the scope of the alterations within the building blocks. Further investigation into alternative network architectures could indicate improved performance for the Spiking Neural Network. Some examples of these alternative network architectures are Convolutional Spiking Neural Network and transfer learning like in[Brinke and Meratnia, 2019].

In conclusion, the results of this research provide valuable insights into the potential utility of Spiking Neural Networks in addressing the domain shift problem. However, it is clear that further research is necessary to fully establish the utility of these networks in solving this issue. The above mentioned areas of future research have the potential to significantly enhance the performance of Spiking Neural Networks. Therefor it is important to consider these areas in future studies.

Chapter 7

Conclusion

To summarize this research and the findings of this thesis, we will answer the research questions 1.1 stated in the introduction. **Can spiking neural networks accurately classify different sign language gestures based on SignFi data?** The findings of this study demonstrate that a Spiking Neural Network, utilizing the various building blocks employed, can classify different sign language gestures using on the SignFi dataset. The Spiking Neural Network is able to recognize different patterns which makes it able to classify a number of different sign language gestures. However the highest accuracy achieved through the experiments is not consistent with the levels of accuracy reported in previous research utilizing the SignFi dataset. When utilizing a subset of only 20 classifiers, the Spiking Neural Network only achieved a 40% accuracy, which corresponds to correctly classifying 8 out of 20 gestures. However when a subset of 75 gestures was used, an accuracy of 34.67% was achieved, which corresponds to 26 gestures.

What are the optimal parameters for the spiking neural network when applied to the SignFi data? In this thesis, an investigation was conducted to determine the optimal set of building blocks for a spiking neural network when applied to a subset of 20 sign language gestures based on the SignFi dataset. The building blocks evaluated included the optimizer of the loss function (Adam or SGD), the number of neurons in the hidden layer (50-150-500-1000), the input encoding method (rate or delta encoding), and the learning rate (1e-6 to 1e-1). The results of the study indicate that, for the subset of 20 classifiers, the Adam optimizer consistently performed better than the SGD optimizer, and that a high number of neurons in the hidden layer, in combination with the Adam optimizer, had a positive impact on the accuracy of the model. Additionally the delta encoding achieved a significant beter performance than the rate encoding on the SignFi dataset. However the model did not achieve the same results when training a Spiking Neural Network using a dataset consisting of 70 classifiers, using the same hyperparameters. The model failed to give any results when trained with a learning rate greater than or equal to 1e-4. In contrast, utilizing a learning rate of 1e-5 and 1e-6 resulted in an accuracy of approximately 35%.

How does the performance of the spiking neural network on the MNIST dataset compare to its performance on the SignFi dataset? The evaluation of the spiking neural network's performance on the MNIST dataset revealed a noteworthy improvement in comparison to its performance on the SignFi dataset. The highest level of accuracy attained on the MNIST dataset was 88.0%, which is substantially superior to the 40% accuracy observed on the SignFi dataset. These findings suggest that, while a basic 3-layered Spiking Neural Network may possess the ability to classify images, it may require optimization and modification to effectively classify complex signals, such as those present in the SignFi dataset.

The research in summary, highlights the significance of selecting appropriate hyperparameters. It demonstrates that even if a particular combination of parameters is effective for one dataset, it may not be the optimal combination for another dataset, even if the model is using the same input data.

Bibliography

[Albadawy et al., 2018] Albadawy, E. A., Saha, A., and Mazurowski, M. A. (2018). Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Medical physics*, 45:1150–1158. 3

[Auge et al., 2021] Auge, D., Hille, J., Mueller, E., and Knoll, A. (2021). A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters*, 53:4693–4710. 4, 6

[Brinke and Meratnia, 2019] Brinke, J. K. and Meratnia, N. (2019). Scaling activity recognition using channel state information through convolutional neural networks and transfer learning. *AIChallengeIoT 2019 - Proceedings of the 2019 International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, pages 56–62. 4, 18

[Deng, 2012] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29:141–142. 8

[Gu et al., 2022] Gu, Y., Zhang, X., Wang, Y., Wang, M., Yan, H., Ji, Y., Liu, Z., Li, J., and Dong, M. (2022). Wigrunt: Wifi-enabled gesture recognition using dual-attention network. *IEEE Transactions on Human-Machine Systems*. 4

[Hu et al., 2021] Hu, P., Tang, C., Yin, K., and Zhang, X. (2021). Wigr: A practical wi-fi-based gesture recognition system with a lightweight few-shot network. *Applied Sciences (Switzerland)*, 11. 4

[Lee et al., 2020] Lee, C., Sarwar, S. S., Panda, P., Srinivasan, G., and Roy, K. (2020). Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, 14. 6, 7

[Li et al., 2016] Li, H., Yang, W., Wang, J., Xu, Y., and Huang, L. (2016). Wifinger: Talk to your smart devices with finger-grained gesture. *UbiComp 2016 - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 250–261. 3

[Ma et al., 2018] Ma, Y., Wang, S., Zhao, H., Jung, W., and Zhou, G. (2018). Signfi: Sign language recognition using wifi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol*, 2:21. 3, 5, 8, 12, 17

[Niu et al., 2021] Niu, K., Zhang, F., Wang, X., Lv, Q., Luo, H., and Zhang, D. (2021). Understanding wifi signal frequency features for position-independent gesture sensing. *EEE Transactions on Mobile Computing*, pages 1–16. 5

[Petersen et al., 2020] Petersen, J., Estépar, R. S. J., and Schmidt-Richberg, A. (2020). *Thoracic Image Analysis*, volume 1. The Medical Image Computing and Computer Assisted Intervention Society. 3

[van Berlo et al., 2022] van Berlo, B., Ozcelebi, T., and Meratnia, N. (2022). Insights on mini-batch alignment for wifi-csi data domain factor independent feature extraction. *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. 5

[Vreeken et al., 2003] Vreeken, J. et al. (2003). *Spiking neural networks, an introduction.* Utrecht University: Information and Computing Sciences. 4, 7

[Wen et al., 2019] Wen, J., Zheng, N., Yuan, J., Gong, Z., and Chen, C. (2019). Bayesian uncertainty matching for unsupervised domain adaptation. *CoRR*, abs/1906.09693. 5

[Zheng et al., 2019] Zheng, Y., Zhang, Y., Qian, K., Zhang, G., Liu, Y., Wu, C., and Yang, Z. (2019). Zero-effort cross-domain gesture recognition with wi-fi. *Applications, and Services (MobiSys '19).* 3

[Zhou and Wang, 2019] Zhou, G. and Wang, S. (2019). Wifi sensing with channel state information: A survey. *ACM Comput. Surv*, 52. 3

[Zinys et al., 2021] Zinys, A., van Berlo, B., and Meratnia, N. (2021). A domain-independent generative adversarial network for activity recognition using wifi csi data. *Sensors*, 21. 8

Appendix A

Apendix

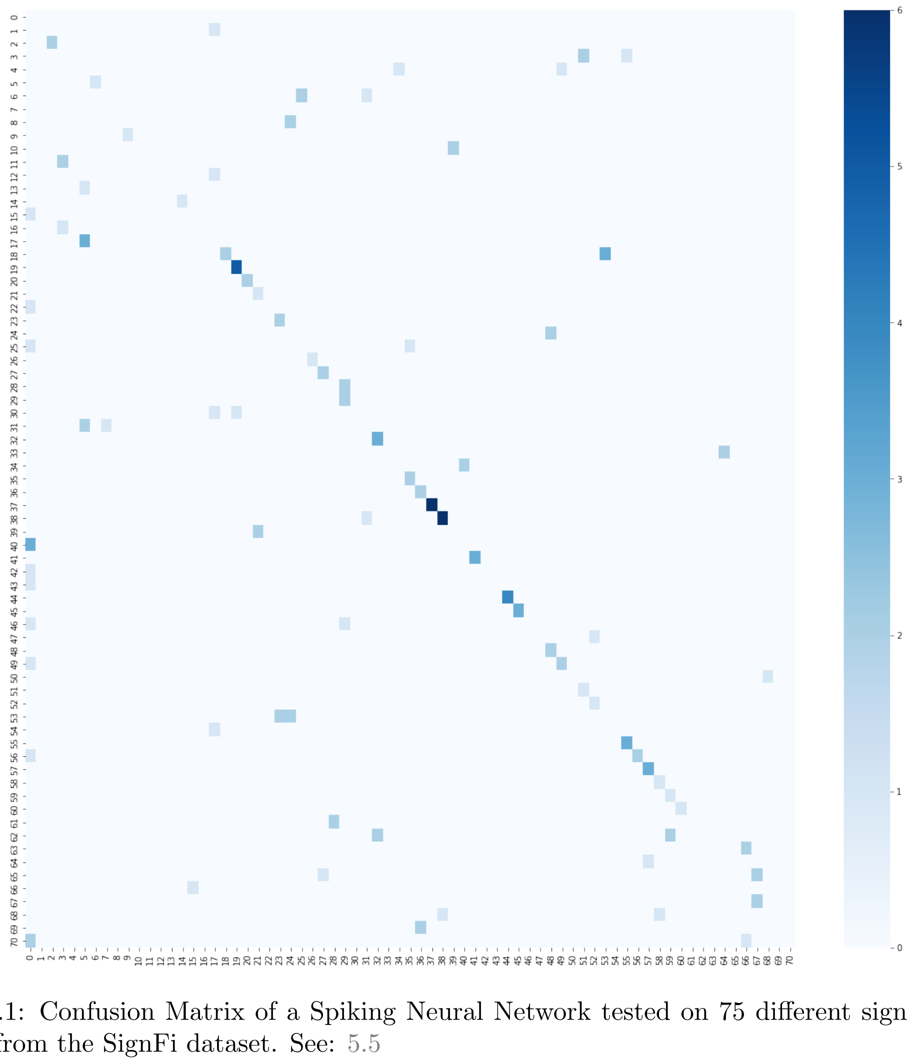


Figure A.1: Confusion Matrix of a Spiking Neural Network tested on 75 different sign language gestures from the SignFi dataset. See: 5.5