TER REPORT

Event-based Lip-reading with Spiking Neural Networks: State of the Art and Detailed Plan

Prepared by
Marcel Schweiker
Hugo Bulzomi

Supervised by
Jean Martinet
Amélie Gruel

Master Informatique
at Université Côte d'Azur
First semester 2022-2023

This document has been written by Marcel Schweiker and Hugo Bulzomi.

Contents

1 Introduction

1.1 Group presentation

This project has been made by Marcel Schweiker (eit Digital) and Hugo Bulzomi (M2 IA-ID). Marcel Schweiker focused on the deep artificial network approach, and tried to apply the current state-of-the-art model to a lip-reading dataset produced at the i3s laboratory. Meanwhile, Hugo Bulzomi explored spiking neural network topologies, and experimented on the DVS-Lip dataset published by Tan et al. (2022). In the end, we present final comparative results on both dataset using both approaches.

1.2 Subject introduction

The goal of this project is to train and compare neural network architectures with the purpose of classifying lip movements. Last year, another research group (Sabatier and Pietrzak (2022)) collected a dataset containing videos captured with an event-based camera of french speakers saying different words. Rather than capturing a whole image for each frame, the camera registers "events": at each time-frame, we only register the pixels that changed at that time. Those cameras are much more energy-efficient than regular cameras, and the videos produced are much lighter. Due to the asynchronous nature of this type of data, event-cameras are a great match for spiking neural network (SNN). These networks use bio-inspired, asynchronous spiking neurons, and are well-adapted to process event-data.

The work will mainly be based on a study published earlier this year by Tan et al. (2022), which achieved state-of-the-art results with an original deep neural network architecture for an automatic lip-reading task : the MSTP network. In the study, a dataset captured with an event-based camera has been used. Due to the asynchronous nature of this data, these events have been preprocessed into fixed frames to use them for the neural network. We would like to compare the results obtained with a network similar to theirs, with the results obtained with a spiking neural network.

2 State of the Art

2.1 Automatic Lip-reading as a Task

Automatic lip-reading is a task where the goal is to transcribe spoken words based solely on visual information. The prospect of systems capable to read-lips is attractive for a variety of real-life application, like assistive devices for persons with disabilities, security and surveillance applications, or automatic transcription of video content. This topic has thus been subject to regular academic publications throughout the years. Compared to other visual tasks, automatic-lip reading deals with especially subtle movements with very precise timing, and is very sensitive to noise and other environmental factors. A model for automatic lip-reading thus needs to excel at extracting both spacial and temporal information. This makes lip-reading a challenging task amongst other visual recognition problems (like face or object recognition), and even humans struggle with it.

The first automatic lip-reading system was made by Petajan (1985) and used image video grayscale thresholding to try to match contours of the face, nostrils and mouth. This approach was very popular until the end the 80s when Yuhas et al. (1989) first applied a statistical

classification model in the form of an early neural network to this problem. 20x25 image frames where used and passed to a feed-forward neural network, and allowed to extract more information than the the previous symbolic method. In later years, hidden Markov chains where applied by Goldschen et al. (1997) and specifically used motions produced by the oral cavities region as features. Hidden Markov chains remained the most popular method throughout the 2000s, but as deep learning started gaining more traction in the mid 2010s, new works using ANNs started being published. In the next section we'll present some of those works, leading to the current state of the art.

2.2 Latest Developments with Deep Learning

2.2.1 Advantages and Adoption

In recent years, deep learning has been applied to automatic lip reading tasks, with some of the earliest examples dating back to the mid-2010s, e.g. Koller et al. (2015) or Petridis and Pantic (2016). The use of deep learning for lip reading has grown in popularity due to its ability to automatically learn and extract features from raw data, such as video frames of a person's face and mouth, without the need for manual feature engineering. This makes it well-suited for tasks like lip reading, where the exact movements of a person's mouth and lips are difficult to model and represent in a traditional, rule-based manner. In general, the adoption of deep learning for automatic lip reading has been driven by the need for more accurate and efficient methods for analyzing and interpreting visual speech, as well as the potential applications of this technology in areas such as speech recognition, sign language translation, and human-computer interaction.

As Chung et al. (2016) discussed, a big challenge for progress in field of Deep Learning for lip reading has been the lack of suitable datasets. But with the availability of large amounts of data and the development of more powerful computing hardware the use of deep learning has been enabled for these tasks, which was not previously feasible.

Existing lip-reading datasets can be divided into several categories depending on the type of recognition object they are designed to capture. Alphabet recognition datasets contain videos of individuals speaking the letters of the alphabet, like the Cox et al. (2008) RMAV or the Matthews et al. (2002) AV Letters Database. Furthermore, there are Digit recognition datasets such as the Anina et al. (2015) OuluVS2, the Messer et al. (1999) XM2VTSDB or the Patterson et al. (2002) CUAVE dataset. The Yang et al. (2018) LRW-1000 dataset as well as the Chung and Zisserman (2017) and Czyzewski et al. (2017) datasets are examples for word recognition datasets. Finally, the last category are sentence recognition datasets such as the Afouras et al. (2018) LRS3-TED, the Cooke et al. (2006) GRID and the Shillingford et al. (2018) LSVSR dataset. All of those datasets were recorded with RGB cameras. The quality and size of these datasets can vary greatly, with some containing thousands of videos and others containing only a few hundred.

All state-of-the-art lip-reading methods are based on deep learning techniques. One approach to lip-reading by Chung and Zisserman (2017) uses a multiple towers architecture, where shallow visual features are first extracted from each frame using 2D convolutions, and then concatenated together. This is followed by the use of several 3D convolution layers, which are used to extract global visual features of the video.

Feng et al. (2020) employ convolutional neural networks (CNNs) as the visual feature extractor, and then use recurrent neural networks (RNNs) to model long-term dependencies and better understand the context and meaning of the words being spoken. Martínez et al. (2020) use a similar approach but with temporal convolutional networks (TCN) instead of RNNs. These RNNs and TCNs are particularly useful for modeling the temporal nature of speech and accurately transcribing words that may span multiple frames of the video.

2.2.2 Event-based data

Most recently, a novel type of sensing devices, event cameras, have been applied to automatic lip reading tasks by Tan et al. (2022). Event-based cameras, also known as asynchronous time-based cameras or event-based vision sensors have several advantages over conventional cameras according to Gallego et al. (2022), including:

1. High temporal resolution: Event-based cameras can capture visual information at a much higher temporal resolution than traditional frame-based cameras with a temporal grain of 1 μs, which can improve the accuracy and robustness of visual tasks such as motion tracking and object recognition.

2. Low power consumption: Event-based cameras consume very little power compared to traditional cameras. They only transmit events when there is a change in the scene, which reduces the amount of data that needs to be transmitted and processed.

3. Asynchronous operation: Unlike traditional frame-based cameras, which capture images at regular intervals, event-based cameras only capture visual information when there is a change in the scene, which can reduce power consumption and improve the dynamic range of the camera.

4. Robustness to motion blur: Event-based cameras are less sensitive to motion blur, which can be a problem for traditional frame-based cameras when capturing fast-moving objects.

Overall, the unique characteristics of event-based cameras makes them especially valuable many computer vision tasks like automatic lip-reading that require high temporal resolution, low power consumption, and robustness to motion blur. Tan et al. (2022) published the first event-based dataset for lip reading tasks, DVS-Lip.

2.2.3 Event-based Lip-reading with MSTP

Tan et al. (2022) also proposed a novel model architecture to perform the event-based lip reading, the Multi-grained Spatio-Temporal Features Perceived Network (MSTP). The performance of the MSTP on this task was experimentally proven to be superior to the state-of-the-art event-based action recognition models and video-based lip-reading models.

One of the most difficult challenge for the lip reading task based on events is an effective event representation for modelling. This is due to the huge amount of events produces by event cameras, since it captures polarity changes for different coordinates every microsecond. As Tan et al. (2022) discuss, the existing event-based action recognition methods such as point-cloud-based, graph-based and fixed-frame-based approaches are not suitable for the lip-reading task, since it requires the perception of finegrained spatio-temporal features from the event data. Furthermore, SNN approaches still lack an efficient back-propagation algorithm. Hence, a voxel grid representation was choosen where each event distributes its polarity to the two closest spatio-temporal voxels.

Within the MSTP (see figure 1), the input events are consequently converted to low-rate and high-rate event frames with different temporal bins to preserve the spatio-temporal information of the event stream better. These two types of event frames are then fed into a multi-branch network with message flow modules between different branches designed to perceive both complete spatial features and fine temporal features from the event data. Followed by that, a sequence model decodes the visual features into words.

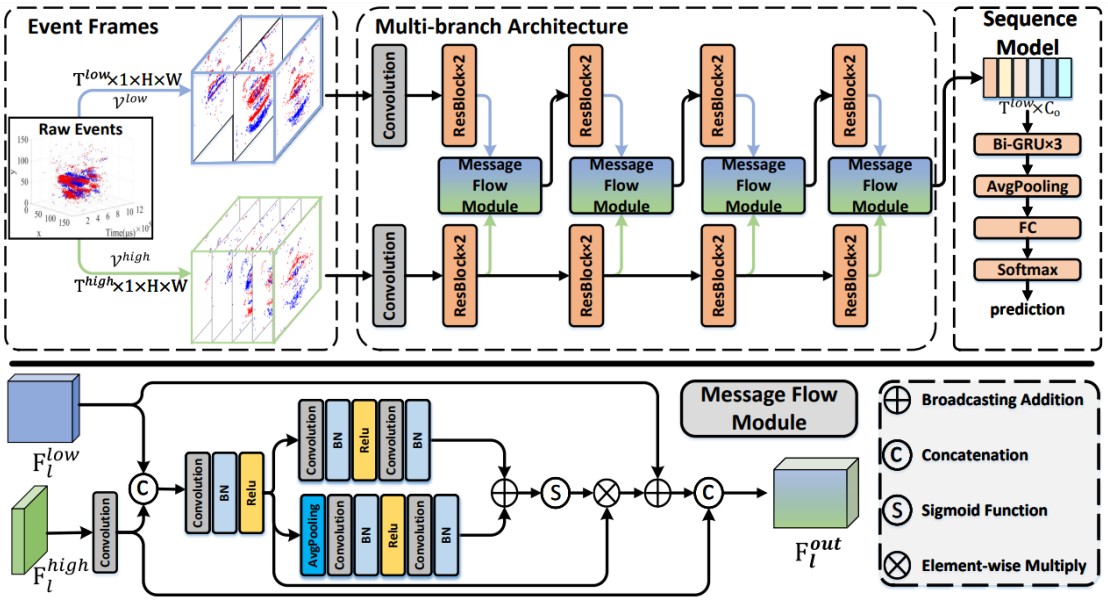


Figure 1: Tan et al. (2022)'s Multi-grained Spatio-Temporal Features Perceived Network

2.3 Spiking Neural Network

2.3.1 Presentation and basic functioning

Spiking neural networks (or SNNs), are bio-inspired learning models that were first popularized in computer science by Maass (1997). In this paper, artificial neurons are divided into three generations:

1. Neurons from the first generation are computational units based on McCulloch and Pitts (1943) neurons and gave birth to the firsts multilayers perceptrons (MLPs), computationally universals for functions with digital inputs and outputs.

2. The second generation builds on top of the previous one by adding "activation functions", such as sigmoid or RELU. This addition allows networks from this generation to be computationally universals for functions with analog input and output. Learning algorithms based on gradient descent like back-propagation are a key feature of neurons of the second generation, and arguably constitute one of the pillars on which modern AI is based.

3. Finally, studies from neurobiology gave the theoretical bases for spiking neurons. Whereas the previous generation greatly simplified biological mechanisms, spiking neurons are meant to function much closer to an actual biological neurons. This leads to models that are potentially much more energy-efficient, and introduces the notion of time during computation (notion totally absent from previous generation, since neurons behaved like mathematical functions). Additionally, SNN are proven to be computationally more powerful than regular neural network models with the same number of neurons.

Where regular artificial neural networks (ANNs) are mathematical functions based on highly simplified brain dynamics, SNNs try to mimic the behaviour of biological neurons by emitting voltage "spikes" with precise timing. Each neuron has a membrane potential (expressed in volts)

that fluctuates based on the neuron's current input. When that potential reaches a certain activation threshold, the neuron fires a "spike" i.e a brief current increase that will travel to all other connected neurons and increase their own membrane potential. The most common spiking neuron model used is the leaky integrate and fire (LIF), which uses a parameter τ to adjust the speed at wich the membrane potential will "leak" towards the resting potential. The figure 1 shows an example of this behaviour.

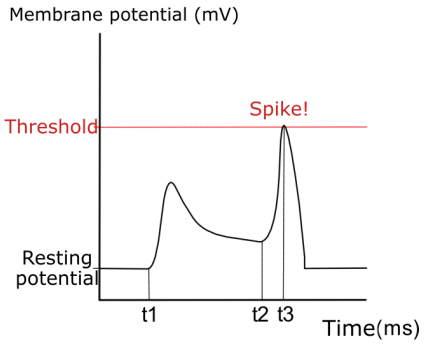


Figure 2: Example of the fluctuation of the membrane potential in a leaky integrate and fire neuron. In the absence of incoming current input, the membrane potential slowly drains to reach a resting value. At t1 and t2, the neuron receives a spike that increases its potential. At t3 the membrane potental reaches the activation threshold value, and the neuron fires a spike before reseting to its resting potential.

Instead of taking an input X, and instantaneously computing an output y based on the model's parameters, SNNs need time for the information to propagate at a certain speed from neuron to neuron: weights and delays are applied between neurons to mimic synaptic conductivity. This causes SNNs to be asynchronous by nature, whereas time has no meaning in regular ANNs. In addition, those neurons are non-differentiable, meaning that they can't natively be seen as derivable functions and thus gradient descent is impossible. Instead, other learning methods had to be employed, like described by Paugam-Moisy and Bohte (2012). STDP for example, is a very popular plasticity mechanism inspired from biology that can easily be applied to SNNs. Also, whereas regular ANNs mainly focus on learning weights between neurons, some studies like Nadafian and Ganjtabesh (2020) have been successful in showing that delays learning is also possible with SNNs. The way one could adjust delays to recognise temporal patterns is shown in figures 2 and 3.

Since gradient descent is impossible with SNN, the training of those networks is notoriously difficult. Until recent years, researchers have mainly been relying either on simple learning mechanisms (like STDP) that are often inefficient for supervised learning, or on the conversion a regular pre-trained ANN into an SNN. More recent works however have been very successful in finding tricks and workarounds to be able to use gradient descent with SNNs. One of the most popular method is described by Shrestha and Orchard (2018): a differentiable surrogate function is used during training instead of the undifferentiable spike function. This surrogate function is an approximation of the spike function, and allows the back-propagation of error through the network. Gradient descent is thus made possible with SNNs. This technique allowed deep spiking networks to equal and sometime surpass regular ANNs in recent works on real-life problems such

as object recognition in automotive data Cordone et al. (2022). Still, this method doesn't utilize the full potential of SNNs since it also brings some of the constraints that regular ANNs have (like the unidirectional flow of information through the network layers).

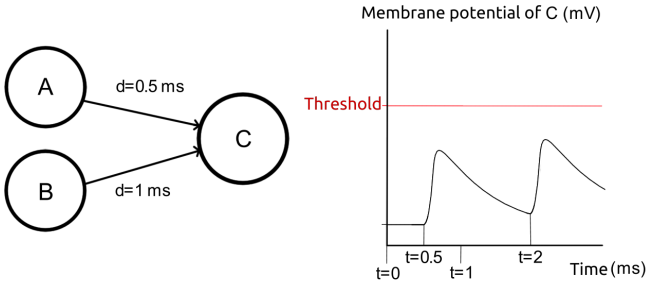


Figure 3: A and B respectively spiking à t=0 and t=1 with synaptic delays $d_{a,c} = 0.5$ and $d_{b,c} = 1$. C doesn't spike.

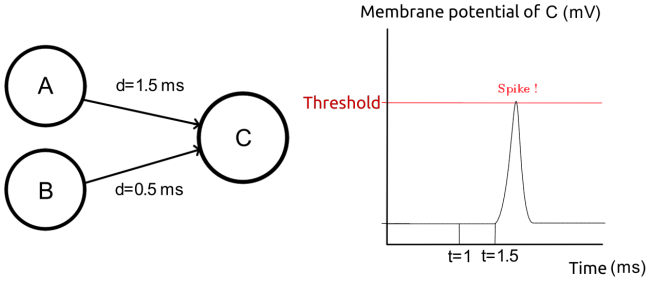


Figure 4: Still with A and B respectively spiking at t=0 and t=1 but with synaptic delays $d_{a,c} = 1.5$ and $d_{b,c} = 0.5$. Both spikes coming from A and B reach C at the same time and causes it to spike.

2.3.2 Uses in the field of computer vision

By the time SNNs became a hot topic of research, regular ANNs had already established solid bases for computer vision. In most cases, studies using SNNs try to take advantage of the techniques previously proven successful.

Just like in computer vision with regular ANNs, convolutions (CNNs) are employed with SNNs. Convolutions allow the model to extract important features (such as edges, corners, and textures) from the input, to reduce its dimensionality for faster processing, and to induce translation invariance: the sliding filters allow to treat all visual inputs the same way no matter their position on the input layer. Cao et al. (2015) first train a deep CNN using regular gradient descent, before converting their network into an SNN that uses the same topology and weights. This approach allows them to take advantage of back-propagation, while still retaining some of the advantages of SNNs.
Though regular ANNs and SNNs share some grounds in computer vision, one key difference between the two is the way the input data is encoded. Regular ANNs can simply take any scalar

vector as input, so a common way to feed RGB images to those networks is to express the image as an array containing 3 scalar values between 0 and 1 for each pixel (representing the 3 color channels). SNNs on the other hand, expect a voltage current as input, either as a spike train for the input layer or a constant voltage input. Still images thus have to be encoded as temporal data to be processed. An example of such encoding would be to convert the RGB image to grayscale and treat each pixel value as a spike probability in the input layer.
Though applicable to image processing, SNNs perform especially well in application domains where time is an important factor because of their asynchronous nature. Visual information from videos is one such type of data where applying regular ANNs can be especially challenging: using each frame as input for the model is computationally very expensive, and frame-rate usually has to be altered to make it usable. In general, some of the temporal information is lost during preprocessing of this type of media for ANNs. Although it's possible to convert a regular video of RGB frames into a format usable with SNNs, those are an especially good fit with the data produced by event cameras.
Event cameras, like presented by Steffen et al. (2019), are a special type of camera using bio-inspired sensors to capture chains of events instead of a sequence of image frames. An event is detected when the relative brightness of a pixel changes past a certain threshold. The camera then simply registers this event as a position (x, y) on the sensor, a polarity (-1 or 1 depending on the brightness change), and a time. This process allows to only track the parts of the image that change, ignoring redundant information, and making these cameras very energy-efficient. Even better, these asynchronous chains of events are a perfect match for SNNs. The perspective of energy-efficient cameras and learning models to go with, made SNNs very attractive for embedded systems like drones, or autonomous cars.
Closer from our own subject, SNNs have been successfully used for gesture recognition in numerous studies like Amir et al. (2017), showing their potential for this type of tasks. Like previously stated, very recent advances have been made in object recognition with event data like in the work of Cordone et al. (2022). Whereas SNNs where only marginally used on very simplistic tasks just a few years back, they are now able to equal the performances of regular ANNs on real-life problems for only a fraction of the computational cost. As of now, SNNs are yet to be applied to automatic lip-reading. Recent studies like Tan et al. (2022) tough, have used data from event cameras coupled with regular ANNs to attain state-of-the-art results for this task. For all the reasons explained earlier, using SNNs with this type of data is very tempting, which motivated our project.

3 Material and Method

3.1 Material

3.1.1 Event Dataset

As mentioned before, event-based recordings provide numerous advantages over regular RGB frames videos. Their high temporal resolution, and robustness to blur make this kind of data an especially promising fit for tasks like lip-reading, where models should learn to process very subtle and fast movements.
At the best of our knowledge, only one publicly available event-dataset for lip reading exists, and has been used in a study: DVS-Lip is a dataset of event-videos produced by Tan et al. (2022), with a DAVIS346 event-camera. It is made of 128x128 mouth-centered videos of 40 volunteers speaking 5 sequences each of 100 different words from an English vocabulary. The final dataset sums to 19,871 individual word recordings of 1.2 seconds (as there was some loss due to file

damage, as explained in their paper), each belonging to one the 100 classes.

Another, not yet published, French event-dataset for lip-reading has been recorded at the i3s lab during the summer 2022: The 2022_i3s_EventLipReadingDataset captures the lower faces of 14 participants speaking 3 sequences of 70 words from a French vocabulary. A GEN3.1 VGA camera was used to record a total of 2.940 videos of 3 seconds with a resolution of 640x480. Each recording lasts for about 3s (that is 3.000.000 us), and generates a very variable number of events between roughly 200 K events and 3 M events. Due to some incidents during recording the dataset, 34 samples were usable, resulting in a total number of 2.906 samples. The i3s dataset, as we will call it from now on for the context of this report, is available on OneDrive.

3.1.2 Code reused and Libraries

We used the Pytorch framework to implement all of our models, along with SpikingJelly for the spiking neural networks.

Regarding the MSTP model, we were able to reuse code published by Tan et al. (2022) together with the corresponding paper and dataset. The code is not publicly available anymore, since the authors have removed the initially published GitHub repository. This might have been due to the poor quality of the code. However, a copy of the code was saved by Lu (2022) within the scope of a summer internship in which he validated their project.

3.2 Our deep learning approach with MSTP

The goal of this part of the project is to adapt the MSTP model presented by Tan et al. (2022) on the i3s dataset. In a first step, the adjustments needed to fit the i3s data to the MSTP code are discussed, before a detailed data analysis is performed on the i3s dataset. Lastly, experimental designs are presented to evaluate the accuracy of the MSTP on the i3s dataset with different pre-processing steps.

3.2.1 MSTP Adaptation

Inspecting the dataset on OneDrive, the i3s dataset's folder structure stands out. The data is sorted first into folders according to their class/spoken word. In each class folder, there are subfolders for every fourteen participants with three recordings each, saved both as .npy as well as .csv files numbered from zero to two. Since the input format of the MSTP is .npy, the first step in adapting the data to the model was to rework the folder structure, removing the subfolders for the participants manually and keeping the .npy files only. A split between test and train was conducted as well, keeping nine samples for testing and 33 for training.

A difference to the DVS-Lip data set used as input to the MSTP by Tan et al. (2022) is the structure of the .npy files. While the DVS-Lip contains the events as an array of Tuple4, x, y, polarity>with defined data type object names ('t', 'x', 'y', 'p') and format ('i8s'), the tuples in the i3s dataset are of the format Tuple4<x, y, polarity, timestamp>without defined data type objects. Hence, the tuple order was restructured and the accesses to column indexes of arrays in the code were changed. Instead of access by data type names (e.g. event_input['t']), the access to values is formulated by indexes (e.g. event_input[:,0]).

Consequently, some hyperparameters were adapted to the data. Here, the parameter 'seq_len' in connection with the parameter 'frame_num' should be highlighted. These parameters are used to control the length of the voxel grid. Tan et al. (2022) explain that if the number of frames

contained in each video recording surpasses thirty, thirty frames are linearly sampled within the MSTP. Otherwise, they are padded to the length of thirty. Since most of the recordings are shorter than 1.20 seconds and all videos have a frame rate of 25FPS, setting the temporal bin of the input event frames of the low-rate branch to thirty results in the same temporal resolution as video-based methods.

Hence, 'seq_len' is set to 30 in the MSTP, while the 'frame_num' is generated from a .json file with the exact number of frames contained in each sample, since the the DVS-Lip recordings differ in length. For the i3s dataset, the situation is different. All the samples are three seconds long and therefore would contain 75 frames in video-based methods. Consequently, we set 'seq_len' to 75 and changed every instance in the fram_num.json to 75 as well. This also means that no linear sampling or padding has to be performed.

In a final step, the centered crop performed by the data loader had to be adjusted to the resolution of the i3s dataset. While the MSTP Model for the DVS-Lip takes in events with the resoolution of 128x128 and crops them initially to a 96x96 pixel excerpt, the i3s datasets resolution is 640x480. Hence, the margins in the functions performing the centered crops have to be adjusted to produce centered crops of the i3s dataset as well.

3.2.2 I3s Dataset Analysis

While the MSTP model can process the data from the i3s dataset with the steps described in the previous chapter, the test accuracy of this current setup remains mediocre. This is why a thorough data analysis step is necessary and will be conducted in the following. The results are based on a small library of self-authored functions tailored for the i3s data set.

To get a clearer picture of the present data, several samples from the i3s dataset have been visualized. In order to do so, it was decided to take 0.5 second frames of the event data and visualize these frames as scatter plots as well as black and white images next to each other. The following figures 5, 6, 7, and 8 show some exemplary samples to highlight some peculiarities of this data set.

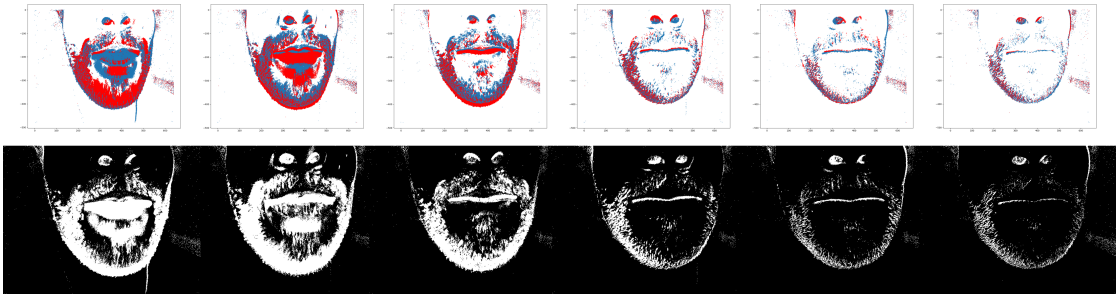


Figure 5: "Addition" Sample 17

For comparison a DVS-Lip depiction is included in Figure 9 as well, showing one of Tan et al. (2022)'s samples. Even though the number of frames remains six for this depiction, the frame duration is only about 0.2 seconds (60% less than the i3s samples) since the DVS-Lip recordings are shorter.

Inspecting these visualizations, it is apparent that i3s dataset is not uniform. First of all, the recordings are not centered around the mouth of the participant. Even though Sample 15's

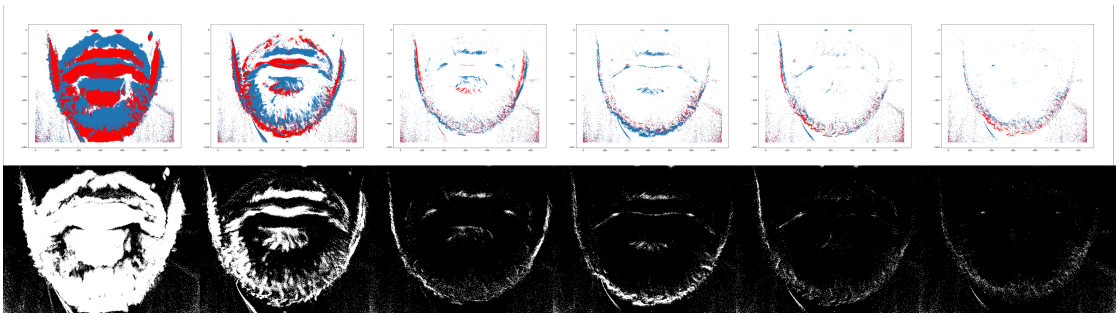


Figure 6: "Addition" Sample 15

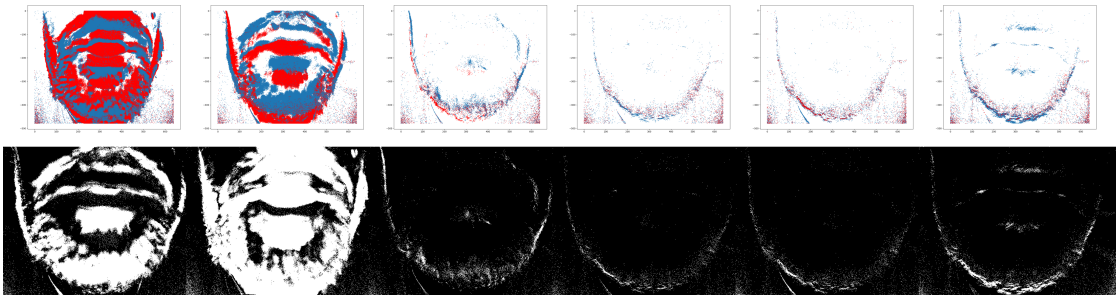


Figure 7: "Joyeux" Sample 15

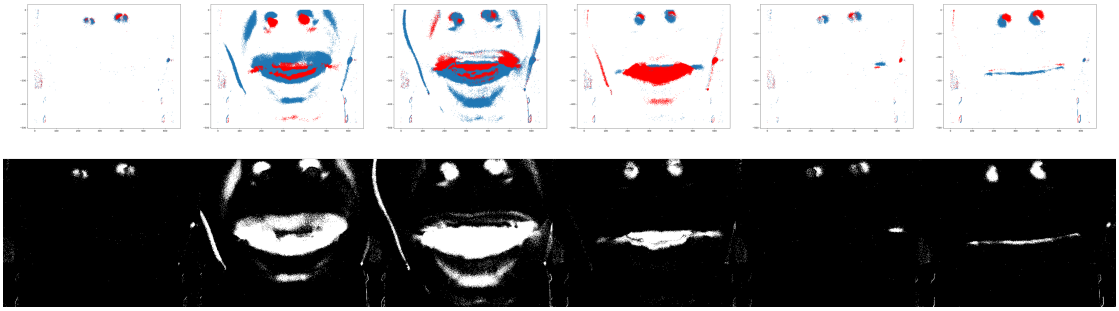


Figure 8: "Carnaval" Sample 9

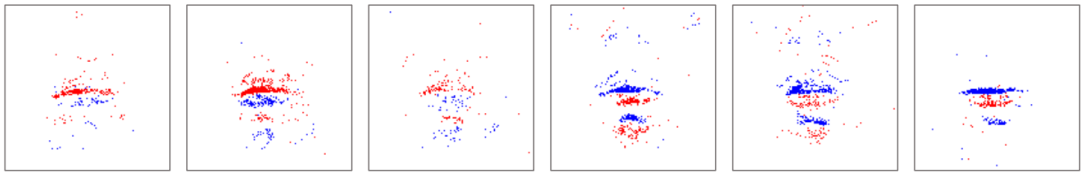


Figure 9: Sample from DVS-Lip

mouth might be centered horizontally, it is shifted upwards. Sample 17 mouth tends to the right, Sample 15 to the bottom of the excerpt.

Furthermore, the partipants maintain different distances to the camera. This results in different image details. Sample 15's mouth occupies a larger area of the excerpt whereas Sample 17 excerpts even shows some details of the lower neck and lower ears. In Sample 9, lots of details of the nose are visible, while on the other hand, the mouth area in those Samples 9 and 17 are significantly smaller in these cases. After all, what does not seem to be a big issue is the movement of the mouth. The area that is occupied by the mouth in the event representation remains the same throughout the recording

Moreover, it is well evident in the visualization that the uniform record duration of three seconds includes unnecessary data. Sample 15 finishes the spoken word within the first second of the recording, resulting in two seconds of almost empty information. Sample 9 begins to speak in the later half of the first second only, finishing the word in around 1.5 seconds.

In contrast to these shortcomings, the DVS-Lip data set consist of mouth centered crops, where the length of the recording corresponds to the actual time needed to articulate the respective word.

Another differentiating aspect between the two data sets is the resolution. The DVS-Lip data set was recorded with an initial resolution of 346x260 pixels, of which a mouth centered cropping to a 128x128 pixels excerpt was performed. In the MSTP the excerpt is cropped further to 96x96 pixels. In training, random 88x88 pixels crops are performed for data augmentation purposes. For testing central crops of 88x88 pixels are performed. This also means that the input size of the network is 88x88 pixels.

The i3s dataset on the other hand is recorded with a resolution of 640x480. The following Figures 10 and 11 depict an examplariy centered crop on these events.

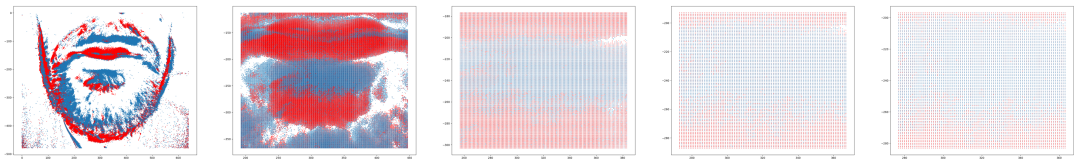


Figure 10: "Addition" Sample 15 Crops

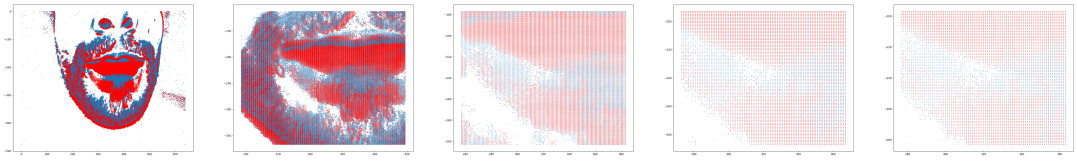


Figure 11: "Addition" Sample 17 Crops

The depictions contain the original image size, as well as centered crops of 256x256, 128x128, 96x96 respectively 88x88 pixels (from left to right). Looking at the 88x88 pixels crops, which is the actual input size of the MSTP, it becomes obvious why the MSTP doe not perform so well on the i3s dataset so far. The 88x88 pixels excerpts are too detailed, showing only a small

area of the lip. Together with the issue of centering and distance to camera, the model might be unable to learn and infer from this data.

3.2.3 Experiment Design

To address these issues, we came up with several ideas to improve the accuracy of the MSTP with the i3s dataset. These will be described in the following.

1. Increasing MSTP Input Size: Since the analysis indicated that the centered crop of 88x88 pixels is too detailed, an obvious idea was to augment the input size of the network. The hypothesis is that the accuracy increases because the model can learn and infer from a more holistic excerpt of the samples.

2. Downscaling Input Event Data: This idea follows the same reasoning as before, but targets the input data instead of the model. The experiments shall show how different resolution reduction factors impact the performance of the model. The hypothesis here is identical to the prior one.

3. Mouth Detection and Downscaling of Event Data: While the previous methods only address the issue of a too detailed excerpt, this method is expected to solve the shortcomings due to non-centered and different-sized mouth areas in the recordings. The idea is to locate the area of the mouth, crop the relevant area and resize the events to 128x128 pixels analogue to the DVS-Lip dataset. This would provide a uniform representation that cancels out the influence of the distance to the camera and the location of the mouth within the frame. Unnecessary details for classification such as necks, ears, noses, glasses etc. will have no effect on the predictions.

3.2.4 Experiment Implementation

Following this experiment design, the steps performed during their execution are described accordingly:

1. Increasing MSTP Input Size: The ability to execute these experiments was very limited. Already small changes of the input sizes turned out to be too computational expensive for the given setup. Increasing the input size more than 10% resulted in memory errors. The set up used was an Ubuntu 20.04.5 LLTS Laptop with an Intel Core i9-10885H CPU (2.4 GHz x 16), 62,5 GB RAM and a NVIDIA TU104GLM[Quadro RTX 5000 Mobile / Max-Q] graphics card with 16 GB of available memory.

2. Downscaling Input Event Data: To downscale the event stream, the event count method introduced in Gruel et al. (2022) has been implemented. The i3s dataset has been preprocessed in a data transformation pipeline (see marceey/i3sEventProcessing) using a function from amygruel/EvVisu, before being fed into the model. Three different downscaling factors have been used, including 2, 3 and 4. This decision was based on the fact that with a downscaling factor of 4, the pixels on the Y axis would exactly correspond to 128 pixels. Hence, reducing the resolution any further would not have any positive impacts on the accuracy of the model. The results are described in chapter 4.1.

3. Mouth Detection and Downscaling of Event Data: In order to detect the mouth area several methods were used to produce clear images of the faces from the event data (see marceey/i3sEventProcessing). Amongst these methods were self-authored black and white image functions as well as functions from the tonic library. Consequently, functions to

detect face landmarks have been applied to these images. First, the referenced face detection tool used by Tan et al. (2022) (ageitgey/face_recognition) to preprocess their dataset has been tried out. But neither this method nor the alternative haarcascade_mcs_mouth classifier from the cv2 library were identifying any faces or mouths in the images. This might be due to the fact that the these method are better suited for full faces and RGB images. In conclusion, the experiments could not have been carried out any further due to the limitations of the project.

3.3 Our approach with Spiking Neural Networks

The previous section presented experimentations with the i3s dataset and the MSTP deep ANN. The following section will now present a new SNN model, along with experimentation based on the DVS-Lip dataset. We will then show comparative results on both DVS-Lip and the i3s datasets in the results section.

SNN have become increasingly popular in the past years, especially as potential energy-efficient models for embedded systems. Though still widely considered as arduous to train properly, recent studies like Cordone et al. (2022) were able to match the performances of regular ANN on image classification/object detection tasks using surrogate gradient descent, while proposing very energy efficient models when deployed on specialised hardware. By approximating the step activation function of spiking neuron with a surrogate differentiable one, we can train our model by descending a surrogate gradient. This method of training showed to be especially successful in the context of supervised training, like in our case. We thus chose to stick with this training method, and used the SpikingJelly (Fang et al. (2020b)) implementation of surrogate functions.

3.3.1 Event data Preprocessing for SNN

As mentioned before, the asynchronous nature of SNN should make them a perfect match for the event data produced by event cameras. In the context of surrogate gradient descent though, it becomes necessary to convert our data to a synchronous form. Finding efficient ways to represent event data is a difficult problem that is, in itself, subject to publications (see Gruel et al. (2022)). Since we are using the DVS-Lip datastet, we chose to use a method similar to the one described by Tan et al. (2022).
In their study, the authors decided to convert the asynchronous events into a 3 dimensional array, i.e a voxel grid. Each event in our dataset is represented as an (x, y) position on the sensor, a time (t), and a polarity {-1, 1}. Equations (1) and (2) (reused from Tan et al. (2022)) show how we can create a voxel grid of a specific length T where the polarity of each event is spread trough the two closest spatio-temporal voxels.

$$t_k^* = \frac{T-1}{t_N - t_1}(t_k - t_1) \tag{1}$$

$$V(t,y,x) = \sum_k p_k max(0, 1 - |t - t_k^*|) \tag{2}$$

With T being the number of frames we want to use (so the time resolution of our grid), and t_x the time of the x^{th} event from the original video, with 1 <x <N. With our event data discretized this way, we get a 3 dimensional grid of shape (t, x, y). In their paper, Tan et al. (2022) use two different values for T: 30 for the low-rate branch of their network, and 210 for

the high rate one. However, the individual performances of each branch when not combined are very similar (accuracy of 69.57% and 69.49% respectively for the low and high branch). Only when both combined in a multi-grained network along with message flow modules blocks do the overhaul accuracy increases to 72.10%. We thus decided to experiment using a T value of 30, as using a higher value would tremendously slow the training processes for little improvement.

3.3.2 Topology exploration

At the best of our knowledge, no published work uses SNNs to classify event-videos for lip-reading. Moreover, the literature for video classification with SNNs is somewhat lacking. The closest studies we could find used the DVS-Gesture dataset introduced by Amir et al. (2017) along with reasonably simple topologies with slight variations, like Yao et al. (2021). The DVS-Gesture dataset itself being comparatively easy to classify, we had to experiment and search for efficient topologies by our own means. This lack of documentation made this part of our work more challenging, but also shows that works like ours can greatly contribute to this domain of study.
We tested several topologies of different levels of complexity. A good starting point was to simply reuse some simple topologies proposed in papers trying to classify DVS-Gesture (we chose Yao et al. (2021)), or for other visual tasks (event-video reconstruction Zhu et al. (2022)). The Figure 12 shows two simple SNN topologies tested, we'll refer to those models as SNN1 and SNN2.

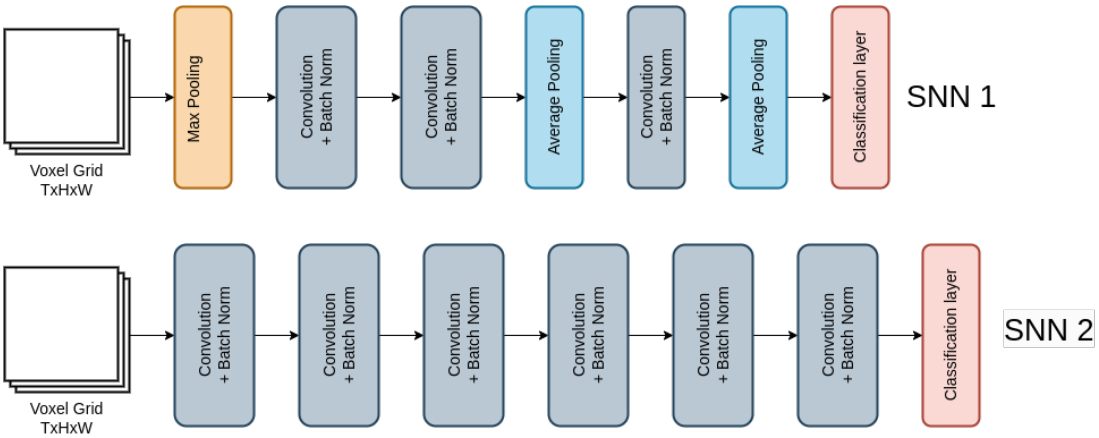


Figure 12: Two basic SNN topologies tested in this work.

SNN1 is taken from Yao et al. (2021), as it was used to classify DVS-Gesture. SNN2 is inspired by Zhu et al. (2022), where the authors use a spiking encoder-decoder architecture for event-video reconstruction. SNN2 is the encoder part of their model, where the residual layers have been replaced by two more convolution layers (this yielded better results during experimentation). Batch normalisation layers have been added after each convolution, as these showed to hugely facilitate the learning process. On this topic, Cordone et al. (2022) showed batch normalisation layers to be vital when using complex SNNs, and reported either significant performance drop, or networks plainly not learning when not using batch normalisation.
Along with these simple models, we decided to make a spiking equivalent of the low-rate brach of MSTP. As this model uses ResNet as backbone (see He et al. (2016)), we had to implement a spiking ResNet first, and then apply the MSTP architecture. Other works have already successfully created a spiking ResNet, like Fang et al. (2021a), which made us optimistic on this

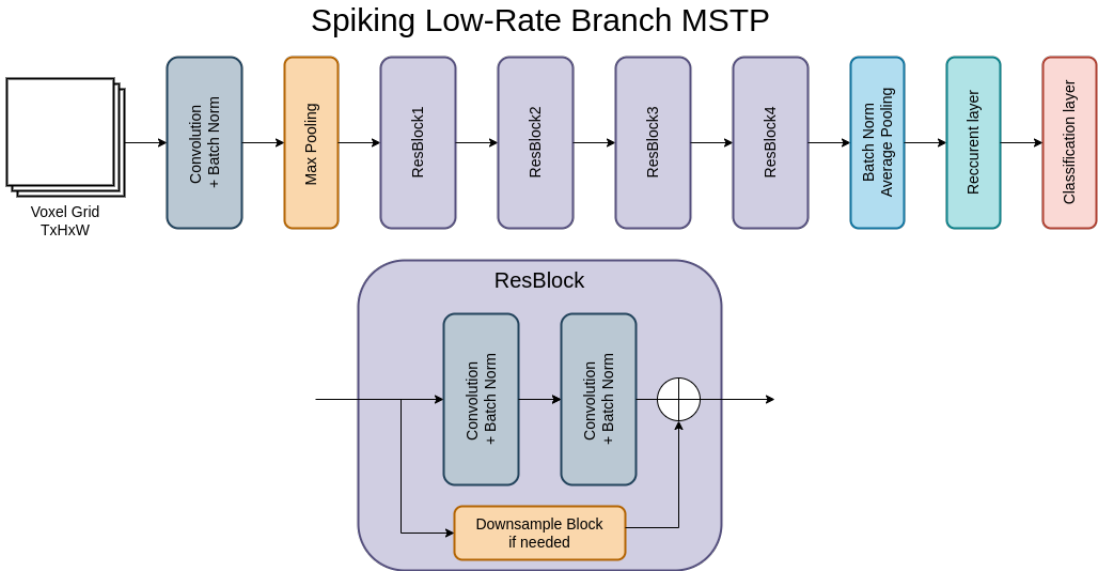


Figure 13: The overhaul spiking low-rate branch MSTP topology along with the ResBlock that we implemented during this project.

front. The topology of our spiking MSTP low-rate branch is presented in fig 13. Compared to the previous basic SNNs presented, there was no need to add batch normalisation layers here, as the base model already employed those. Overall, little modification have been made to the original model, and the resulting spiking low-rate branch MSTP ended-up being very similar to the spiking element-wise residual network (SEW-ResNet) presented by Fang et al. (2021a).
One key difference between our spiking adaptation of MSTP and the original, lies in the gated recurrent units layer (GRU) employed towards the end of the model. Recurrent neural networks (RNN) in general, are typically employed when information has to be extracted from temporal sequences. Their output depends both from their current input, and their hidden state. GRU in particular have been introduced by Chung et al. (2014), and allows each neuron to learn how much of the previous information needs to be forgotten, and how much of the current new input should be memorized. Though a spiking LSTM layer (adaptable into a spiking GRU) has been proposed by Lotfi Rezaabad and Vishwanath (2020), we found those to give very underwhelming results in our case. Especially when trying to stick to the original 3 layers bidirectional GRU used in MSTP, our network exhibited a very slow and inefficient learning process. These types of layers don't seem to be very widely used in the SNN literature, and we ultimately decided to try to find alternatives for extracting temporal information. We will describe in further details in the next section what techniques we tried using to replace this GRU layer.

3.3.3 Experiments

Experiments with SNNs were performed on the DVS-Lip dataset using 30 timesteps (T=30 in Equation 1), and were meant to help us find the most promising spiking topology before doing a comparison on the i3s_lip_reading dataset between MSTP and our best SNN. We used parametric leaky integrate and fire neurons (PLIF, see Fang et al. (2021b)) with Adam optimizer with a learning rate of $1e^{-3}$, and a cosine annealing scheduler to adjust the learning rate during

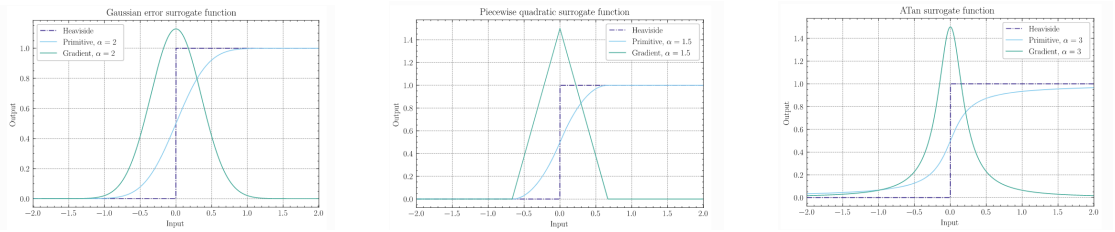
training in a manner similar to the work of Cordone et al. (2022). The PLIF neuron in particular is interesting, as these behave similarly to regular leaky integrate and fire (LIF) neurons, but with a learnable time constant. This means that the speed at which the membrane potential of those neurons will leak will be learned during training. Their membrane potential evolves following Equation 3.

$$V[t] = V[t-1] - \frac{1}{\tau}(V[t-1] - V_{reset}) + X[t] \tag{3}$$

with V[t] the membrane potential at time t, V_{reset} the resting potential, X[t] the neuron's input at time t, and τ the time constant that will be learned.

Using these settings, we ran experiments focused on the following points:

First, we needed to decide which surrogate activation function to use. As introduced earlier, surrogate gradient descend requires to choose an activation function to replace the regular non-differentiable step function used by spiking neurons. Amongst the most prominent ones we tried the Gaussian error surrogate function (Erf), piecewise quadratic, and ATan. Those functions are shown in Figure 14.



(a) Erf (b) Piecewise quadratic (c) ATan

Figure 14: Three surrogate activation functions tested (taken from the SpikingJelly documentation Fang et al. (2020b)).

Then, we wanted to find out which of the presented topologies gave the best results. Even though we suspected our spiking MSTP to perform the best, we still wanted to try basic models to establish a spiking baseline. Especially in this case, the basic models presented come from the literature. It is thus even more interesting to compare the results between published SNNs topologies, along with the one we propose.

Finally, we aim to find the best way to replace or adapt the GRU layer from MSTP in our spiking adaptation. As mentioned, using a 3 layers bidirectional GRU yields underwhelming results. The point of using GRU is to help extract temporal information, but other methods to do this exist. We mainly tested linearly recurrent spiking neurons, and stateful synapses. We also tried to replace the GRU layer by a simple fully connected spiking layer. linearly recurrent spiking neurons are regular spiking neurons with a recurrent linear connection that makes their current output also depend on their previous one (in a manner similar to vanilla RNN). Stateful synapses are placed after a spiking layer, and provide additional memory by accumulating input spike current, making their output depend from both present spike input and previous ones (see Fang et al. (2020a)). Fig 15 shows an example of how those synapses behave given some spikes in the previous layer.

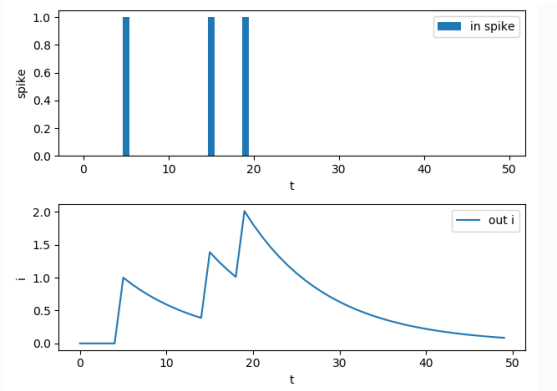


Figure 15: Output of the stateful synapse given some input spikes (taken from SpikingJelly documentation Fang et al. (2020b)).

Those synapses can also be seen as leaky integrate neurons that instead of firing spikes, simply output their membrane potential. Once again, the speed at which the current is being leaked is defined by a learnable τ parameter in manner similar to Equation 3.

From these premises, we ran tests in several batches: first we ran fast tests runs on a subsection of the DVS-Lip dataset to quickly find the most promising surrogate function to use. For these, we picked our smallest model, SNN1. We ran 50 epochs on a subset of 10 words using the Erf, piecewise quadratic, and ATan functions. Then, we ran full-on training sessions of 100 epochs using the whole dataset for each topology presented (SNN1, SNN2, and spiking MSTP), using the best surrogate function found during the previous training session. Finally, we tried different versions of spiking MSTP using either no GRU layer (replaced GRU with fully connected layer), replaced it with a linear recurrent piking neurons layer, replaced it with a fully connected layer with stateful synapses, and finally placing stateful synapses after each spiking layer.

All experiments were performed on a laptop with an Intel Core i9-12950HX CPU (2.5 GHz x 16), 62,5 GB RAM and an NVIDIA RTX A5500 laptop GPU with 16 GB of VRAM.

4 Results

4.1 Deep learning approach with MSTP

The results of the experiments carried out in the course of the project will be presented and discussed in this chapter. The implementation can be found and reproduced with the repository marceey/MSTP_on_i3s on github. The experiments were first performed on a subset of the dataset containing the nine classes 'Addition', 'Carnaval', 'Decider', 'Ecole', 'Fillette', 'Huitre', 'Joyeux', 'Musique' and 'Pyjama'. The 42 samples in each class were split randomly into a test set of nine samples and a train set of 32, which is roughly a 80% to 20% split. The model produced an accuracy of 85.2% accuracy with the original event stream of 640x480 pixel resolution (see Table 1). Downscaling the events with any of the three factors helped improve this accuracy. The best downscaling factor for the event count method turned out to be 3, resulting in 97,5%

accuracy - more than 10% improvement over the baseline model.

Models	Experiment	Test Accuracy
1	Initial State	0.852
2	Downscaling 2 — Centered Crop	0.938
3	Downscaling 3— Centered Crop	0.975
4	Downscaling 4 — Centered Crop	0.914

Table 1: Subset of dataset with random split

After these experiments, the train-test split has been changed. Similar to Tan et al. (2022) approach, the train set was now composed of the first 13 participants, while users 12 to 14 made up the test set. The split ratio therefore remained the same. The idea behind this split is that the speakers corresponding to the training set and the test set do not overlap. Thus, the model is evaluated on unseen individuals. As expected, this had a negative impact on the accuracy of the model, resulting in 61, 7% accuracy for the initial state (see Table 2). The experiments with downscaling the input stream did not prove to be helpful in improving this accuracy. In the contrary, the measured accuracies were at least 5% worse than the baseline. This was a surprising finding after the analysis performed in Chapter 3.2.2 as well as the improvements observed for the random train-test split earlier.

A possible explanation might be the fact that convergence in Deep Learning is dependent on the initialization of the model. Since the initialization at the start of the learning is performed randomly, the model could sometimes converge to a good solution, while other training runs converge to a poor solution stuck in a local minimum. It should be mentioned, that some of the runs for the same experiment have been performed several times, noting down only the maximum accuracy in the result table. It was observed that the same set up sometimes delivered accuracies differing up to 12% between training runs.

Models	Experiment	Test Accuracy
1	Initial State	0.617
2	Downscaling 2 — Centered Crop	0.567
3	Downscaling 3 — Centered Crop	0.555
4	Downscaling 4 — Centered Crop	0.444

Table 2: Subset of dataset with individual split

Lastly, the experiments were scaled to the full i3s dataset with all 70 classes. The test-train split was performed based on individuals according to the prior experiments and the approach by Tan et al. (2022). Furthermore, the experiment with a downscaling factor of 4 was omitted since its effect on the accuracy has been the worst in the earlier experiments.

The experiments carried out showed that both downscaling factors could improve the baseline performance of 33,0% accuracy (see Table 3). While the downscaling factor 2 resulted in 37,8% the best result has been achieved by a downscaling factor 3 with a final accuracy of 41,1%.

4.2 Approach with Spiking Neural Networks

We will now present the result of our experiments using various SNNs topologies and hyperparameters for classifying the DVS-Lip dataset.

Models	Experiment	Test Accuracy
1	Initial State	0.330
2	Downscaling 2 — Centered Crop	0.378
3	Downscaling 3 — Centered Crop	0.411

Table 3: Whole dataset with individual split

First, Table 4 shows the results of our first batch of trainings, that were meant to help us choose a surrogate activation function to use for the rest of the project.

Models	Experiment	Test Accuracy
SNN1	using Erf	0.546
SNN1	using Piecewise	0.531
SNN1	using ATan	0.534

Table 4: SNN1 accuracy on a subset of 10 classes from DVS-Lip, using different surrogate functions. Those classes correspond to the words: allow, allowed, america, american, benefit, benefits, billion, called, challenge, and change.

The choice of surrogate functions can vastly influence how our networks perform. This preliminary test shows Erf to perform slightly better than ATan and Piecewise, and we thus kept using it for the rest of the project. These short training sessions allowed us to save time by only doing full-on training runs on the whole dataset using the most promising surrogate function. These preliminary tests concluded, we then tested the three topologies presented earlier to see which one performs the best. Table 5 shows their respective performances.

Models	Experiment	Test Accuracy
SNN1	base model	0.395
SNN2	base model	0.514
Spiking MSTP	GRU replaced by fully connected spiking layer	0.522

Table 5: SNN experimentations results on the entire DVS-Lip dataset

This second batch of experiments showed that using deeper, more elaborate models really did allowed to improve performances significantly when compared to our simplest model (SNN1). In this state, our spiking MSTP only obtain significantly better results than SNN2. At this stage, the spiking MSTP tested did not use any recurrent layer, and the original GRU was simply replaced by a spiking fully connected one. We thus wanted to experiment with possible GRU replacement in order to see if performances could be gained by using other temporal information extraction methods, as described in the previous section. The results of our third batch of experiments, and final accuracy for the DVS-Lip Dataset using SNN are presented in Table 6.

This last Table shows that significant performance growth can be gained by using stateful synapses either as a spiking replacement for the GRU layer, or by adding those all throughout our network after each spiking layer. We also included the accuracy of the original MSTP published by Tan et al. (2022), which is still higher than our best spiking model. Our work however, demonstrates that SNNs can be a good fit for this type of task, and with more refinement, could maybe equal the current state-of-the-art while being much more energy efficient.

Models	Experiment	Test Accuracy
MSTP	**Original ANN from Tan et al. (2022)**	**0.721**
Spiking MSTP	No GRU	0.522
Spiking MSTP	1 layer spiking bidirectional GRU	0.463
Spiking MSTP	GRU replaced by linear recurrent spiking neurons	0.476
Spiking MSTP	**GRU replaced by stateful synapse**	**0.602**
Spiking MSTP	stateful synapses placed after each spiking layer	0.575

Table 6: Spiking MSTP experimentations results on the entire DVS-Lip dataset, trying different GRU substitutions.

4.3 Comparison

As final results, we will now compare the performances of MSTP and our spiking model on both datasets.

Models	Dataset	Experiment	Test Accuracy
MSTP	**DVS-Lip**	**ANN from Tan et al. (2022)**	**0.721**
Spiking MSTP	DVS-Lip	GRU replaced by stateful synapse	0.602
MSTP	**i3s dataset**	**ANN from Tan et al. (2022)**	**0.411**
Spiking MSTP	i3s dataset	GRU replaced by stateful synapse	0.081

Table 7: Results comparison between MSTP and spiking low-rate branch of MSTP on the DVS-Lip and i3s lip-reading (downscaled with a factor 3) datasets.

Table 7 shows that our SNN was able to get promising results on the DVS-Lip dataset. On the i3s dataset however, the ANN seems to be much more tolerant to the irregularities mentioned in Part 3.2.2, whereas our SNN seems to struggle learning on this dataset.

5 Conclusion

5.1 Limitations

Several limitations to the research project have impacted our results:

1. Limited time for experimentation: The project was limited by the amount of time available to conduct experiments. This led to a limited pre-processing of the data as well as the number of trials and a lack of opportunity to refine the model for better performance.

2. Computational resources and technical difficulties: Running complex models like MSTP can be computationally expensive, and without enough resources some experiments could not be run efficiently or at all. Even simple SNNs networks take a lot of time to train, as simulating neuromorphic models on conventional computer architecture is very demanding. Furthermore, the project was limited by technical difficulties such as memory errors or GPU/Cuda errors especially in the early phase of the project. These issues were based on insufficient personal computational resources leading to delays in the project, the advancement of understanding MSTP, and how to efficiently train SNNs for this task. These were only resolved in a later stage of the project.

3. Difficulty in understanding the model: The MSTP model is complex and not well commented, making it difficult to understand and customize. This could have resulted in suboptimal parameter settings.

4. The lack of studies applying advanced SNNs for video classification made the process of coming up with new topologies to test a lot slower. As already mentioned, most SNNs used for video classification are still pretty simple, and usually only try to classify fairly easy datasets. The fact that we wanted to show comparative results between MSTP (state-of-the-art for lip-reading) and an SNN, made this even harder, as obtaining results that are even remotely close to those of MSTP demands a lot of work.

It is important to acknowledge these limitations when interpreting the results of the research project. It may be necessary to conduct further experiments, improve data pre-processing, obtain more computational resources, and improve the understanding of the MSTP model to obtain better results. On the SNN side, we think that a lot of improvement can still be made, but in the absence of published complex SNNs for similar tasks, we hope to provide a spiking baseline for future works.

5.2 Perspective

In conclusion, the primary goal to adapt the state-of-the-art MSTP model to the i3s dataset, and to compare the results with those of an SNN was successful. However, the new dataset did not yield optimal results despite several experiments and modifications due to various limitations, including time, pre-processing of input data, computational resources, difficulty in understanding the MSTP model, and technical difficulties. The final best accuracy of MSTP amounts to 41,1% using the event count reduction method with a downscaling factor of 3 on the full i3s dataset. While it is not state of the art, it is still a good classifier, especially considering that this lip reading task would be hard even for humans, and far better than random. Moreover, the project provides valuable insights for future work in this area, and we can say that we proposed the first spiking neural network for automatic lip-reading.

The results of the project highlight the importance of careful pre-processing of i3s dataset, while the analysis performed outlines a strategic foundation for further pre-processing steps. The challenges encountered in understanding and customizing the MSTP model also underscore the need for better documentation and accessibility of complex models. It also shows that surrogate gradient descent does provide an worthwhile option for supervised training of SNNs. We overall, showed results that are coherent with the current consensus around this method of training, and our final spiking model shows potential to have competitive results with those of a regular ANN.

Looking forward, further work in this area could explore the outlined pre-processing on the dataset, namely a mouth detection step followed by a mouth-centered crop and resizing of the data. This would realize a uniform data representation independent of the participants distance to the camera or the position of their mouth in the frame. In addition, it might also be worth to work on the length of the video recording. As shown in the analysis, since all recordings last 3 seconds but most of the words don't take nearly as long to pronounce, there is a lot of unnecessary data captured, roughly about two third of a recording. Cutting the event streams accordingly would result in a more efficient training while potentially impacting the performance of the model positively. Efforts could also be focused on improving the MSTP model through customization and tuning of hyperparameters.
We think that improvements can still be made on our final SNN, especially in regard of the

way we replaced the GRU layer from MSTP. Our stateful synapses allowed us to break the 60% accuracy line, but we still think that most of the performance gap between MSTP and this model lies in this replacement. Given more time, most of our efforts in trying to equal MSTP would go into finding more efficient ways to extract information from the temporal component of our data. Furthermore, even though surrogate gradient descent is the current best training method for supervised learning for SNN, it is still far from perfect. By forcing gradient descent this way, we bring a lot of the limitations of regular ANN to our SNNs, while also using a lot approximations during training. In our personal opinions after this project, surrogate gradient descent is great for creating energy-efficient networks able to rival with regular ANN, but is ultimately giving-up on some of the core aspects that make SNN such promising bio-inspired models. We thus hope that other training methods will be developed in future, allowing us to tap into the full potential of SNN.

To sum up, while the project did not achieve great results for the classification of the i3s dataset, it provides a solid baseline by successfully adapting the MSTP and a detailed analysis of current shortcomings of the data set. And while the proposed SNN does not improve over the current state-of-the-art, it still constitutes the first attempt at challenging this task using an advanced deep SNN. By addressing the identified limitations, future work has the potential to yield improved results and advance the field.

References

Afouras, T., Chung, J. S., and Zisserman, A. (2018). LRS3-TED: a large-scale dataset for visual speech recognition. *CoRR*, abs/1809.00496.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252.

Anina, I., Zhou, Z., Zhao, G., and Pietikäinen, M. (2015). Ouluvs2: A multi-view audiovisual database for non-rigid mouth motion analysis. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 1:1–5.

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J. S., Senior, A. W., Vinyals, O., and Zisserman, A. (2016). Lip reading sentences in the wild. *CoRR*, abs/1611.05358.

Chung, J. S. and Zisserman, A. (2017). Lip reading in the wild. pages 87–103.

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (2006). An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424.

Cordone, L., Miramond, B., and Thierion, P. (2022). Object detection with spiking neural networks on automotive event data. *arXiv preprint arXiv:2205.04339*.

Cox, S. J., Harvey, R. W., Lan, Y., Newman, J. L., and Theobald, B.-J. (2008). The challenge of multispeaker lip-reading. In *AVSP*, pages 179–184. Citeseer.

Czyzewski, A., Kostek, B., Bratoszewski, P., Kotus, J., and Szykulski, M. (2017). An audio-visual corpus for multimodal automatic speech recognition. *Journal of Intelligent Information Systems*, 49.

Fang, H., Shrestha, A., Zhao, Z., and Qiu, Q. (2020a). Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network. *arXiv preprint arXiv:2003.02944*.

Fang, W., Chen, Y., Ding, J., Chen, D., Yu, Z., Zhou, H., Masquelier, T., Tian, Y., and other contributors (2020b). Spikingjelly. https://github.com/fangwei123456/spikingjelly. Accessed: YYYY-MM-DD.

Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., and Tian, Y. (2021a). Deep residual learning in spiking neural networks. *Advances in Neural Information Processing Systems*, 34:21056–21069.

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., and Tian, Y. (2021b). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2671.

Feng, D., Yang, S., Shan, S., and Chen, X. (2020). Learn an effective lip reading model without pains. *CoRR*, abs/2011.07557.

Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., and Scaramuzza, D. (2022). Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180.

Goldschen, A. J., Garcia, O. N., and Petajan, E. D. (1997). Continuous automatic speech recognition by lipreading. In *Motion-Based recognition*, pages 321–343. Springer.

Gruel, A., Martinet, J., Serrano-Gotarredona, T., and Linares-Barranco, B. (2022). Event data downscaling for embedded computer vision. In *17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, Online, Portugal.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Koller, O., Ney, H., and Bowden, R. (2015). Deep learning of mouth shapes for sign language. volume 2015-, pages 477 – 483. IEEE.

Lotfi Rezaabad, A. and Vishwanath, S. (2020). Long short-term memory spiking networks and their applications. In *International Conference on Neuromorphic Systems 2020*, pages 1–9.

Lu, Y. (2022). Lip-reading event data classification - final internship report.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671.

Martínez, B., Ma, P., Petridis, S., and Pantic, M. (2020). Lipreading using temporal convolutional networks. *CoRR*, abs/2001.08702.

Matthews, I., Cootes, T., Bangham, J., Cox, S., and Harvey, R. (2002). Extraction of visual features for lipreading. *IEEE Trans. on Pattern Analysis and Machine Vision*, 24(2):198–213.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Messer, K., Matas, J., Kittler, J., Luettin, J., and Maître, G. (1999). Xm2vtsdb: The extended m2vts database.

Nadafian, A. and Ganjtabesh, M. (2020). Bio-plausible unsupervised delay learning for extracting temporal features in spiking neural networks. *arXiv preprint arXiv:2011.09380*.

Patterson, E. K., Gurbuz, S., Tufekci, Z., and Gowdy, J. N. (2002). Cuave: A new audio-visual database for multimodal human-computer interface research. *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2:II–2017–II–2020.

Paugam-Moisy, H. and Bohte, S. M. (2012). Computing with spiking neuron networks. *Handbook of natural computing*, 1:1–47.

Petajan, E. (1985). Automatic lipreading to enhance speech recognition. *Proc. CVPR'85*.

Petridis, S. and Pantic, M. (2016). Deep complementary bottleneck features for visual speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 2304–2308. IEEE Press.

Sabatier and Pietrzak (2022). Création d'un jeu de données de classification de données événementielles. Projet de TER Université Côte d'Azure. Accessed: YYYY-MM-DD.

Shillingford, B., Assael, Y., Hoffman, M. W., Paine, T., Hughes, C., Prabhu, U., Liao, H., Sak, H., Rao, K., Bennett, L., Mulville, M., Coppin, B., Laurie, B., Senior, A., and de Freitas, N. (2018). Large-scale visual speech recognition.

Shrestha, S. B. and Orchard, G. (2018). Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31.

Steffen, L., Reichard, D., Weinland, J., Kaiser, J., Roennau, A., and Dillmann, R. (2019). Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms. *Frontiers in neurorobotics*, 13:28.

Tan, G., Wang, Y., Han, H., Cao, Y., Wu, F., and Zha, Z.-J. (2022). Multi-grained spatio-temporal features perceived network for event-based lip-reading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20094–20103.

Yang, S., Zhang, Y., Feng, D., Yang, M., Wang, C., Xiao, J., Long, K., Shan, S., and Chen, X. (2018). LRW-1000: A naturally-distributed large-scale benchmark for lip reading in the wild. *CoRR*, abs/1810.06990.

Yao, M., Gao, H., Zhao, G., Wang, D., Lin, Y., Yang, Z., and Li, G. (2021). Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230.

Yuhas, B. P., Goldstein, M. H., and Sejnowski, T. J. (1989). Integration of acoustic and visual speech signals using neural networks. *IEEE Communications Magazine*, 27(11):65–71.

Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., and Tian, Y. (2022). Event-based video reconstruction via potential-assisted spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3594–3604.