Deep Learning Based Biometric Verification Using Dynamic Lip Features

Chong-Wei Wang [1], Zheng-Yan Guo[2], Chi-Hung Chuang[3,*]
[1] Undergraduate Program in Intelligent Computing and Big Data, Chung Yuan Christian University, Taoyuan, Taiwan, R.O.C.
[2] Department of Computer Science & Information Engineering, National Central University, Taoyuan, Taiwan, R.O.C.
[3] Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan, Taiwan, R.O.C.
chchuang@cycu.edu.tw

* Corresponding Author

Abstract

With the rapid development of technology, biometric authentication systems have been widely applied in daily life in recent years, such as face recognition and voice recognition. However, enhancing authentication security remains a significant challenge. Traditional authentication methods based on static biometric features have become increasingly susceptible to breaches due to the continuous advancement of various forgery techniques. In light of this, this thesis proposes a method based on biometric feature sequences. Since this approach requires verification through sequential data, it is relatively difficult to forge. In order to enhance security, this thesis employs deep neural networks to implement an identity authentication model based on lip images and keypoint sequences, using a dataset recorded for the purposes of this research for both training and testing. In standard authentication experiments, the model trained in this thesis achieved an HTER of 8.86%, demonstrating the effectiveness of utilizing lip images and keypoint sequence data. Furthermore, to test whether inputting sequential data can enhance security, this thesis used static sequences as forged data fed into the model and obtained an FAR_S of 84.09%, indicating that directly inputting sequential data does not help improve security. To counter static sequence attacks, this thesis calculates the frame difference of image sequences as input. Finally, in general authentication experiments, an HTER of 6.53% was achieved, and in static sequence attack experiments, an FAR_S of 9.09% was attained, proving the effectiveness and security of frame difference in lip image sequences in the authentication problem.

Keywords: Face Recognition, Biometric Authentication, Lip Image, Sequence, Neural Network

1. Introduction

With the rapid development of technology, people are increasingly concerned about privacy and security. Traditional methods of identity authentication, such as using account passwords or ID cards, are gradually failing to meet these needs, leading to the emergence of biometric authentication systems. Biometric authentication systems use biometric features (e.g., gait, hand veins, facial expressions, and lip shapes) instead of specific personal identification numbers (PINs) for authentication, thereby not only enhancing security but also improving system usability.

In recent years, biometric recognition systems have been widely applied in everyday life. In addition to common features such as gait, facial characteristics, voice waveform analysis, and fingerprint images, lip information has also been proven to serve as a biometric feature for identity authentication systems [1–15]. Lip motion features are image-based features consisting of sequences of lip images captured when a person speaks, containing both static information (lip appearance) and dynamic information (motion patterns during speech). Besides being easy to acquire without requiring special sensors, lip motion features can also be combined with facial or voice information to enhance system security or adapt to various environments [3].

Because lip motion features are formed by a series of images, they are dynamic features and have better anti-spoofing capabilities compared to static features such as facial or fingerprint data. Moreover, unlike voice authentication, lip motion features do not suffer from reliability degradation in noisy environments. Based on these reasons, this thesis employs lip motion features to implement identity authentication.

Although past literature has already demonstrated the discriminative power of lip motion features, there has been little exploration of their resistance to attacks. If they cannot resist attacks, then the value of adopting sequential features is lost. Therefore, in addition to using deep learning neural networks for identity authentication, this thesis will also verify the model's resistance to static sequence attacks and propose effective improvements.

To experiment with lip images and lip keypoint data, this thesis will collect a new speaker dataset. Building upon the neural network proposed in [15], a network capable of integrating both features has been designed for experimentation. In addition to comparing performance in general authentication tasks, this thesis also conducts experiments on the resistance of these two features and frame difference data to static sequence attacks.

The contributions of the paper are summarized as follows:

- A new speaker dataset has been collected.
- A neural network that combines lip images and lip keypoints as two features for identity authentication is proposed, and its effectiveness in identity authentication is evaluated.
- The performance differences between lip images and lip keypoints are analyzed and

compared, demonstrating the effectiveness of lip keypoint data for authentication in a neural network.

- It is tested and demonstrated that directly inputting lip images or keypoint data cannot defend against static sequence attacks.
- It is proven that data processed using frame differences provides better resistance to static sequence attacks.

2. Related Works

Lip-based biometric authentication was commonly implemented using Hidden Markov Models (HMM) in its early stages. However, with the maturation of artificial neural networks in recent years, many studies have attempted to utilize convolutional neural networks to extract lip features. This section reviews and discusses relevant research in this field from the past few years.

In the experiments conducted by Shi, Xiao-Xing [14] and colleagues, the authors extracted four kinds of features from lip images: lip shape, texture, motion vectors, and locp features. They segmented the input images into different segments based on the transitions between words and within words. For each segment, they used HMM-UBM to estimate the probability that the segment belonged to a legitimate user and then calculated the weighted average of the scores from each segment to obtain the final authentication result. The experimental results on a dataset of 40 subjects showed an HTER of 1.26%. Although a relatively low error rate was achieved in a general authentication scenario, the study did not conduct any experiments regarding resistance to attacks.

In the research by Feng Cheng [15] and others, in order to prevent recorded legitimate images from easily compromising the system, a random password is generated each time the system is used for authentication. Therefore, in addition to identity verification, the system also needs to verify the correctness of the password. They employed a 3D convolutional neural network architecture to extract lip features, then fed these features into two separate neural networks for identity verification and voice verification, respectively. The final authentication result was determined based on the outputs of these two verification processes. Experimental results on a dataset of 40 subjects showed an HTER of 3.85%. Although the use of two authentication networks enhanced the system's security in [15], it also increased the training cost of the model, especially in the lip-reading component. Apart from making training more difficult, it also placed higher demands on the dataset. Moreover, that study did not individually verify the resistance to attacks of the identity authentication network that is the focus of

this thesis. If that network itself cannot withstand attacks, then the model likely relies only on static information within the sequential data, which would negate the advantages offered by sequential data.

From the above literature, it is clear that although the effectiveness of lip image sequences in authentication has been demonstrated, there has been very limited exploration of their resistance to attacks. This is a critical issue because a lack of robust anti-attack capability would make this method no different from static authentication methods such as facial recognition, failing to achieve the intended goal of using dynamic features to counter attacks. To address this problem, this thesis will employ static sequences to test the model's resistance and attempt to propose recommendations for improvement.

3. Research Content and Method

3.1. Architecture

The system architecture of this study is shown in Figure 1. Users first input a sequence of lip images and an ID. The image sequence then undergoes preprocessing before being fed into the corresponding model, and finally, the authentication result is obtained.

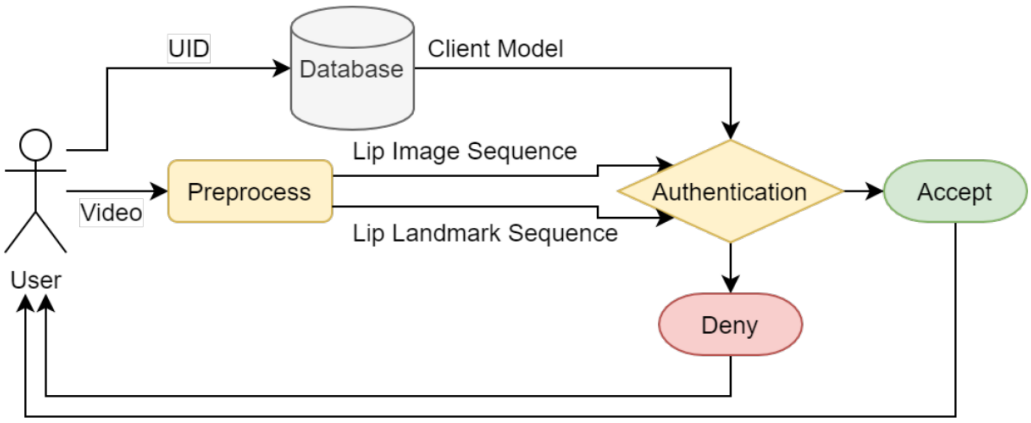


Figure 1: System Architecture

In the preprocessing stage, in addition to extracting the lip region and keypoints, histogram matching is performed on the images in the sequence to reduce the impact of lighting conditions. In order to filter out static information, this thesis also processes image and keypoint sequences using a frame-difference method and carries out relevant experiments.

For the authentication part, this thesis adopts the artificial neural network approach. Each user is associated with a binary classification model that can simultaneously or separately take lip images and

keypoint sequences as input and determine whether the input belongs to the target user. In designing the neural network, features are primarily extracted via a convolutional neural network, and classification is then performed through fully connected layers.

During the training phase, each user provides a small number of samples to train the corresponding model. The data generated during subsequent authentications can continue to be used for retraining the model, in order to achieve better performance and adaptability.

3.2. Neural Network Architecture

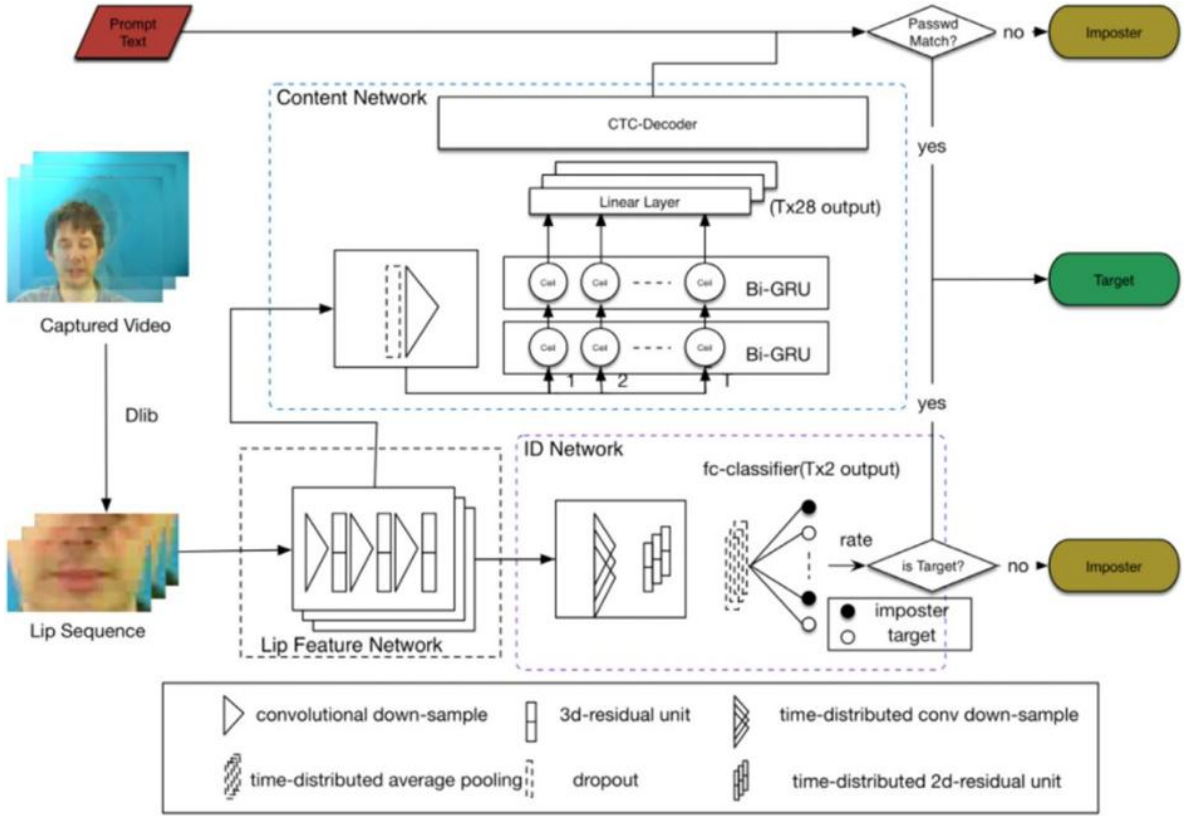


Figure 2: Neural Network Architecture Proposed in [15]

Source: [15]

The network architecture used in this thesis is based on the authentication network (Figure 2) proposed in [15]. We remove the C-Net (Content Network) responsible for lip reading tasks and retain only the LF-Net (Lip Feature Network) and ID-Net (ID Network). As shown in Figure 3, the structure consists of the LF-Net, a Time Distributed 2D convolution layer, and a 3D convolutional residual module. This network uses 3D and 2D convolutions to extract spatiotemporal features from the lip image sequence, and finally classifies them via fully connected layers.

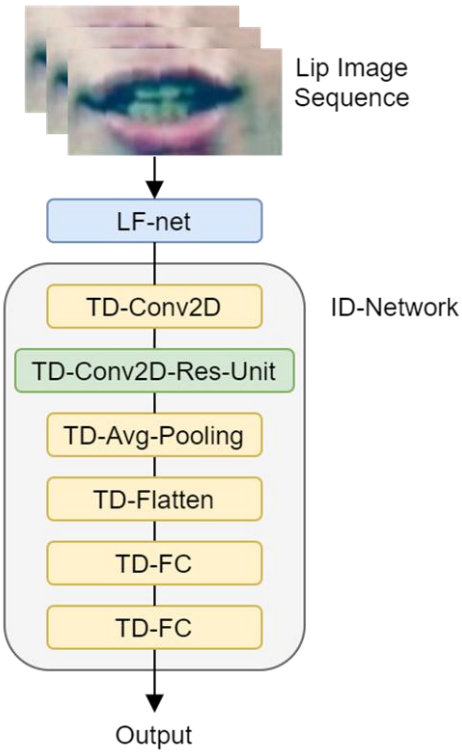


Figure 3: Base Network Architecture Figure

In Figure 3, “TD” stands for Time Distributed Layer. In the TD layer, the input at each time step is fed into the network layer separately, and the input and output have the same temporal dimension. Because of the Time Distributed method, the final authentication result is calculated by averaging the outputs at each time step. In the base network, ID-Net adopts the Time Distributed approach to locally extract temporal features from the spatiotemporal information provided by LF-Net and then classify the features for each time step.

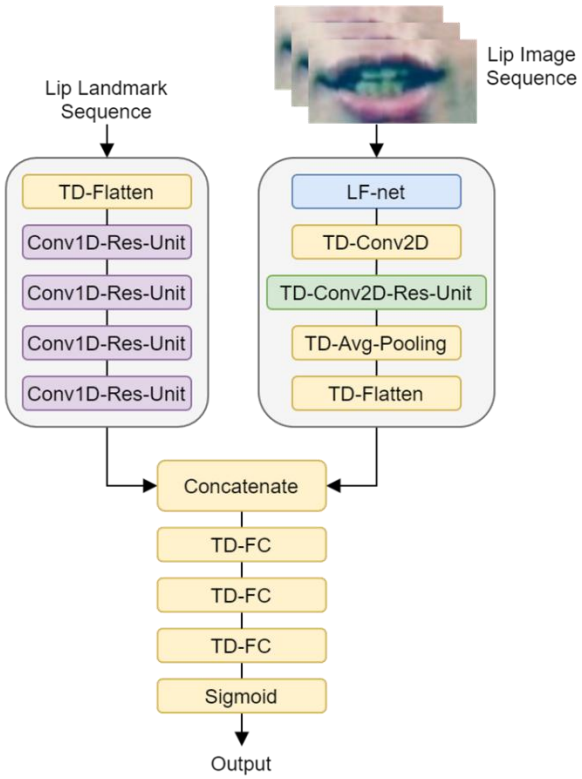


Figure 4: Neural Network for Separately Extracting Features and Then Merging

To integrate lip keypoints with image data, this thesis employs two network architectures for experiments. The first approach inputs images and keypoints into separate networks to extract features and then merges these extracted features before feeding them into a classifier, as shown in Figure 4. Detailed parameters are listed in Table 6 of Appendix A.

Building upon the base network, this architecture adds a feature extraction network for lip keypoints and merges both image and keypoint features, which are then classified. The image component is the same as in the base network, using 3D and 2D convolution layers to extract temporal and spatial features from the sequence. For the keypoints, 1D convolutions are used to extract temporal sequence features.

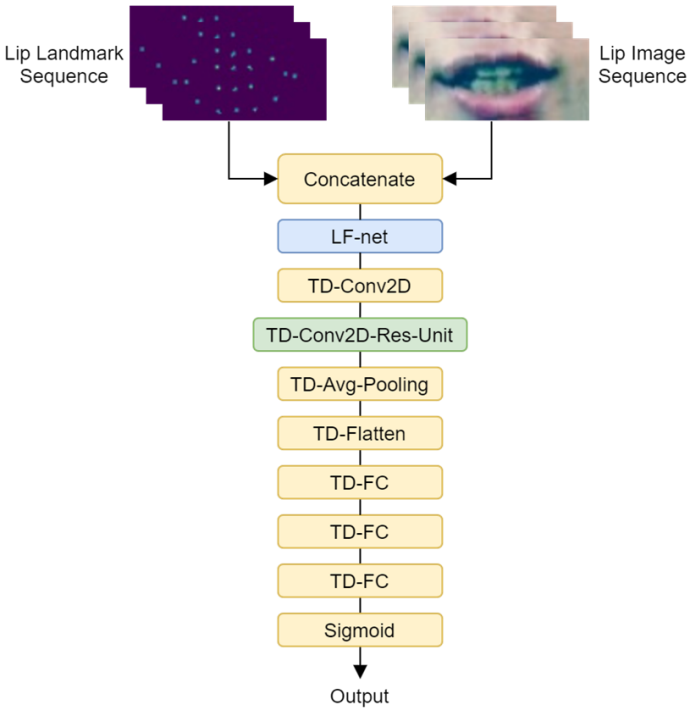


Figure 5: Neural Network for Merging First and Then Extracting Features

The second approach treats the keypoints as an additional channel and combines them with the RGB channels of the images before feeding everything into the base network, as shown in Figure 5. Detailed parameters are given in Table 7 of Appendix A.

3.2.1. LF-Net

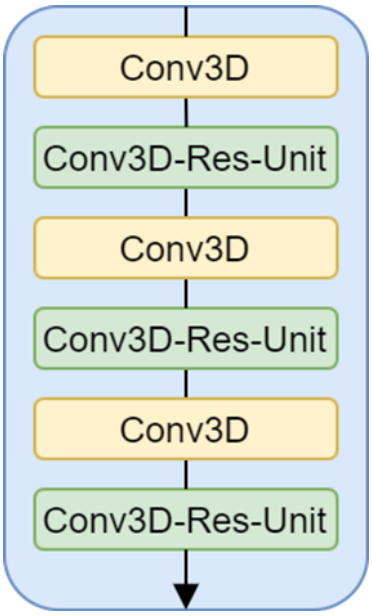


Figure 6: Lip Feature Network

LF-Net (Lip Feature Network) is the feature extraction network proposed in [15]. As illustrated in Figure 6, it primarily comprises three layers of 3D convolutions and 3D convolutional residual modules. After each 3D convolution layer comes a 3D convolutional residual module, and each of those modules contains two 3D convolution layers. Consequently, the entire LF-Net applies a total of nine 3D convolution layers to extract temporal and spatial features.

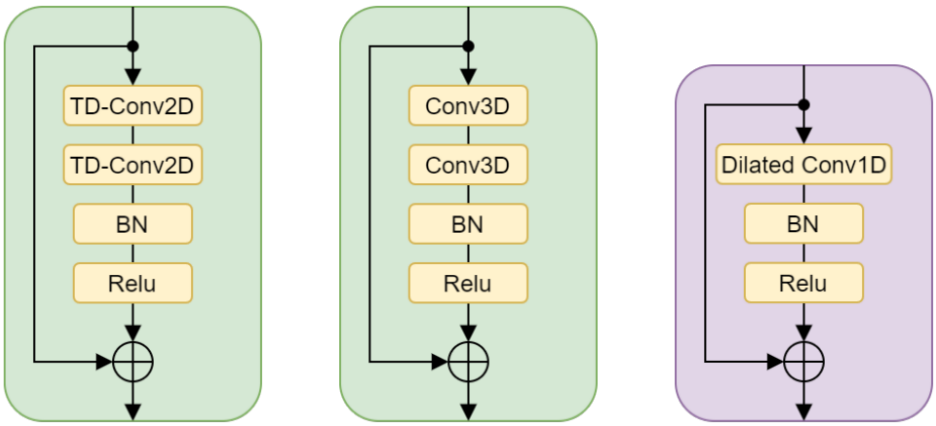
In LF-Net, the size of all convolution kernels is set to 3. Therefore, after passing through nine 3D convolution layers, the temporal receptive field of each output is 19. In other words, each output considers not only the current input but also the information from the preceding and subsequent nine frames. According to [15], with an FPS setting of 25 to 30, 19 frames roughly correspond to the time it takes a person to utter a single word. Since each output takes multiple time steps into account, the network can still perform well when speakers in the dataset speak at different speeds.

3.2.2. Time Distributed 2D Convolution Residual Unit

The TD-2D convolution residual module contains two layers of Time Distributed 2D convolutions, batch normalization, and a ReLU activation function. Its architecture is shown in Figure 7(a).

3.2.3. 3D Convolution Residual Unit

The 3D convolution residual module contains two layers of 3D convolutions, batch normalization, and a ReLU activation function, as illustrated in Figure 7(b).



(a) (b) (c)

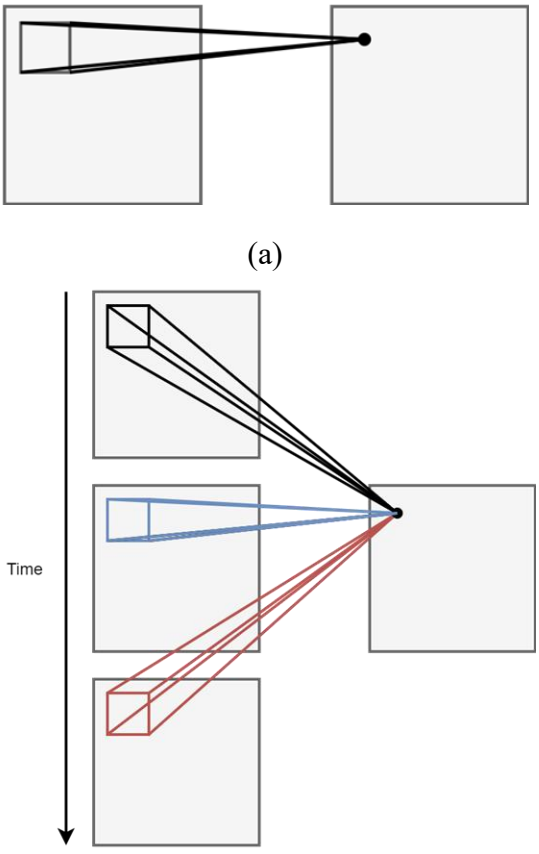
Figure 7: (a) TD-2D convolution residual module (b) 3D convolution residual module (c) 1D convolution residual module

3.2.4. 1D Convolution Residual Unit

The 1D convolution residual module includes one layer of 1D dilated convolution, batch normalization, and a ReLU activation function, as shown in Figure 7(c). This module performs convolution only along the time dimension of the input keypoint sequence and uses dilated convolution to increase the receptive field.

3.2.5. 3D Convolution Layer

A convolution layer refers to a neural network layer that uses filter weights in the convolution operation as its training targets. Typically, convolution layers applied to images are 2D convolutions, meaning they convolve over the height and width dimensions of the image, as shown in Figure 8(a). A 3D convolution layer, on the other hand, also takes the time dimension into account, as illustrated in Figure 8(b).



(a)

(b)

Figure 8: (a) 2D convolution (b) 3D convolution

The 3D convolution layer was first introduced in 3DCNN [16] for action recognition. It was later proven to be more effective at extracting spatial and temporal features of the lips in previous literature [15] and has also been widely applied in lip-reading tasks [17–19].

3.2.6. Residual Module

As artificial neural networks continue to evolve, researchers deepen networks to learn more expressive features. However, deep neural networks face two major challenges: the vanishing gradient problem and network degradation.

1) Vanishing Gradient Problem: When training neural networks using gradient descent, the weight update values are derived from backpropagation, which calculates partial derivatives of the loss function via the chain rule. As the network grows deeper, a large number of small values get multiplied repeatedly, causing the weight updates of the front layers to diminish, thereby reducing learning efficiency.

2) Network Degradation: This refers to the phenomenon where the accuracy of the model decreases as the network depth increases. Suppose there is an optimal 10-layer network architecture. When training with more than 10 layers, the extra layers must learn to map the input to the same output (an identity function). If the extra layers fail to learn the correct identity function, the performance may be worse than the original optimal architecture.

The concept of the residual block was first proposed in ResNet [20]. As shown in Figure 9, its core idea is to introduce skip connections between network layers, converting the multiplication in backpropagation into addition and thus preventing the vanishing gradient problem. Meanwhile, the network layer outputs $H(x) = F(x) + x$, so when the extra layers need to learn an identity function, their weight layers only need to learn $F(x) = 0$, which is easier than learning $F(x) = x$, thereby solving the network degradation issue.

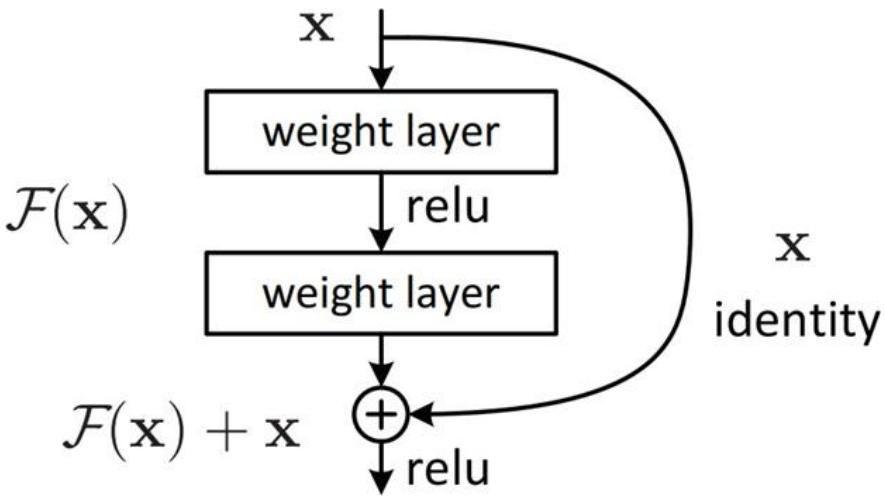


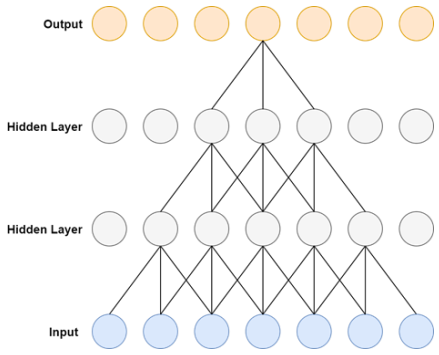
Figure 9: Residual Module

Source: [20]

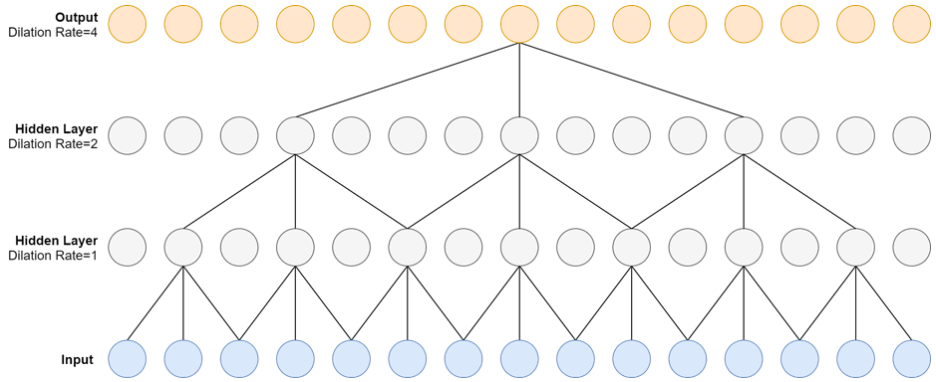
Residual modules are now widely used in deep learning applications across various fields. In addition to achieving better results in deep neural networks, they also improve the performance of shallower networks. Therefore, this thesis adopts residual modules to enhance network training.

3.2.7. Dilated Convolution

Dilated convolution was first applied in semantic segmentation [21]. By introducing dilation during the convolution operation, the receptive field can be expanded without increasing the number of parameters and computational cost, while avoiding the feature loss caused by downsampling. Figure 10 shows a comparison between ordinary convolution and dilated convolution with a kernel size of three for 1D convolution. It can be seen that at the same number of layers, dilated convolution effectively increases the receptive field.



(a)



(b)

Figure 10: (a) 1D Convolution (b) 1D Dilated Convolution

Because dilated convolution can increase the receptive field without relying on downsampling, it has also been widely used in tasks requiring sequential data, such as speech synthesis [22] and machine translation [23]. In this study, when extracting features from the lip keypoint sequences, dilated convolution was also employed to expand the receptive field in the time dimension under a smaller network depth, thereby ensuring that each output can consider a broader range of temporal information.

3.2.8. Frame Difference

Frame difference is calculated by taking the differences between adjacent frames in a sequence and forming a new sequence from these differences, thereby filtering out static information, as shown in Figure 11.

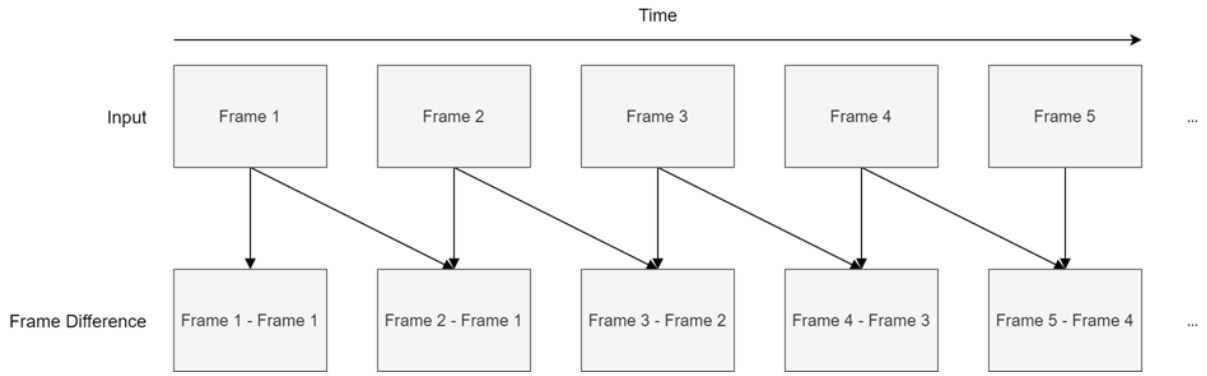


Figure 11: Frame Difference Method

In this study, to resist static sequence attacks, the network needs to learn more dynamic features. Hence, the classic frame difference method is chosen to remove static information from the sequence, allowing the model to distinguish between static and dynamic sequences.

3.2.9. GRU

A GRU (Gated Recurrent Unit) [24] is an improved version of LSTM (Long Short-Term Memory) [25], belonging to the family of recurrent neural networks (RNNs).

During the development of neural networks, researchers leveraged RNNs to handle temporal sequence data. RNNs can pass information from the current time step to the next, enabling the final output to take into account all previous time steps. However, in traditional RNNs, as the time sequence becomes longer, the influence of distant time steps on the weight updates gradually diminishes due to the vanishing gradient problem, resulting in poorer performance when it comes to long-term memory.

In 1997, Hochreiter and Schmidhuber proposed LSTM to solve the vanishing gradient issue. By adding a cell state in addition to the hidden state of the traditional RNN, LSTM can store long-term information and manage it through input, output, and forget gates. Its calculations are shown in Equations (1) to (5).

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$h_t = o_t \circ tanh(c_t) \tag{5}$$

Among these equations, x is the input, t denotes time, and i, f, o respectively represent the input, forget, and output gates. c and h denote the cell state and hidden state, while W, U, and b are the learnable weights and biases. $\circ$ refers to element-wise multiplication, and σ is the sigmoid function.

From Equations (1) to (5), one can see that the input gate and forget gate respectively control how much of the values computed from x_t and h_{t-1}, as well as the previous time's cell state, affect the current cell state. The output gate controls the extent to which the current cell state influences the hidden state. The values of the three gates are determined by the current input and the previous time's hidden state.

Although LSTM improves long-term memory capability via the cell state and three control gates, it also introduces more parameters and greater computational cost. To reduce computation time and memory usage, in 2014 Kyunghyun Cho et al. proposed GRU, which removes the cell state and simplifies the three control gates to an update gate and a reset gate. Its calculations are shown in Equations (6) to (9).

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{6}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{7}$$

$$h'_t = \tanh(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{8}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ h'_t \tag{9}$$

In these equations, x is the input, t is time, z and r are the update gate and reset gate, h is the hidden state, and h' is the candidate hidden state used to compute the final output hidden state h. W, U, and b are the learnable weights and biases, $\circ$ denotes element−wise multiplication, and σ is the sigmoid function.

From Equations (6) to (9), one can see that the reset gate controls the influence of the previous time's hidden state on the current candidate hidden state. The update gate controls how the previous time's hidden state and the current candidate hidden state together affect the final output hidden state. The values of these two gates are determined by the current input and the previous time's hidden state.

Although GRU does not perform as well as LSTM in terms of accuracy, it still achieves a similar level of performance and offers a clear advantage in run-time speed. Therefore, to test the effectiveness of an RNN on lip image sequences for identity authentication, this thesis adds two layers of bidirectional GRUs before the output layer of the network and conducts experiments using image sequences. The detailed parameters are shown in Table 5 of Appendix A, and the overall architecture is illustrated in Figure 12.

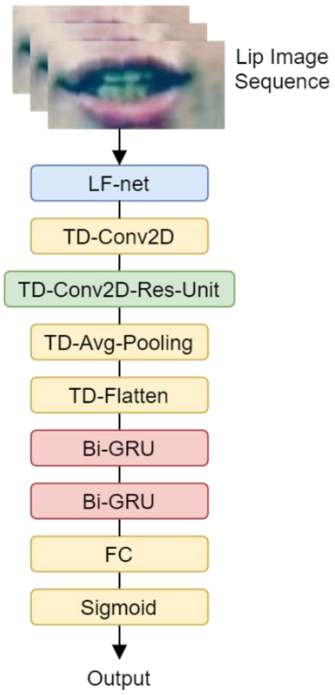


Figure 12: Network with Added GRU Layers

In the network shown in Figure 12, the final time-step output from the last GRU layer serves as the feature input to the fully connected layer. Since the GRU outputs in this paper are 512-dimensional vectors, we use a single fully connected layer at the end of the network to map these features into the desired one-dimensional output for authentication.

3.2.10. Binary Cross-Entropy

Binary cross-entropy is a commonly used loss function for binary classification problems. Since the model in this paper aims to determine whether the input corresponds to a legitimate user (a binary classification task), we employ this loss function for training, with its computation shown in Equation (10).

$$loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log\hat{y}_i + (1 - y_i) \cdot log(1 - \hat{y}_i) \quad (10)$$

Here, N is the number of outputs, y_i and $\hat{y}_i$ represent the target output and the predicted output, respectively. Because it uses a negative log for the loss calculation, the loss value grows exponentially larger when there is a greater discrepancy between the prediction and the target value.

3.3. Dataset

3.3.1. Data Collection

This paper collects a brand-new speaker video dataset consisting of nine male and six female subjects (a total of fifteen speakers). Each speaker provides twenty samples, with the specific procedures outlined below:

1) Environment

The data are captured against a white background, with adequate lighting, and recorded using a stable, non-shaking camera.

2) Recording Procedure

The speaker stands in front of the camera, faces it directly, and remains centered in the frame. The speaker is asked to say “57134” in English within the first three seconds after recording begins.

3) Recording Frequency

Multiple recording sessions are conducted, with at least one week between each session. Each speaker records five samples per session, for a total of four sessions, yielding twenty samples in total.

4) Format

Each sample is three seconds long, has a resolution of 1920×1080, 30 FPS, and uses the h.264 video codec.

The choice of “57134” as the password is based on the experimental results in [15], derived from a set of digit combinations that exhibit high discriminative power for authentication. Figure 13 shows an example from the dataset.

Figure 13: Sample from the Dataset

3.3.2. Preprocessing

The preprocessing in this paper mainly comprises three steps: detecting lip keypoints, cropping the lip region, and performing histogram matching. The following explains each step:

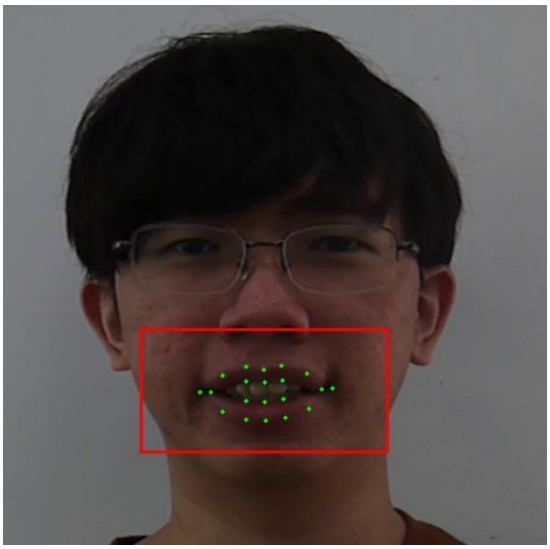


Figure 14: Detecting Lip Region and Keypoints

1) Detecting Lip Keypoints

First, we use the face detection method based on HOG and SVM from the dlib library to obtain the face region. This region is then passed to the facial landmark detection method, also in dlib and implemented via the ERT algorithm, to obtain 68 facial keypoints. Finally, we select the 20 keypoints corresponding to the lips, as indicated by the green dots in Figure 14.

2) Cropping the Lip Region

Taking the centroid of the lip keypoints as the center and using the horizontal distance between the two chin keypoints as the width, we crop out the lip region according to the target output ratio (shown by the red box in Figure 14), and then resize it to the target output dimensions. In this paper's experiments, the output image size is 50×100.

3) Histogram Matching

Histogram matching is a technique that transforms the histogram distribution of a target image to approximate that of a reference image. By matching all images in the dataset to the same reference image, we reduce lighting variations. In this paper, the reference image is chosen to be the one closest to the mean within the dataset.

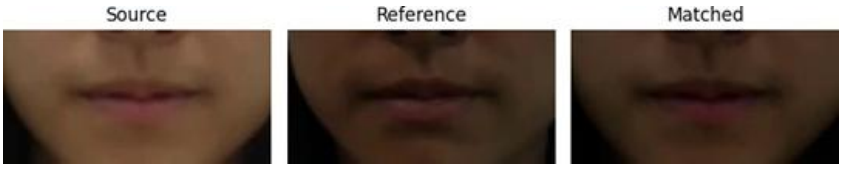


Figure 15: Input, Reference, and Output Images in Histogram Matching

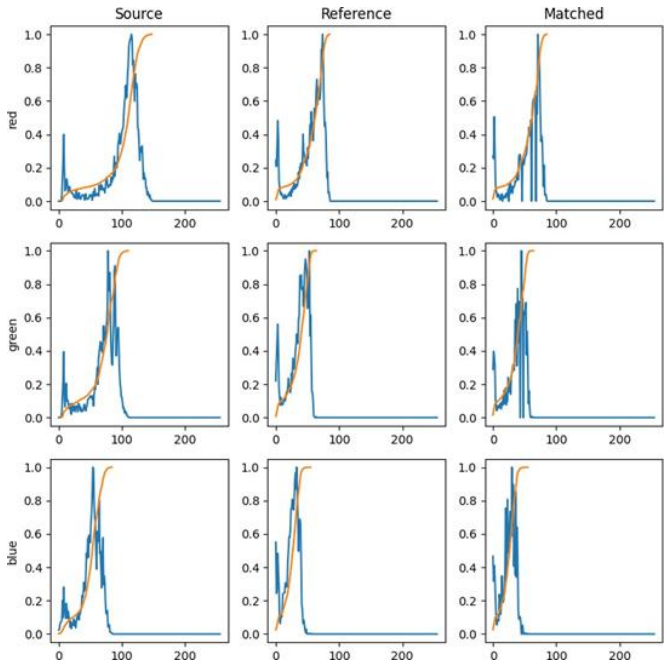


Figure 16: Histograms of Input, Reference, and Output Images in Histogram Matching

Figures 15 and 16 respectively show the original image, the reference image, and the image after matching, along with their corresponding histograms. It can be observed that the lighting in the original image is matched to resemble that of the reference image, and the RGB channels of the matched

image's histogram are each approximately distributed like those of the reference image. Although neural network methods tend to be relatively less sensitive to lighting variations, including histogram matching in preprocessing can still accelerate model convergence and thus shorten training time.

3.3.3. Data Augmentation

During the training phase of our experiments, data augmentation is used to expand the training samples and balance positive and negative samples. The image processing parameters are as follows:

1) Random Flip: Horizontal
2) Random Zoom: 1 to 1.1 times
3) Random Shift: -0.1 to 0.1 times the width and height
4) Random Rotation: -5 to 5 degrees

Figure 17: Image Sequence after Data Augmentation

All images in the same sequence receive the same set of random parameters. Figure 17 shows an example of an image sequence after data augmentation.

3.3.4. Frame Difference

To resist static sequence attacks, this paper applies frame difference in the preprocessing stage. The method calculates the pixel-wise difference between consecutive frames for each frame. All pixel values in the first frame are set to zero. After this processing, the images appear as shown in Figure 18. For the keypoint data, which is coordinate data, we compute the difference between the current frame's keypoint coordinates and those of the previous frame to obtain the motion vector.

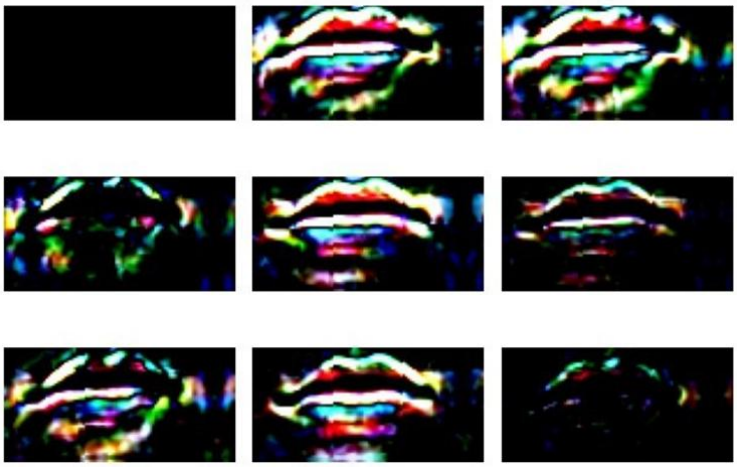


Figure 18: Image Sequence after Frame Difference Processing

3.4. Experiments

3.4.1. Experimental Procedure

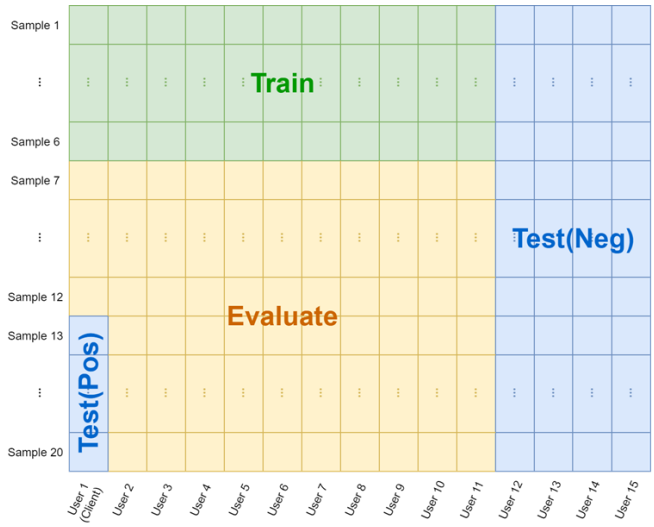


Figure 19: Distribution of Dataset Samples (Example when User 1 is the Client)

The experimental procedure in this paper is divided into three stages—training, evaluation, and testing. The number of samples used in each stage is allocated with reference to [26], as shown in Figure 19.

1) Training Stage

For each target user (Client), 30% of their own data and 30% of other users' data are extracted as positive and negative samples, respectively, for training. This produces the corresponding model.

2) Evaluation Stage

For each target user, 30% of their own data and 30% of other users' data (both distinct from the training set) are extracted as positive and negative samples, respectively, for testing. The EER threshold obtained here is used as the authentication threshold during testing.

3) Testing Stage

Each target user uses all remaining data as positive samples. Data from speakers not learned by the model are used as negative samples for testing. The threshold derived from the evaluation stage is employed as the authentication threshold. Finally, the average value of the test results across all user models is taken as the final result. To test resistance to static sequence attacks, replace the negative samples with static sequences of the positive samples.

3.4.2. Static Sequence Attack Experiment

A static sequence attack is easy to carry out. In this experiment, we assume a scenario where an attacker places a legitimate user's photograph in front of the camera, attempting to deceive the system with a static image sequence. Under normal circumstances, since the user in the static image did not speak the password, the authentication result should be a rejection. However, if the model makes a misjudgment, the outcome is a false acceptance. To test the system's resistance to this attack, this paper conducts an experiment using static sequence data. In this data, every frame is taken from the first frame of the original sequence, and all frames remain the same.

From the experimental results on static sequence attacks, we can not only observe the model's ability to resist this attack but also check whether the model has correctly learned the dynamic information in the sequence. This is one of the objectives of our experiment. If the model cannot withstand a static sequence attack, it implies that the model relies on static information in the sequence for its judgment. In that case, it would be no different from facial recognition or other static methods, failing to achieve the goal of enhancing security through sequential data.

3.4.3. Evaluation Metrics

Table 1: Classification of Authentication Results

		Predict	
		Positive	Negative
Ground Truth	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

This paper adopts common evaluation metrics for authentication tasks: FAR (False Acceptance Rate), FRR (False Rejection Rate), HTER (Half Total Error Rate), and EER (Equal Error Rate). This section introduces their calculation methods and meanings.

In the authentication problem, authentication outcomes can be divided into four categories, as shown in Table 1. True Positive (TP) represents correct acceptance (positive), meaning both the ground truth and prediction are "accept." False Positive (FP) represents false acceptance (a false positive), meaning the ground truth is "reject" but the prediction is "accept." True Negative (TN) represents correct rejection (negative), meaning both the ground truth and prediction are "reject." False Negative (FN) represents false rejection (a false negative), meaning the ground truth is "accept" but the prediction is "reject." The following metrics are calculated from these four outcomes:

1) FAR

False Acceptance Rate is the ratio of samples that should have been rejected at a certain threshold but were accepted, over all negative samples. Its calculation is shown in Equation (11).

$$FAR = \frac{FP}{TN + FP} \tag{11}$$

2) FRR

False Rejection Rate is the ratio of samples that should have been accepted at a certain threshold but were rejected, over all positive samples. Its calculation is shown in Equation (12).

$$FRR = \frac{FN}{TP + FN} \tag{12}$$

3) HTER

Half Total Error Rate is the average of FAR and FRR at a given threshold, representing the overall performance of the model. Its calculation is shown in Equation (13).

$$HTER = \frac{FAR + FRR}{2} \tag{13}$$

4) EER

As the threshold increases, FAR decreases and FRR increases. On the curve of FAR and FRR versus the threshold, there must be a point of intersection. Using the threshold at this point for authentication yields identical FAR and FRR. The error rate at this point is called the EER (as shown in Figure 20). EER can also be used to evaluate the overall performance of the model and is not influenced by the choice of threshold.

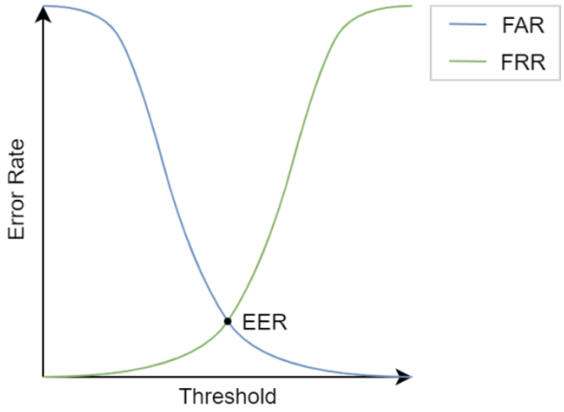


Figure 20: The Relationship among FAR, FRR, EER, and Threshold

3.4.4. Experimental Environment

In terms of hardware, this experiment uses an Intel(R) Core(TM) i7-8700 CPU and a GeForce GTX 1080 GPU.

On the software side, the experiments are run under the Ubuntu 18.04 operating system, and development is performed using the TensorFlow framework.

3.4.5. Network Training Parameters

The network training parameters used in this paper are shown in Table 2.

Table 2: Network Training Parameters

Parameter name	Value
Batch size	36
Epoch	25
Steps per Epoch	50
Optimizer	Adam
Learning rate	0.0001

4. Experimental Results and Discussion

To compare the effectiveness of image data, keypoint data, and frame difference, this paper conducts experiments using several methods, which include:

1) rgb: Using the lip image sequence as input
2) lm: Using the lip keypoint sequence as input
3) rgb-lm: Using both the lip image sequence and the lip keypoint sequence as input
4) rgb-fd: Using frame difference of the lip image sequence as input
5) lm-fd: Using frame difference of the lip keypoint sequence as input
6) rgb-lm-fd: Using frame difference of both the lip image sequence and the lip keypoint sequence as input

Table 3: Authentication Experiment Results Under the Post-Merge Neural Network (Shown in Percentages)

Method	*FAR*	*FRR*	*HTER*	*EER*	FAR_S	$HTER_S$	EER_S
rgb	0.80	18.18	9.49	1.66	82.95	50.57	54.92
lm	5.80	**10.23**	8.01	6.79	89.77	50.00	50.00
rgb-lm	0.68	17.05	8.86	1.02	84.09	50.57	54.55
rgb-fd	**0.57**	12.50	6.53	0.68	**9.09**	**10.80**	**9.09**
lm-fd	14.09	15.91	15.00	14.00	84.09	50.00	50.00
rgb-lm-fd	0.68	**10.23**	**5.45**	**0.57**	22.72	16.48	12.50

Table 3 shows the experimental results of each method under the post-merge neural network (Figure 4). In this section, an analysis and discussion of these results is presented. Note that FAR_S, $HTER_S$, and EER_S indicate the performance of each respective metric under static sequence attacks.

4.1. Lip Images and Keypoints

In Table 3, the methods rgb and lm compare the effectiveness of lip images and keypoints in the authentication problem. From the HTER, there is not much difference in the results whether using lip images or keypoint sequences as input. However, if we look at the EER, the image data clearly outperform the keypoint data. This discrepancy is attributed to the varying amount of feature information contained in each type of data; image data include significantly more comprehensive features than the coordinate data of keypoints.

4.2. Frame Difference for Resisting Static Sequence Attacks

In Table 3, the methods rgb, lm, rgb-fd, and lm-fd compare the ability of direct input (original image and keypoint sequences) and frame-difference-processed sequences to resist static sequence attacks. From the FAR_S values of rgb and lm in Table 3, it can be seen that, whether using images or keypoints, direct input yields extremely low resistance to static sequence attacks. Based on these experimental results, we can infer that even if the neural network design takes temporal relationships into account, during actual training the model still tends to learn spatial features, thus failing to achieve the goal of enhancing security with sequential features.

Conversely, comparing the FAR_S and FRR of rgb and rgb-fd in Table 3 reveals that, after frame difference processing, the lip image sequence data not only show a 73.86% increase in resistance to static sequences but also achieve a 5.68% improvement in error rate under normal authentication scenarios. Evidently, frame-difference-processed image data can enhance resistance to static sequence attacks while maintaining authentication performance, a finding also reflected in the comparison of EER and EER_S.

Although frame difference exhibits a favorable performance on image data, comparing the lm and lm-fd methods in Table 3 shows that frame-difference-processed keypoint data not only fail to improve resistance to static sequence attacks but also perform worse in normal authentication scenarios. This may be because lip keypoints show relatively small changes while speaking, so after applying frame difference to the keypoint sequence, the location information is lost, and the remaining dynamic information is insufficient for identity authentication or distinguishing static sequence attacks.

4.3. Merging Lip Images and Keypoints

In Table 3, the rgb-lm method shows the result of merging image and keypoint sequence data. Compared with methods that do not merge the two, it can be seen that combining the data can achieve better results in normal authentication scenarios. For instance, regarding HTER, merging both features yields a 0.63% decrease compared to using only image features. In terms of EER, there is also a 0.64% improvement. Hence, keypoint features can serve as an auxiliary to image features in authentication tasks, enhancing performance. However, observing FAR_S shows that, without frame difference processing, the defense against static sequence attacks remains insufficient.

Looking at the EER and EER_S of the rgb-lm-fd method in Table 3, although there is a slight improvement in authentication performance compared to rgb-fd (which uses only images) under normal authentication scenarios, resistance to static sequence attacks is diminished. This result may be due to the weaker resistance of keypoint data to static sequence attacks, which affects the overall performance of the model. Since rgb-lm-fd performs worse under static sequence attacks, this paper concludes that rgb-fd, i.e., using frame-difference-processed lip images, achieves the best results under the neural network architecture used here.

Table 4: Authentication Experiment Results Under the Pre-Merge Neural Network (Shown in Percentages)

Method	*FAR*	*FRR*	*HTER*	*EER*	FAR_S	$HTER_S$	EER_S
rgb-lm	3.07	15.91	9.49	2.39	81.82	48.86	52.84
rgb-lm-fd	0.57	3.41	1.99	0.73	32.95	18.18	14.77

To find a better method of merging features, this paper also conducts an experiment using the pre-merge network architecture (Figure 5), in which keypoint information is treated as one channel and merged with the RGB channels of the lip images. The results of this experiment are shown in Table 4. Comparing these with Table 3, both methods experience a decline in EER, and looking at EER_S, it can be seen that the rgb-lm-fd method also has a higher error rate. Thus, the post-merge method (Figure 4) for combining lip images and keypoints is more effective for the authentication problem.

4.4. GRU

To test the capability of GRU in extracting dynamic features from the lips, this paper adds two layers of bidirectional GRUs before the output layer of the network in Figure 4, using the image sequence for experiments. The results are shown in Table 5.

Table 5: Authentication Experiment Results After Adding GRU Layers (Shown in Percentages)

Method	*FAR*	*FRR*	*HTER*	*EER*	FAR_S	$HTER_S$	EER_S
rgb	3.41	9.09	6.25	0.80	85.23	47.16	47.2

Comparing the EER_S in Table 3 and Table 5 shows that, although there is a slight improvement in resistance to static sequence attacks, the effect is still inferior to that of frame-difference-processed image sequence data.

5. Conclusion

This paper collects a new fixed-password speaker dataset and conducts experiments on a lip-image-sequence-based identity authentication problem using the proposed artificial neural network, which can merge image and keypoint sequences. The experimental results show that lip image sequence features, when used alone, outperform lip keypoint sequence features in discriminative power. Nevertheless, when the two are merged, the performance is better than using only image sequence features. Specifically, the results achieve an FAR of 0.68%, an FRR of 17.05%, and an EER of 1.02%, indicating that keypoint features can assist image sequence features for authentication.

However, from the static sequence attack experiments, it can be seen that directly training on lip images or keypoint sequences yields very poor resistance to static sequence attacks. This finding suggests that if sequential data is used directly for training, the network will tend to learn spatial features rather than temporal relationships; therefore, the goal of enhancing security with sequential data is not achieved. To address this problem, this paper proposes using frame-difference-processed input sequences. The experimental results show that the frame-difference-processed lip image sequence achieves the best outcomes. Under attacks, its FAR_S is 9.09%, a 73.86% improvement compared to unprocessed data, confirming that this method can effectively enhance resistance to static attacks.

In the static sequence attack experiments, this paper processes the data using the frame difference method rather than adding attack data into the training set. This approach enables the network to learn dynamic features to guard against more types of attacks, not just static sequences, while also avoiding increased training costs. However, because other forms of attack data were not collected, only the easiest-to-obtain static sequences were tested. In the future, gathering different types of attack data (e.g., video of the same speaker saying different sentences, or synthesized video of a target speaker saying the password) to verify the ability of sequential data in a neural network to counter attacks is an important goal.

The proposed method exhibits broad application potential, enhancing security and accuracy in financial transactions, smartphone unlocking, and human-computer interaction. Moreover, when integrated with existing facial or voice recognition systems, it can further improve the reliability and adaptability of multimodal biometric authentication technologies.

References

[1] P. Jourlin, J. Luettin, D. Genoud, and H. Wassner, "Acoustic-labial speaker verification," *Pattern Recognition Letters*, vol. 18, no. 9, pp. 853–858, 1997.

[2] T. Wark, D. Thambiratnam, and S. Sridharan, "Person authentication using lip information," in *TENCON '97 Brisbane - Australia. Proceedings of IEEE TENCON '97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications* (Cat. No.97CH36162), vol. 1, pp. 153–156 vol. 1, 1997.

[3] R. Frischholz and U. Dieckmann, "BiolD: a multimodal biometric identification system," *Computer*, vol. 33, no. 2, pp. 64–68, 2000.

[4] C. C. Broun, X. Zhang, R. M. Mersereau, and M. Clements, "Automatic speechreading with application to speaker verification," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I–685–I–688, 2002.

[5] H. Cetingul, Y. Yemez, E. Erzin, and A. Tekalp, "Discriminative analysis of lip motion features for speaker identification and speech-reading," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2879–2891, 2006.

[6] U. Sanchez and J. Kittler, "Fusion of talking face biometric modalities for personal identity verification," in 2006 *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, pp. 1073–1076, 2006.

[7] M. Faraj and J. Bigun, "Motion features from lip movement for person authentication," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 1059–1062, 2006.

[8] S. A. Samad, D. A. Ramli, and A. Hussain, "Lower face verification centered on lips using correlation filters," *Information Technology Journal,* vol. 6, no. 8, pp. 1146–1151, 2007.

[9] Y.-F. Liu, C.-Y. Lin, and J.-M. Guo, "Impact of the lips for biometrics," *IEEE transactions on image processing*, vol. 21, pp. 3092–3101, 06 2012.

[10] C. H. Chan, B. Goswami, J. Kittler, and W. Christmas, "Local ordinal contrast pattern histograms for spatiotemporal, lip-based speaker authentication," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 602–612, 2012.

[11] S.-L. Wang and A. W.-C. Liew, "Physiological and behavioral lip biometrics: A comprehensive study of their discriminative power," *Pattern Recognition*, vol. 45, no. 9, pp. 3328– 3335, 2012. Best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).

[12] X. Liu and Y.-m. Cheung, "Learning multi-boosted HMMs for lip-password based speaker verification," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 233–246, 2014.

[13] J.-Y. Lai, S.-L. Wang, A. W.-C. Liew, and X.-J. Shi, "Visual speaker identification and authentication by joint spatiotemporal sparse coding and hierarchical pooling," *Information*

Sciences, vol. 373, pp. 219–232, 2016.

[14] X.-X. Shi, S.-L. Wang, and J.-Y. Lai, "Visual speaker authentication by ensemble learning over static and dynamic lip details," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3942–3946, 2016.

[15] F. Cheng, S.-L. Wang, and A. W.-C. Liew, "Visual speaker authentication with random prompt texts by a dual-task CNN framework," *Pattern Recognition*, vol. 83, pp. 340–352, 2018.

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[17] T. Stafylakis and G. Tzimiropoulos, "Combining residual networks with LSTMs for lipreading," *CoRR*, vol. abs/1703.04105, 2017.

[18] T. Afouras, J. S. Chung, A. W. Senior, O. Vinyals, and A. Zisserman, "Deep audio-visual speech recognition," *CoRR*, vol. abs/1809.02108, 2018.

[19] S. Petridis, T. Stafylakis, P. Ma, F. Cai, G. Tzimiropoulos, and M. Pantic, "End-to-end audiovisual speech recognition," *CoRR*, vol. abs/1802.06424, 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[22] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalch- brenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," arXiv:1609.03499, 2016.

[23] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," arXiv:1610.10099, 2017.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] J. Luettin and G. Maître, "Evaluation protocol for the extended M2VTS database (XM2VTSDB)," *IDIAP*, Tech. Rep. Idiap-Com-05-1998, 1998.

Appendix A

Tables 6 and 7 list the architectures and outputs of the post-merge and pre-merge networks, respectively (Figures 4 and 5). The outputs and detailed parameters of the network layers rgb_net, landmark_net, and output_net are shown in Tables 8 to 10.

Table 6: Architecture and Outputs of the Post-Merge Network (Figure 4)

Layer name	Output shape	Connected to
rgb_seq (Input)	$T \times 50 \times 100 \times 3$	-
landmark_seq (Input)	$T \times 20 \times 2$	-
rgb_net	$T \times 1024$	rgb_seq (Input)
landmark_net	$T \times 1024$	landmark_seq (Input)
Concatenate	$T \times 2048$	rgb_net, landmark_net
output_net	$T \times 1$	Concatenate

Table 7: Architecture and Outputs of the Pre-Merge Network (Figure 5)

Layer name	Output shape	Connected to
rgb_seq (Input)	$T \times 50 \times 100 \times 3$	-
landmark_seq (Input)	$T \times 50 \times 100 \times 1$	-
Concatenate	$T \times 50 \times 100 \times 4$	rgb_seq (Input), landmark_seq (Input)
rgb_net	$T \times 1024$	Concatenate
output_net	$T \times 1$	rgb_net

Table 8: Outputs and Detailed Parameters of rgb_net

Layer name	Output shape	Kernel/Stride/Pad/No. of Filters (or Parameter)
rgb_seq (Input)	$T \times 50 \times 100 \times 3$	- / - / - / -
conv3D_1	$T \times 25 \times 50 \times 32$	3 × 5 × 5/ 1, 2, 2/ 1, 2, 2/ 32
3D-Res-block_1	$T \times 25 \times 50 \times 32$	- / - / - / 32
conv3D_2	$T \times 13 \times 25 \times 64$	3 × 3 × 3/ 1, 2, 2/ 1, 1, 1/ 64
3D-Res-block_2	$T \times 13 \times 25 \times 64$	- / - / - /32
conv3D_3	$T \times 7 \times 13 \times 96$	3 × 3 × 3/ 1, 2, 2/ 1, 1, 1/ 96
3D-Res-block_3	$T \times 7 \times 13 \times 96$	- / - / - /32
TD-conv2D_1	$T \times 4 \times 7 \times 128$	3 × 3/ 2, 2/ 1, 1/ 128
TD-2D-Res-block_1	$T \times 4 \times 7 \times 256$	- / - / - / 256
TD-Avg-Pool_1	$T \times 2 \times 2 \times 256$	(2, 3)/ - / - / -
TD-Flatten	$T \times 1024$	- / - / - / -

Table 9: Outputs and Detailed Parameters of landmark_net

Layer name	Output shape	No. of Filters/Dilation rate
landmark_seq (Input)	$T \times 20 \times 2$	- / -
TD-Flatten	$T \times 40$	- / -
1D-Res-block_1	$T \times 128$	128/ 1
1D-Res-block_2	$T \times 256$	256/ 2
1D-Res-block_3	$T \times 512$	512/ 4
1D-Res-block_4	$T \times 1024$	1024/ 8

Table 10: Outputs and Detailed Parameters of output_net

Layer name	Output shape	Kernel/Stride/Pad/No. of Filters (or Parameter)
TD-FC_1	$T \times 1024$	1024
TD-FC_2	$T \times 512$	512
Dropout	$T \times 512$	p=0.5
TD-FC_3	$T \times 1$	1

Table 11: Outputs and Detailed Parameters of the Network with GRU Layers

Layer name	Output shape	Parameter
rgb_seq (Input)	$T \times 50 \times 100 \times 3$	-
rgb_net	$T \times 1024$	-
Bi-GRU_1	$T \times 1024$	512
Bi-GRU_2	1024	512
FC_3	1	1