applied sciences

MDPI

Article

Lipreading Using Liquid State Machine with STDP-Tuning

Xuhu Yu, Zhong Wan, Zehao Shi and Lei Wang *

The College of Computer Science, National University of Defence Technology, Changsha 410073, China
* Correspondence: leiwang@nudt.edu.cn

Abstract: Lipreading refers to the task of decoding the text content of a speaker based on visual information about the movement of the speaker's lips. With the development of deep learning in recent years, lipreading has attracted extensive research. However, the deep learning method requires a lot of computing resources, which is not conducive to the migration of the system to edge devices. Inspired by the work of Spiking Neural Networks (SNNs) in recognizing human actions and gestures, we propose a lipreading system based on SNNs. Specifically, we construct the front-end feature extractor of the system using Liquid State Machine (LSM). On the other hand, a heuristic algorithm is used to select appropriate parameters for the classifier in the backend. On small-scale lipreading datasets, our recognition accuracy achieves good results. We claim that our network performs better in terms of accuracy and ratio of learned parameters compared to other networks, and has superior advantages in terms of network complexity and training cost. On the AVLetters dataset, our model achieves a 5% improvement in accuracy over traditional methods and a 90% reduction in parameters over the state-of-the-art.

Keywords: lipreading; liquid state machine; STDP

Citation: Yu, X.; Wan, Z.; Shi, Z.; Wang, L. Lipreading Using Liquid State Machine with STDP-Tuning. *Appl. Sci.* **2022**, *12*, 10484. https://doi.org/10.3390/app122010484

Academic Editor: Vincent A. Cicirello

Received: 10 September 2022
Accepted: 11 October 2022
Published: 17 October 2022

1. Introduction

Language perception is a multimodal process, which is one of the most effective ways of conveying information. Communication can be achieved not only through sound but also through visual observation, such as the movement of lips, teeth, and body. Lipreading is the task of recognizing speech by observing lip movements, also known as visual speech recognition. Lipreading is useful in many situations. For example, in extremely harsh acoustic environments, visual information plays a more important role than audio features, and lipreading can improve intelligibility in noisy conditions [1]. In the field of security, lip-language passwords are generated using lipreading technology as biometric identification [2]. In addition, hearing-impaired listeners can understand spoken language more effectively with visual clues [3].

Lipreading systems have evolved from traditional hand-crafted methods to end-to-end deep learning networks, which mainly consist of three parts: preprocessing, extraction, and classification. Petridis et al. [4] propose a method to extract deep bottleneck features directly from pixels, using long short-term memory (LSTM) to train the model, which achieves 58.1% accuracy on the AVLetters dataset. Mesbah et al. [5] proposed a new structure HCNN based on Hahn moments and Convolutional Neural Networks (CNN), which achieved 59.23% accuracy on the AVLetters dataset. However, most models ignore the temporal features of the input, transforming the time-dependent input into a static input. The latest lipreading research focuses on large-scale datasets, such as the LRW English word dataset and the LRS2 sentence dataset. Martinez et al. [6] proposed Multi-Scale Temporal Convolutional Networks (MS-TCN) to improve temporal encoding and improve word-level lipreading performance. Afouras et al. [7] used the Transformer model combined with the Connectionist Temporal Classification (CTC) [8] loss function to achieve sentence-level recognition tasks, and achieved good results on the LRS2 sentence dataset. Due to the data sample size, these methods are not suitable for tasks with small data volumes.

In addition, traditional CNNs require a lot of computing resources, so it is necessary to explore alternative lightweight and energy-efficient lipreading methods. Neil et al. [9] used dynamic visual sensors and dynamic auditory sensors to transcribe an existing lipreading dataset into an event data stream and used a CNN network to fuse the data streams of the two sensors. This work is the first time to introduce the lip language problem into event domain processing, which is different from the traditional image and video processing. However, this work adopts the method of the artificial neural network and does not explore the method of spiking neural networks that is more suitable for neuromorphic sensors.

Spiking Neural Network (SNN) [10] is a third-generation artificial neural network inspired by brain science, which is composed of biological neuron models. Because of its event-driven characteristics and low power consumption, it has received more and more attention. LSM [11] is a recurrent neural network based on spiking neurons. It can convert the input spike train into a liquid state. LSM has strong advantages in spatiotemporal data processing, such as gesture recognition and speech signals. These studies [12,13] show the superior performance of LSM in processing spatiotemporal data. However, the method based on a spiking neural network has not been deeply studied in the field of lipreading tasks.

Communication between people is not only through voice communication but also involves the interaction of various information such as lipreading, movements, and expressions, and the human brain can handle such tasks well. The spiking neural network as the third-generation neural network inspired by the brain should be more suitable for such tasks. So this paper introduces the spiking neural network into the field of lipreading. In particular, this work considers the spiking neural network's ability to model spatiotemporal sequences and the characteristics of low computational consumption. Based on Liquid State Machines (LSM), we propose a spiking lipreading network model for small-scale lipreading tasks. To improve model performance, we apply STDP rules to our model. Furthermore, we explore the best classifier structures using a heuristic algorithm. In summary, the main contributions of this paper are as follows:

- We design an LSM structure model with STDP-tuning, which consists of 512 spiking neurons, for the lipreading task.
- Under the condition that the LSM structure is fixed, we employ a heuristic search algorithm to search for the best classifier structure parameters.
- On the AVLetters dataset, our model achieves a 5% improvement in accuracy over traditional methods. In comparison with state-of-the-art models, the number of parameters of our model is reduced by 90%.

2. Background

2.1. Liquid State Machine

The liquid state machine (LSM) is one specific form of reservoir computing. Maass et al. [11] first proposed an LSM that uses randomly interconnected spiking neurons as a feature extractor. The structure of LSM is shown in Figure 1, which mainly consists of three parts: an input layer, a liquid layer of recursively connected spiking neurons, and a readout layer (through training to decode the liquid layer information), the core of which is the liquid layer.

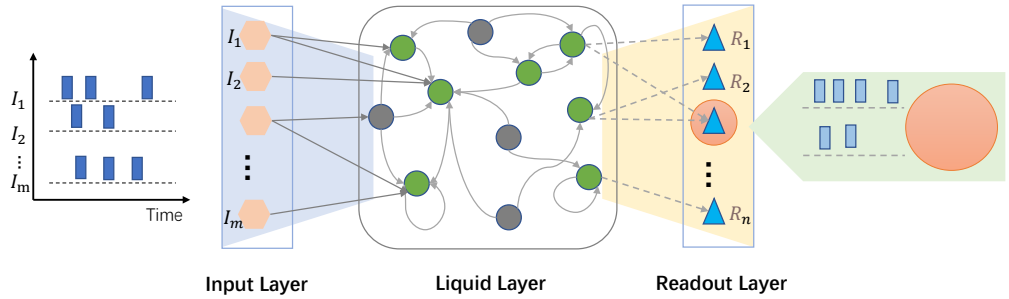

Figure 1. The structure of LSM.

The input layer is sparsely connected to the neurons in the liquid layer, which provides input spike trains into the liquid layer. For the liquid layer, each input produces a response in the liquid layer, and different inputs produce different responses, which are called liquid states. The readout layer is the function f that converts this liquid state into a feature vector. The function should be a memoryless function [14]- it has no memory. The output $y(t)$ can be written as a function of the liquid $x^M(t)$ in Equation (1).

$$y(t) = f^M(x^M(t)) \tag{1}$$

2.2. STDP Learning Rules

The Spike-time-dependent plasticity (STDP) learning rule proposed by Henry Markram [15], is an improvement on the classic Hebbian learning rule [16] with improved temporal asymmetry. It adjusts the strength of the connection between neurons according to the order of neuron learning. If the presynaptic neuron fires a spike before a postsynaptic neuron, the connection between the two neurons strengthens. Conversely, if a presynaptic neuron spikes after a postsynaptic neuron, there is no correlation between the two neurons. The connection weights between neurons are weakened. Equation (2) gives the weight correction model under the STDP rule.

$$\Delta W = \sum_{t_{pre}} \sum_{t_{post}} W(t_{pre} - t_{post}) \tag{2}$$

In the above formula, the ΔW of synaptic weight change is the sum of all pre-synaptic spike time and post-synaptic spike time on the function W, and the W is defined by Equation (3).

$$W(\Delta t) = \begin{cases} A_{pre} e^{-\Delta t/\tau_{pre}} & \Delta t > 0 \\ \\ A_{post} e^{-\Delta t/\tau_{post}} & \Delta t < 0 \end{cases} \tag{3}$$

The connection weights between neurons in early LSM architectures are randomly generated. Some researchers have applied STDP learning rules to the synaptic weight update process of LSM. Srinivasan et al. [17] introduced STDP to modulate synaptic weights between the input and liquid layer. Wang et al. [18] introduced STDP to modulate synaptic weight within the liquid layer.

3. Methodology

The overall processing flow of the lipreading algorithm is shown in Figure 2. The workflow is mainly divided into three parts: the transformation of the lipreading datasets, using LSM extract lipreading features, and genetic algorithm searching for optimal classifier architecture. We utilize ESIM [19], which is an event camera simulator, to convert a dataset of ordinary cameras into a dataset in the form of event cameras. In the part of lipreading feature extraction, we input the transformed samples as spike trains into the LSM. The LSM

will generate the corresponding liquid and read it out at the readout layer to get the feature vector. In the classifier architecture search part, we use the idea of a genetic algorithm with an elite strategy to automatically search for the optimal classifier network architecture suitable for this problem.

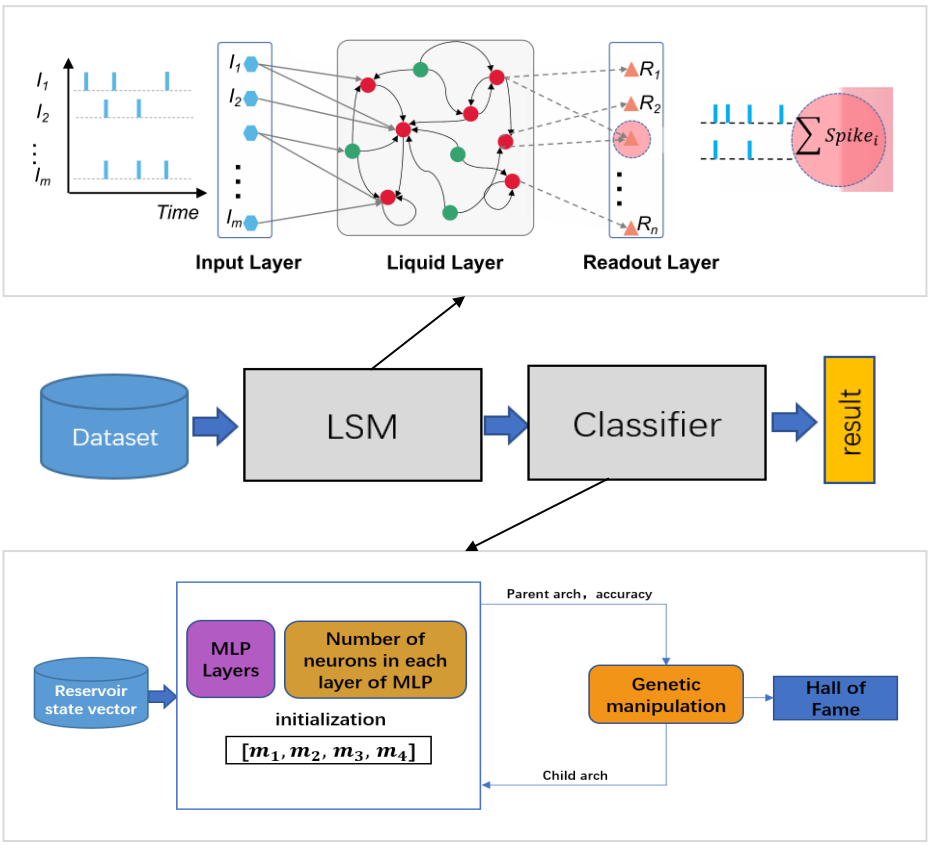

Figure 2. Workflow of proposed method.

3.1. Lipreading Datasets Transformation

ESIM is an event-based camera simulator [19], which can convert video data captured by traditional cameras into event stream data. Its implementation principle is different from the event-based camera principle. The ESIM samples each pixel independently and proposes an adaptive sampling strategy that dynamically adjusts the sampling speed according to the prediction of the visual signal.

We transform existing lipreading datasets using the ESIM simulator. Separate the grayscale video frames as input to the ESIM simulator. After processing by ESIM, a series of pixel-level event outputs are generated, one of which is denoted as (x, y, t, P). (x, y) is the coordinate position of the event in the two-dimensional space, and t is the timestamp of the event, indicating the moment when the event occurred. P is the polarity of the event and the polarity represents whether the pixel is brighter or darker than before.

3.2. Lipreading Feature Extraction

3.2.1. Liquid Structure

The LSM model in this paper uses a total of 512 neurons, which is composed of a neuron model with a cubic structure of $8 \times 8 \times 8$, which simplifies the model structure to a certain extent. Neurons are of two types: excitatory neurons and inhibitory neurons. The spikes of excitatory neurons can increase the membrane potential of postsynaptic neurons, while the spikes of inhibitory neurons can decrease the membrane potential of

postsynaptic neurons. In the liquid layer, we randomly divided 80% of them into excitatory neurons and 20% into inhibitory neurons. Both excitatory and inhibitory neurons are modeled as leaky integral-firing neurons. The kinetic Equation (4) is as follows:

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inhi} - V) \quad (4)$$

V is the variable of membrane potential and τ is the time constant. E_{rest} is the resting membrane potential. E_{exc} and E_{inhi} are the equilibrium potentials of excitatory and inhibitory synapses, respectively, and g_e and g_i are the total conductance of the excitatory and inhibitory synapses, respectively, of all connections that transmit the spike.

Synaptic connections are used between neurons and neurons, and the connection probability of synapses is affected by the distance between neurons. The connection probability $P_{i,j}$ between the i^{th} neuron and the j^{th} neuron can be defined as follows:

$$P_{i,j} = C \times e^{-(D_{i,j}/\lambda)} \quad (5)$$

The scale factor C is a parameter used to control the connection probability between neurons, which determines the upper limit of the connection probability. λ is a parameter that controls the connection distance. The connection distance between different types of neurons is determined by the value of λ.

3.2.2. STDP Learning

Based on previous work [18], we use STDP to modify some synaptic weights, including the synaptic connection weights between the input layer and the liquid layer and the synaptic connection weights inside the liquid layer, to improve the sensitivity of the network to specific pattern inputs. To achieve the effect of "strengthening causal connections and weakening non-causal connections". This improves the separation and approximation of liquids. Specifically, the input spike trains we get from the training examples are sequentially fed into the liquid. Under the STDP rule, all synaptic weights are modified and in the following training and testing process, the synaptic weights remain fixed.

3.3. Readout Layer

In the readout layer, the liquid state is converted into a feature vector, which is used for classification. Liquid state [20] refers to the state of the liquid neuron after the sample is input into the LSM. There are many ways to read the state of liquid neurons, and Figure 3a presents a traditional method, which obtains the length of the state vector of each sample equal to the number of liquid neurons.

The key to affecting the lipreading task is whether the lip motion information can be effectively extracted. In order to extract the neuron state information more finely, we adopt the time window division sampling technique [21]. As shown in Figure 3b, we divide the input time of a sample into four sub-windows equally. In each sub-window, the number of spikes of each neuron is calculated separately, so that four liquid vectors can be obtained. Finally, these four liquid vectors are spliced into a larger liquid vector as the extracted feature vector. Compared with the traditional method, the state vector generated by the time window division sampling theoretically provides more time dimension information, which is more conducive to the classification of the classifier.

In the selection of the classifier, this paper not only considers the efficiency of the classification algorithm but also considers the implementation of the algorithm on the hardware platform. On the one hand, LSM can extract the features of the data well and can get better results without more complex classifiers [22]. On the other hand, the classifier of ANN represented by MLP is relatively more friendly to hardware implementation. Therefore, we chose three commonly used classifiers. We selected three commonly used classifiers for this task. They are Multilayer Perceptron (MLP), Support Vector Machine (SVM), and K-NearestNeighbor (KNN). We compared the accuracy of these three classifiers

on the datasets. For the exploration of the structural parameters of the perceptron network, we adopted an automated method for optimization in the follow-up.

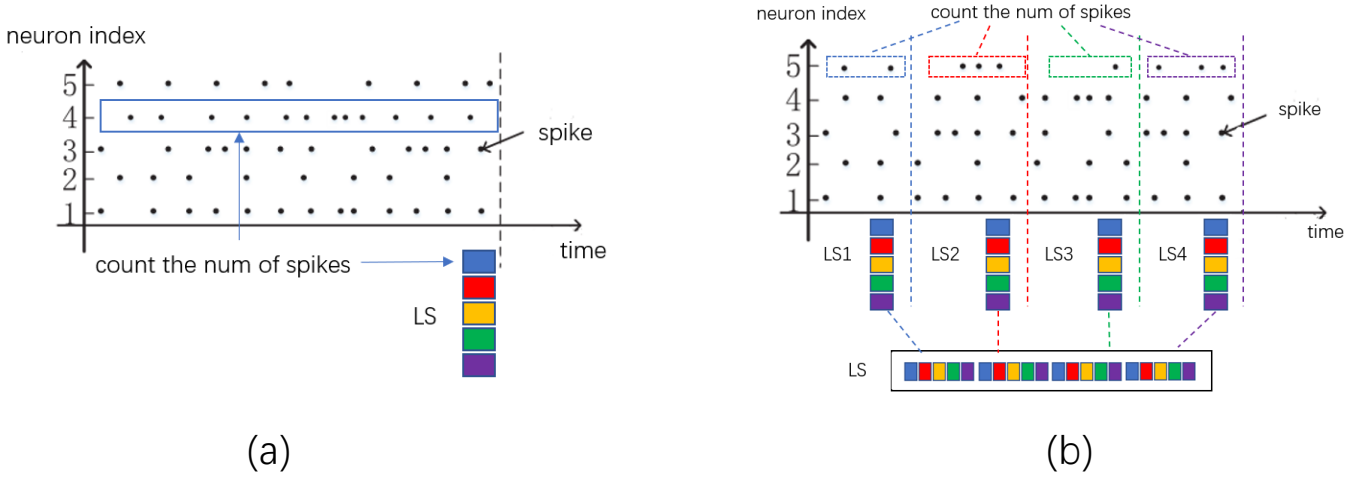

Figure 3. Two ways to generate the liquid state. (**a**) Traditional liquid state generation method. (LS represents the liquid state vector). (**b**) Time window division sampling technique.

3.4. Classifier Structure Exploration

It is well known that the number of hidden layers and nodes in a Multilayer Perceptron (MLP) will directly affect the performance of classification. The parameters for MLP are generally determined based on historical experience. Inspired by heuristic algorithm [23], this paper adopts a genetic algorithm with an elitism strategy to search for the optimal number of MLP layers and nodes. The overall frame is shown in Figure 4. First, an initial MLP network structure is randomly given. Then, according to the fitness function, the fitness score of the individual is calculated to determine whether it is good or bad, and the genetic operations of selection, crossover, and mutation are carried out to obtain a new generation of offspring. Among them, the best individuals in each generation are saved to the Hall of Fame using an elite strategy. Keep iterating until convergence or limit is reached.

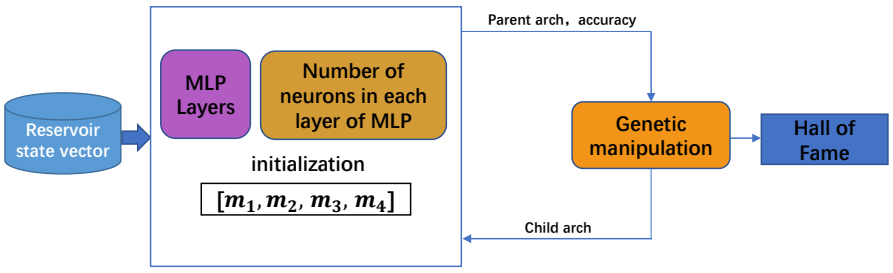

Figure 4. Exploration of classifier architecture based on heuristics.

3.4.1. Initialize

The number of layers of MLP and the number of neurons in each layer are represented by chromosomes, and we fixed the length of chromosomes, setting the maximum number of layers to four layers. We also set a "null" bit on the chromosome to enable the constructed model to terminate early. For example, to build a network with four hidden layers, the chromosome shape is as follows:

$$[m_1, m_2, m_3, m_4] \quad (6)$$

Among them, m_i represents the number of neurons in the i^{th} hidden layer. It is stipulated that when m_i is zero or negative, it is regarded as a termination signal. At the same time, in order to ensure that there is at least one hidden layer, it is mandatory to set the first parameter to always be greater than zero.

3.4.2. Fitness and Genetic Manipulation

The fitness function is also called the evaluation function, which is mainly used to measure and distinguish the quality of individuals in the group [24]. In this paper, the classification accuracy of the classifier is directly used as the fitness function. Selection [25] refers to randomly selecting a part of individuals from the parent generation to survive according to a pre-selected strategy regarding the fitness function. The tournament selection [26] method is adopted in this paper. The strategy selects the best individual from the population to keep each time. The intersection operation in this paper uses the two-point intersection, randomly setting two intersection points in an individual, and then swapping some genes. The mutation operation selects the boundary mutation and randomly selects one of the two corresponding boundary genes on the gene to replace the original gene value.

3.4.3. Elite Retention Strategy

The elite retention strategy is used to ensure that the optimal individuals appearing in the evolutionary process will not be lost and destroyed due to selection, crossover, and mutation operations [27]. Furthermore, adopting the elite retention strategy can speed up the speed and ability of global convergence of network search.

4. Experimental Setup

4.1. Experimental Platform Settings

The experiments in this paper utilize the open-source SNN simulator Brain2 [28]. It provides a description implementation of the neuron and synaptic behavior on which our model network can be built. Mainly the implementation of the input layer and the liquid layer. The implementation of the classifier uses the Keras library. The optimal classifier architecture search based on a genetic algorithm mainly uses the genetic algorithm toolbox of deap. These are implemented based on python 3.6. All SNN simulation and classifier optimization software programs in this paper are run on the CPU. The MLP classifier uses GPU to accelerate the training. The specific configuration is shown in Table 1.

Table 1. Configuration information of the experimental platform.

Hardware	Software
NVIDIA GeForce RTX 2060 Inter(R) Core i7 11700K CPU @3.6GHz	CUDA 10.0 cudnn 7.6.5 Ubuntu 18.04 Python 3.6 Brian2.4

4.2. Datasets

AVLetters

This dataset is the first audiovisual speech dataset. The data set initially contains 10 speakers, each of whom stated 26 English letters 3 times independently, for a total of 780 utterance instances [29]. After manually locating the position of the lips in each image, the entire image is cropped to 80 × 60 pixels to form the final dataset. For division of the training set, this paper adopts the same partitioning principle as in Ngiam et al. [30] and Matthews et al. [29]. The first two speaking videos of each speaker are used for training and the last one is used for testing. This means there are 520 training utterances and 260 test utterances. Under such a principle, Mesbah et al. [5] proposed a novel structure based on Hahn moments as the first layer of the convolutional neural network structure—Hahn Convolutional Neural Network (HCNN), and achieved an accuracy of

59.23%. Petridis et al. [31] proposed an end-to-end visual speech recognition system based on fully connected layers and long-short memory (LSTM) network and achieved the highest accuracy of 69.2%.

5. Experimental Result

5.1. Accuracy

Table 2 compares our method with other works on the AVLetters dataset. The experimental configuration in the table adopts the best configuration (STDP-Tuning, Classifier: a layer MLP with 240 neurons). We compared the model size and the accuracy. Our model improves accuracy by about 5% over traditional methods. Compared with some deep learning methods, we lose some accuracy, but greatly reduce the number of neurons and network model parameters. Furthermore, we performed ablation experiments for STDP-tuning. The results show that the accuracy of LSM with STDP-tuning is about 2-3% higher than the traditional LSM.

Table 2. Results on the AVLetters Dataset

Method	Parameters or Model Size	Number of Neurous	Max Accuracy
HMM	-	-	44.6%
DCT + DBNF [4]	-	-	58.1%
HCNN [5]	-	2340	59.23%
Raw + Diff Images [31]	>5 M	8900	69.2%
Our (LSM + MLP)	0.5 M	512 + 240	61.85%
Our (LSM + MLP + STDP)	0.5 M	512 + 240	64.65%

5.2. Classifier Selection

We chose three common classifiers. The first category is Support Vector Machines (SVM), a generalized linear classifier for binary classification of data. This article uses an SVM with a Gaussian kernel. The second classifier is KNN. The third classifier is an MLP classifier with only one hidden layer as a baseline for comparison, where the number of neurons is 150. We train our network using SGD (learning rate 0.01, learning rate decay factor 1×10^{-6}, batch-size 256). During training, we also used a dropout parameter of 0.1. Figure 5 shows a comparison of classification accuracy, recall, and F1 score for three classifiers.

Figure 5. Performance of different classifiers on the AVLetters Dataset.

5.3. MLP Structure Parameter Search

Next, we use a heuristic to search for the number of layers of the MLP and the number of neurons in each layer. During the search process, we fix other parts of the model, including the settings of the input layer and the LSM structure, and only change the structure of the MLP. In the genetic algorithm, we set a total of 20 iterations, and each iteration has 50 individuals. In Table 3, we list part of the MLP structures found during the search (number of layers and number of neurons in each layer) and the final accuracy. Through our experimental results, it is found that increasing the number of hidden layers alone does not improve the accuracy, or even affects the accuracy. The main reason is that the number of hidden layers in the last layer is too small, which will seriously affect the accuracy of the classifier. Therefore, the reasonable number of layers and the number of neurons in each layer restrict the final model accuracy.

Table 3. Performance of different MLP structures on the AVLetters dataset.

Layers	Number of Neurous	Accuracy (%)
4	237-128-93-28	50.93
3	246-107-6	30.45
2	173-55	55.32
1	143	60.76

Figure 6 shows the change in the overall lipreading accuracy of the model with each child generation update during the genetic search process. From the observation of the results, it can be found that the average precision of the first generation is often very low due to random selection. However, after five iterations, the average precision initially tends to be stable. Because of the elite strategy, the outstanding individuals that appear in a certain generation can save the good structure and speed up the convergence speed of the search. The final convergence accuracy is also significantly higher than that of the MLP structure we set with human experience.

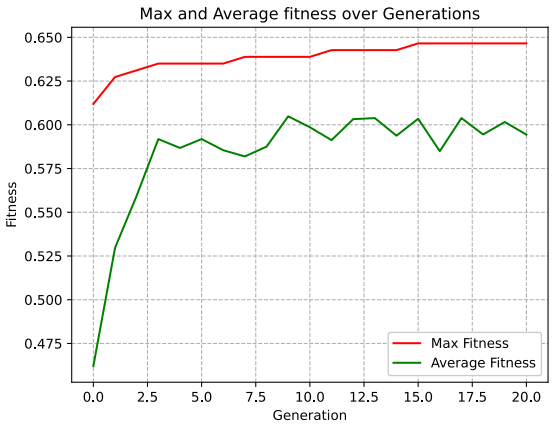

Figure 6. Iterative Algebra and Accuracy.

5.4. Discussion

5.4.1. Confusion Matrix

We observed the confusion matrix of the AVLetters dataset, as shown in Figure 7. We can find that the letters W, Y, Z, O, and A have a higher accuracy of correct recognition. However, the most common confusion is in (B, P), (C, D, T), and (Q, U) among these letter pairs, this is similar to the conclusion of Barnard. M et al. [32]. The main reason for this phenomenon is that lipreading is performed through a single visual modality by distinguishing the visemes of these letters, but different letters may produce the same visemes. This is also the reason why it is so hard for lipreading experts to solve lipreading problems.

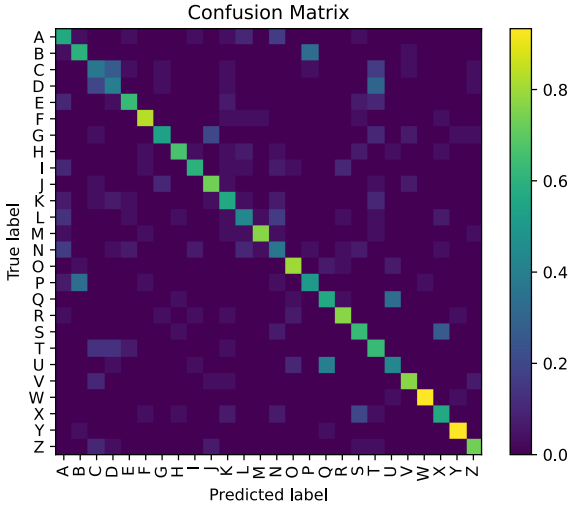

Figure 7. Confusion Matrix on the AVLetters Dataset.

5.4.2. Speakers Differences

Figure 8 shows the recognition accuracy of each speaker in AVLetters dataset. There is a significant difference in the accuracy with which different speakers are recognized, with the largest difference between speakers being around 30%. Among them, S2 and S7 have lower recognition degrees. The possible reason is that they have beards. In particular, the S2 beards lead to a decrease in the definition of the mouth contour, which may lead to a decrease in the final recognition accuracy. On the other hand, the mouth cropping of the original dataset is also not standardized. Some speakers may crop leaving part of the nose contour, while others do not. This may be the introduction of these invalid redundant information, which brings about the difference in accuracy.

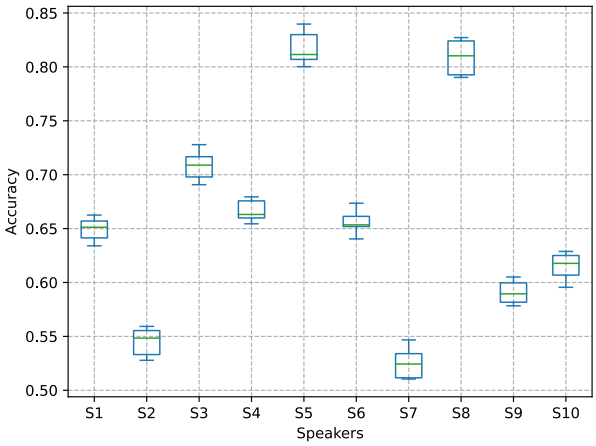

Figure 8. Per-speaker recognition accuracy on the AVLetters Dataset.

5.4.3. Limitations

The method in this paper has achieved certain results on the existing alphabet datasets, but there is still a certain gap between the actual application scenarios. On the one hand, the alphanumeric datasets are small in scale, and the effects and capabilities of spiking neural networks on large-scale datasets cannot be tested. On the other hand, the effect of practical application also depends on the generalization ability of the model. The method in this paper only stays on the data set, and the consideration of generalization ability is also the direction we will strengthen in the next step.

In addition, we conclude from the experiment that the liquid state machine is more like a feature extractor with a simple structure, which has certain advantages for timing problems. However, it can achieve certain effects for different types of problems, which is still difficult to explain and prove from the mechanism. At present, spiking neural networks are still unable to solve the difficult problems of lipreading tasks, which also explains why it is difficult for human experts to achieve completely accurate results on the problem of lip recognition.

6. Conclusions

In this paper, a Spiking Neural Network (SNN) structure based on a liquid state machine (LSM) and MLP classifier are proposed to solve the lipreading problem. This paper also uses a heuristic algorithm to search for the best MLP network structure parameters for classifying liquid states and compares the impact of different classifiers on the final accuracy. The results on the AVLetters dataset show that our method has an accuracy advantage over traditional methods, a computational advantage over partial-depth methods, and is more suitable for small-scale datasets.

As a continuation of this work, we will further explore the network structure of LSM to improve the recognition accuracy of the network. We also plan to use the idea of Spiking Neural Network (SNN) to try to solve the large-scale lipreading problem.

Author Contributions: data curation, X.Y., Z.W. and L.W.; investigation, X.Y., Z.W., Z.S. and L.W.; methodology, X.Y., Z.W., Z.S. and L.W.; software, X.Y.; visualization, X.Y. and Z.W.; validation, X.Y., Z.W., Z.S. and L.W.; resources, X.Y, Z.W. and L.W.; writing—original draft preparation, X.Y., Z.W. and L.W.; writing—review and editing, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under grant 61872136.

Data Availability Statement: The data presented in this study are available on request from corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Potamianos, G.; Neti, C.; Gravier, G.; Garg, A.; Senior, A.W. Recent advances in the automatic recognition of audiovisual speech. *Proc. IEEE* **2003**, *91*, 1306–1326.
2. Akhtar, Z.; Micheloni, C.; Foresti, G.L. Biometric liveness detection: Challenges and research opportunities. *IEEE Secur. Priv.* **2015**, *13*, 63–72.
3. Sommers, M.S.; Tye-Murray, N.; Spehar, B. Auditory-visual integration and lipreading abilities of older adults with normal and impaired hearing. *J. Acoust. Soc. Am.* **2006**, *120*, 3347–3347.
4. Petridis, S.; Pantic, M. Deep complementary bottleneck features for visual speech recognition. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 2304–2308.
5. Mesbah, A.; Berrahou, A.; Hammouchi, H.; Berbia, H.; Qjidaa, H.; Daoudi, M. Lip reading with hahn convolutional neural networks. *Image Vis. Comput.* **2019**, *88*, 76–83.
6. Martinez, B.; Ma, P.; Petridis, S.; Pantic, M. Lipreading using temporal convolutional networks. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 6319–6323.
7. Afouras, T.; Chung, J.S.; Senior, A.; Vinyals, O.; Zisserman, A. Deep audio-visual speech recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**. https://doi.org/10.1109/TPAMI.2018.2889052.
8. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006, pp. 369–376.
9. Li, X.; Neil, D.; Delbruck, T.; Liu, S.C. Lip reading deep network exploiting multi-modal spiking visual and auditory sensors. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Hokkaido, Japan, 26–29 May 2019; pp. 1–5.
10. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw.* **1997**, *10*, 1659–1671.

11. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560.
12. Zhang, Y.; Li, P.; Jin, Y.; Choe, Y. A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2635–2649.
13. Al Zoubi, O.; Awad, M.; Kasabov, N.K. Anytime multipurpose emotion recognition from EEG data using a Liquid State Machine based framework. *Artif. Intell. Med.* **2018**, *86*, 1–8.
14. Liu, R.; Maric, I.; Spasojevic, P.; Yates, R.D. Discrete memoryless interference and broadcast channels with confidential messages: Secrecy rate regions. *IEEE Trans. Inf. Theory* **2008**, *54*, 2493–2507.
15. Gerstner, W.; Kempter, R.; Van Hemmen, J.L.; Wagner, H. A neuronal learning rule for sub-millisecond temporal coding. *Nature* **1996**, *383*, 76–78.
16. Gerstner, W.; Ritz, R.; Van Hemmen, J.L. Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biol. Cybern.* **1993**, *69*, 503–515.
17. Srinivasan, G.; Panda, P.; Roy, K. Spilinc: Spiking liquid-ensemble computing for unsupervised speech and image recognition. *Front. Neurosci.* **2018**, *12*, 524.
18. Wang, Q.; Li, P. D-lsm: Deep liquid state machine with unsupervised recurrent reservoir tuning. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2652–2657.
19. Rebecq, H.; Gehrig, D.; Scaramuzza, D. ESIM: An open event camera simulator. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 969–982.
20. Kaiser, J.; Stal, R.; Subramoney, A.; Roennau, A.; Dillmann, R. Scaling up liquid state machines to predict over address events from dynamic vision sensors. *Bioinspiration Biomimetics* **2017**, *12*, 055001.
21. Tian, S.; Qu, L.; Wang, L.; Hu, K.; Li, N.; Xu, W. A neural architecture search based framework for liquid state machine design. *Neurocomputing* **2021**, *443*, 174–182.
22. Bertschinger, N.; Natschläger, T. Real-time computation at the edge of chaos in recurrent neural networks. *Neural Comput.* **2004**, *16*, 1413–1436.
23. Chu, C.W. A heuristic algorithm for the truckload and less-than-truckload problem. *Eur. J. Oper. Res.* **2005**, *165*, 657–667.
24. Baresel, A.; Sthamer, H.; Schmidt, M. Fitness function design to improve evolutionary structural testing. In Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, New York, NY, USA, 9–13 July 2002; pp. 1329–1336.
25. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197.
26. Miller, B.L.; Goldberg, D.E.; et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex Syst.* **1995**, *9*, 193–212.
27. Luo, F.; Lyu, F.; Hou, Z. An Improved Genetic Algorithm based on Elite Retention Strategy and Explosion Operators. *J. Xihua Univ. (Nat. Sci. Ed.)* **2018**, *3*, 83–88.
28. Stimberg, M.; Brette, R.; Goodman, D.F. Brian 2, an intuitive and efficient neural simulator. *Elife* **2019**, *8*, e47314.
29. Matthews, I.; Cootes, T.F.; Bangham, J.A.; Cox, S.; Harvey, R. Extraction of visual features for lipreading. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 198–213.
30. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal deep learning. In Proceedings of the ICML, Bellevue, WA, USA, 28 June–2 July 2011.
31. Petridis, S.; Wang, Y.; Ma, P.; Li, Z.; Pantic, M. End-to-end visual speech recognition for small-scale datasets. *Pattern Recognit. Lett.* **2020**, *131*, 421–427.
32. Zhao, G.; Barnard, M.; Pietikainen, M. Lipreading with local spatiotemporal descriptors. *IEEE Trans. Multimed.* **2009**, *11*, 1254–1265.