Bi-directional Long-term Recurrent Convolutional Network: Recognizing Stress in Thermal Videos*

Siyuan Yan

Research School of Computer Science
Australian National University
ACT 2601 AUSTRALIA
u7050317@anu.edu.au

Abstract. Stress is a serious concern in our daily life and can affect people's health. So, it is important to develop a model that can recognize people's stress conditions. Contact-free stress recognition methods like RGB or thermal images based methods have been widely explored. However, researchers find the visual perceptual features in static images are very limited. In this paper, we first develop a balanced Bidirectional Neural Network(BDNN) to recognize people's stress conditions and study whether Bi-directional training (BDT) can achieve better generalization ability by using a compressed dataset. We also explore the bimodal method that utilizes feature-level fusion from both RGB and thermal images. Then, we propose a Bi-directional Long-term Recurrent Convolutional Network(BD-LRCN) that can automatically recognize stress from thermal videos by joint learning visual perceptual and time-sequential representations. Moreover, to make our model can learn both forward and backward information from video clips, we use the BDT scheme to better learn temporal information. Experiments show that our BD-LRCN can achieve 68.4% classification accuracy on the ANUStressDB dataset.

Keywords: Stress Recognition · Bimodal · Thermal Video · Long-term Recurrent Convolutional Network · Bi-directional training.

1 Introduction

Stress is common in our daily life and it can cause negative emotions like anger, anxiety, and fear. If stress stays for a long time, it can cause cardiovascular diseases[12] and cancer[19]. Recognizing stress is a hot topic in research and has attracted many researchers to work on it. Many researchers used physiological signals like heart rate[18] and skin temperature[20] to recognize stress, while these methods require sensors to be in touch with the body of people and are time-consuming. A more reasonable way to recognize stress is to use facial data like thermal and RGB data, which can be simply captured by cameras. Sharma et al [17] used a feature-level fusion of RGB and thermal images to recognize stress. Irani et al [5] proposed a method that extracts the features from the super-pixel of facial images, achieving a better performance. To exploit both spatial and temporal features, Kumar et al [10] developed a Spatio-temporal network to predict ISTI signals from thermal videos and then feed them into a classifier. This method achieves better performance than the static images-based method but the limitation of this method is the pipeline is too complex and is not an end-to-end model.

Traditional Recurrent Neural Networks(RNN) try to simulate the human brain and it can learn information through time by back-propagation training[15]. One problem is when we use the RNN to learn temporal information from thermal videos, it only relies on past and current information, which fails to model global information. By a bidirectional learning scheme, RNN can learn information from both past and future sequences, which achieves a higher generalization ability to do classification. Inspired by the human brain, bidirectional associative memories[7, 8] have been developed to perform bidirectional learning. But they both have low capacities and inconsistent learning during different layers. Bidirectional Neural Network(BDNN)[13] avoids these problems and it can perform back-propagation in both forward and reverse direction. It has been proved that BDNN can remember input patterns as well as output classes, given either of them. Inspired by BDNN, we adapt its Bi-directional training scheme into the Long Short-Term Memory (LSTM) unit. LSTM is a recurrent module that can achieve long-range learning and avoid gradient vanishing by gating mechanism, we will use it to learn temporal information. To get the spatial feature of each time step needed for LSTM, we choose to use the Convolutional Neural Network (CNN) [9] as our feature extractor to get the spatial features.

In this paper, we first fine-tune a one-layer NN model using different hyper-parameters on the compressed ANUStressDB dataset. Then, we compare the BDNN model with the baseline using the same setting. After that, we implement a Bi-directional Long-term Recurrent Convolutional Network(BD-LRCN) on the video version ANUStressDB dataset. The BD-LRCN consists of stacked CNNs to extract spatial features for each time step and Bi-LSTMs to learn temporal information.

* Supported by Australian National University.

We summarize our contribution as: (1) We use a BDNN to learn the bidirectional mapping between PCA features and classes and show Bi-directional training can achieve better generalization ability. (2) We utilize the feature-level fusion from both RGB and thermal images to exploit complementary information between the two modalities. (3) We propose a Bi-directional Long-term Recurrent Convolutional Network that can be end-to-end trainable on thermal videos. (4) A Bi-directional LSTM module to learn not only forward information but also backward information from time sequences. (5) Extensive experiments demonstrate that our proposed Bi-LRCN can significantly improve performance than traditional sequence modeling-based models on video datasets.

2 Method

2.1 Dataset Description

Dataset In this paper, we use the ANUStressDB dataset[5]. It involves 31 subjects and each subject is asked to watch 20 different films. Then, their facial data are collected by RGB camera and thermal camera. So, the dataset contains 620 patterns. For the data v1, each pattern consists of 10 PCA features from RGB and thermal videos, each pattern is labeled as "Stressful" or "Calm". To further analyze the data v1, we use principal component analysis (PCA) to visualize the data. We can find the data is highly overlapping and hard to be classified. Also, features are on different scales. For data v2, it is the complete video dataset of data v1. The dataset contains 31 videos for 31 subjects and each video is about 32 mins. Each video can be divided into 20 "film reactions" each 1.5 mins long. There is a 5-8 s set up at the beginning and about 6 s gap between every 1.5 mins "film reaction". In the remaining part of the paper, we will use "video" denotes 32 videos, and use "film" denotes the divided 1.5 mins "film reaction".

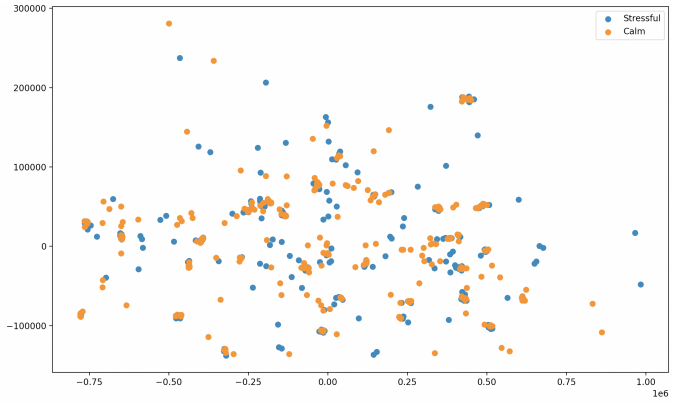

Fig. 1. Visualization of dataset v1 using PCA.

Data pre-processing For data v1, we remove "ID Subject" as it should not be considered when performing stress recognition. Followed by most classification tasks, we treat the label "Stressful" as 1 and label "Calm" as 0. By seeing Figure 1, the remaining 10 PCA features are on different scales, it is necessary to perform feature scaling so that no features dominate during training. We choose standard techniques in Equation 1 to our 10 features.

$$X' = \frac{X - \mu}{\sigma} \tag{1}$$

where μ is the mean of feature values and σ is the standard deviation of the feature values.

For data v2, we transform the 31 videos into 620 folders and each folder contains 28 extracted frames from one film. The video processing and frame extraction method are described in Figure 2. For each video, we divide it into 20 same-length films. As the change of face in videos is slow, we only sample 36 frames for each film. To avoid the wrong segmentation between adjacent films and neutralize the subjects' emotions before playing the next film, we discard the first 4 frames and the last 4 frames of each film. We finally get 28 frames for each film. Also, before feeding frames into our model, we also standard all frames.

2.2 Baseline Method

In our baseline model, we use a fully connected neural network with only one hidden layer due to the model has got a very high accuracy in the training dataset. As our stress recognition is a binary classification task, we use binary cross-entropy as our loss function, shown in Equation 2.

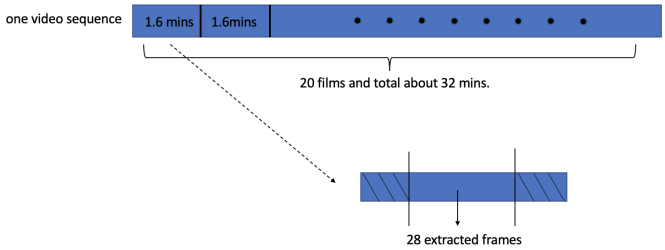

Fig. 2. video processing and frame extraction.

$$\mathcal{L}_{bce} = -\frac{1}{N}\sum_{i=1}^{N} t_i log(y_i) + (1 - t_i log(1 - y_i)) \tag{2}$$

where N is the dataset size, t_i is the ith ground-truth label, and y_i is the ith predicted value.

For our model, the input features are from RGB and thermal images. The output is one neural that can produce "Stressful" or "Calm" using the Sigmoid function. In Section 3, we will investigate the result with different numbers of hidden neurons, optimizers, activation functions, and modalities.

2.3 Bidirectional Neural Network

Architecture Different from our baseline network, our BDNN does not only require recognizing the stress by input patterns but also needs to produce the input pattern by classes. Thus, the relation between input and output should be invertible. To achieve it, our BDNN should be a one-to-one mapping function. However, stress recognition is a many-to-one problem, which means BDNN cannot directly be used on it. Inspired by [13], we add an extra node on the output of BDNN. For each sample, we add the mean value to the ground truth for each pattern, so that can perform the one-to-one mapping. Our BDNN follows a similar architecture as the baseline, the difference is the output has two neurons. One neuron is used to perform classification and the second neuron is used to perform regression with the mean value of the input pattern. Also, to simplify the training, all biases are removed and we find it does not affect the performance. The architecture of our BDNN is shown in Fig. 3.

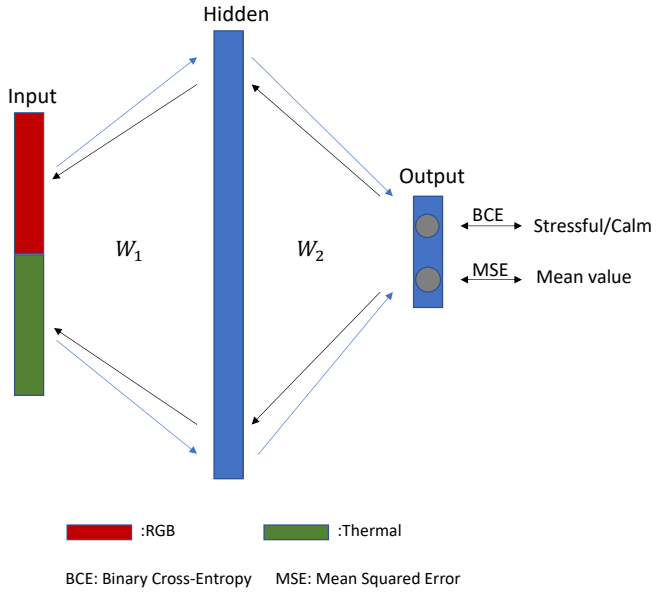

Fig. 3. Illustration of our BDNN model. We follow a similar architecture as the baseline, the difference is the output of our BDNN contains an extra neuron, performing regression with an extra value (mean value).

Training BDNN We applied back-propagation in both forward and reverse directions. For the first 50 epochs, the training is in the forward direction, two kinds of loss functions are used. Binary Cross-Entropy(BCE) is used to classify whether a person is stressed and Mean Squared Error(MSE) is used to do regression between the extra neuron and the mean value of the pattern. The training direction is reversed if maximum epochs are reached or BCE error is lower than a tolerance number. For the reverse training, the two output values are combined as the input vector, the reverse direction tries to predict the input patterns using MSE loss.

The total loss in the forward direction can be summarized in Equation 3.

$$\begin{aligned} \mathcal{L}_{bce} &= -\frac{1}{N}\sum_{i=1}^{N} t_i log(y_i) + (1 - t_i log(1 - y_i)), \\ \mathcal{L}_{mse} &= \frac{1}{N}\sum_{i=1}^{N}(E_i - \hat{E}_i)^2, \\ \mathcal{L}_{total} &= \mathcal{L}_{bce} + \mathcal{L}_{mse}. \end{aligned} \tag{3}$$

where N is the dataset size, t_i is the ith ground truth label and y_i is the ith predicted value, E_i is the ith extra value, and $\hat{E}_i$ is the ith predicted the mean value. The two kinds of loss functions in the forward direction can be trained together.

The loss in the reverse direction is shown in Equation 4.

$$\mathcal{L}_{reversed} = \mathcal{L}_{mse} = \frac{1}{N}\sum_{i=1}^{N}(X_i - \hat{X}_i)^2 \tag{4}$$

where X_i is the ith input pattern and $\hat{X}_i$ is the ith predicted input pattern.

Training BDNN with a balance term BDNN in the original paper train in 50 epochs in each direction, we argue that the same training epochs in two directions will affect the prediction performance of each other. As the class prototypes depend on the classification accuracy, the network should pay more attention to the classification task. We introduce a simple balance term between the two-direction training, making the training be more stable. The Formula is summarized as Equation 5.

$$E_{reversed} = \beta E_{forward}. \tag{5}$$

where $E_{reversed}$ is the number of reverse training epochs, $E_{forward}$ is the number of forward training epochs, and β is the balance term, we set it as 0.5.

2.4 Bi-directional Long-term Recurrent Convolutional Network

To recognize stress in videos, inspired by LSTM-based architecture [2] in the Action recognition field, we propose the Bi-directional Long-term Recurrent Convolutional Network (Bi-LRCN) combining a spatial feature extractor with a sequential model that can model long-range temporal feature, the architecture is shown in Figure 4. Let us first define each video: D_v=$\{x_i\}_i^T \rightarrow y$, where x_i is the input frame in the ith time step and y is the label for the video D_v. We feed frames $x_1, ..., x_T$ into parallel spatial CNNs $F_1, ..., F_T$ to extract fixed-length spatial features $F_1(x_1), ..., F_T(x_T)$, then we feed them into time-sequential modeling module (Bi-LSTM), the results are $y_i, ..., y_T$. Finally, we use two fully connected layers to get the final classification results of the video. Notice that our spatial CNNs can be trained parallel over time steps and thus save a lot of computation time. Also, our Bi-LCRN model is end-end trainable, which is simpler than previous work [10]. Next, we will go into detail about the two most important components of our model, which are spatial CNN and Bi-directional LSTM.

Spatial CNN In general, a deeper neural network trends to achieve better performance. However, training a very deep neural network is very difficult due to gradient vanishing and exploding. Residual Network(ResNet) [10] tries to solve the gradient vanishing and exploding problem in deep neural networks. It achieves very high performance in a lot of tasks. Our spatial CNN starts with a pre-trained ResNet followed by Batch Normalization, Relu, and Fully connected layer. The skip-connection in Resnet can perform identity mapping, so that avoid gradient vanishing, the operation is shown in Equation 6.

$$x_{l+1} = x_l + \mathcal{F}(x_l, w_l) \tag{6}$$

where $\mathcal{F}$ denotes a transformation, x_l is the input before the transformation and x_{l+1} is the result after the transformation, and w_l is the weights of the transformation.

The Batch Normalization can make the network learns the optimal mean and scale of input for each layer, speeding up and stabilizing the training. It also can perform regularization, reducing the overfitting of the very deep network.

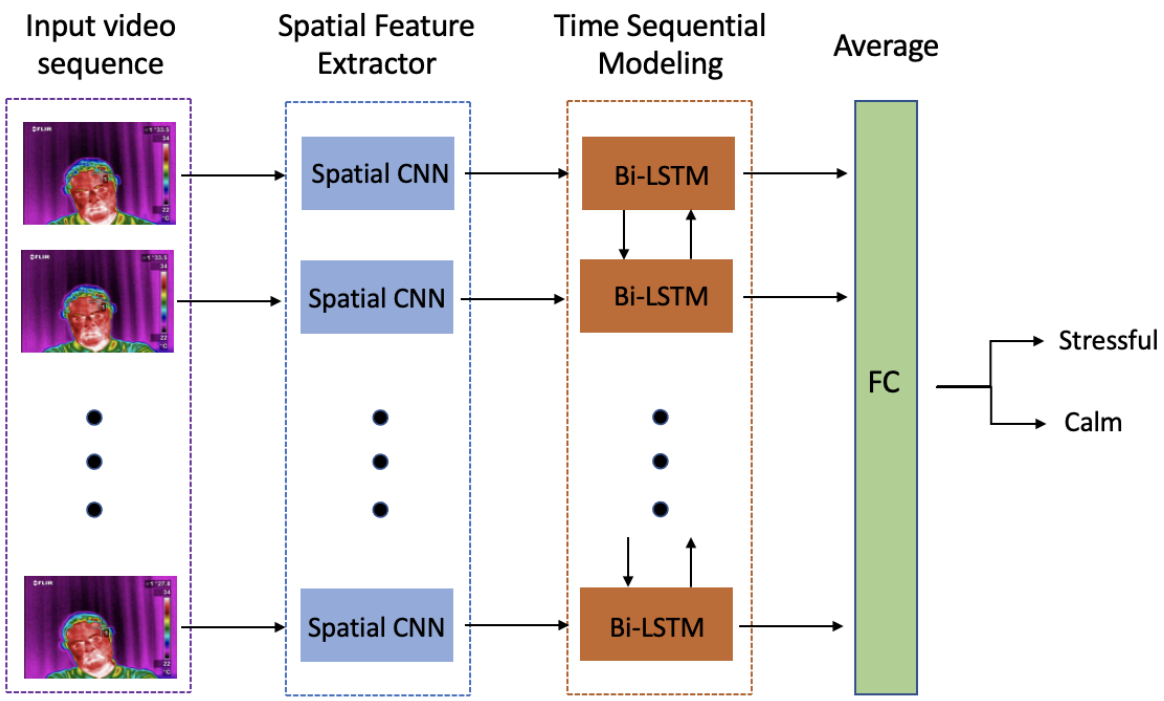

Fig. 4. Illustration of our BD-LRCN model. The model uses parallel spatial CNNs to extract fix-length visual features and then feed them into stacked sequence models Bi-LSTMs. Finally, the results from stacked sequence models are fused and get the final classification result.

Bi-LSTM module Long Short-Term Memory (LSTM) is a module that can capture long-range dependencies of sequence and avoid gradient vanishing. A simple example of the LSTM is shown in Figure 5a where x_1, x_2, and x_3 denote the input feature sequences, h denotes the hidden neuron, and $\hat{y}$ denotes the output sequence. LSTM is especially useful on our video classification task as it has a very powerful memory ability for long sequences, which is achieved by its gating mechanism that consists of an input gate, a forget gate, and an output gate. As stress is not only related to previous video frames but also next video frames, to further consider the forward and backward direction information in video sequences, we extend the LSTM into Bi-LSTM, which is shown in Figure 5b where the blue arrow denotes forward pass, green arrow denotes backward pass, f_i denotes forward hidden state, and b_i denotes backward hidden state.

Training Bi-LRCN Our network only needs a Binary cross-entropy loss, which is the same as Equation 2. To train the network, for spatial CNNs part, it can be trained by back-propagation. For Bi-LSTMs part, it is a Bi-directional training, we follow the training method in Bi-directional RNN[16]. The training process can be summarized as 3 steps:

1. *FORWARD PASS* :
   (a) Do forward pass for forward hidden states $f_1, ...f_T$
   (b) Do forwad pass for backward hidden states $b_T, ..., b_1$.
   (c) Do forward pass for all outputs $\hat{y}_1, ...\hat{y}_T$.
2. *BACKWARD PASS*
   (a) Do backward pass for for all outputs $\hat{y}_1, ...\hat{y}_T$.
   (b) Do backward pass for forward hidden states $f_T, ...f_1$
   (c) Do backward pass for all backward hidden states $b_1, ...b_T$.
3. *UPDATE WEIGHTS*

3 Results and Discussion

3.1 Experiment Setup

For data v1, we first split 80% of data as training data and 20% of data as test data. For both baseline and BDNN, we use the same setting, we train all models in 1000 epochs, the Adam optimizer with a 0.01 learning

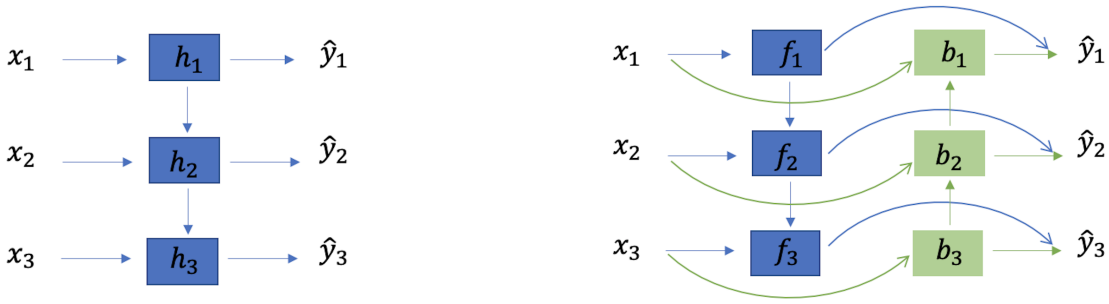

Fig. 5. (a) LSTM (b) Bi-LSTM

rate is chosen, the activation function is Relu, and the number of hidden neurons is 1000. For input patterns, both RGB features and thermal features are fused.

For data v2, we also split 80% of data as training data and 20% of data as test data. However, we only use the first 6 "films" for each video to consistent with previous work on video data [17]. For our model, we use ResNet152 as the backbone network. In the Bi-LSTMs part, benefits from the new functions of Pytorch [14], we can simply achieve Bi-directional training using the LSTM function. We train our Bi-LRCN on data v2 in 40 epochs. To be a fair comparison, we also train all other models on data v2 in 40 epochs and fine-tune them. For the optimizer, we use Adam with a 1e-3 initial learning rate. The whole training takes about 3 hours with a training batch size 64 on a server with 3 NVIDIA GeForce RTX 2080 GPUs. In the following experiments on data v2, we will train some popular CNN models to do a comparison but will not compare with [17] due to some details about their experiments are lack and thus hard to do fairly comparison. Further, due to the label distribution is balanced, we only use accuracy as our evaluation metric.

3.2 Baseline Method

Model tuning For the baseline model, we train it with many different settings. We find the number of hidden neurons and activation functions can significantly affect the performance. So, we try Relu and Sigmoid activation function, we also change the number of hidden neurons with 10, 50, 200, and 500 respectively. The result is shown in Table 1. We find our model with the Sigmoid function performs better than the model with the Relu function. The reason is Relu function can avoid gradient vanishing, but our model is a shallow network, so, the Sigmoid function is good enough and it usually performs better on the binary classification task. We also can find the performance is relatively better when changing the number of hidden neurons into 200.

Table 1. Accuracy(%) for different activation functions and number of hidden neurons

n	train with Relu	test with Relu	train with Sigmoid	test with Sigmoid
10	73.99	50.00	73.99	41.94
50	85.48	45.97	94.96	50.81
200	88.31	48.39	**95.77**	**51.61**
500	91.33	48.39	92.54	49.19

The model with different modalities We also explore the performance when using different modalities and optimizers. For this experiment, we follow the setting in Section 3.1, we train the model in 1000 epochs using the Sigmoid function and 200 hidden neurons. For optimizer, we try Adam and SGD with a 0.01 learning rate. For our input data, we try RGB, thermal, and bimodal input. The RGB input contains the top 5 PCA features from RGB images, the thermal input contains the top 5 PCA features from thermal images, and bimodal input combines the RGB input and thermal input using feature-level fusion. Our result is shown in Table 2. We find the model with Adam performs better than the model with SGD. Also, the model with bimodal input achieves the best performance. So, we can conclude the complementary information between RGB features and thermal features is helpful in improving performance. Finally, the best setting we choose for the baseline is bimodal input, Adam optimizer, Relu function, and 200 hidden neurons.

Table 2. Accuracy(%) for different modalities and optimizers

modalities	train with Adam	test with Adam	train with SGD	test with SGD
RGB	81.65	51.61	52.82	50.81
thermal	81.05	45.97	51.81	48.39
RGB+thermal	**95.77**	**51.63**	53.63	51.61

3.3 Experiments on Bidirectional Neural Network

Performance with balanced Bidirectional Neural Network We use the same setting as our best baseline. But BDNN has extra hyper-parameters, which are the forward training epochs and balance term. For our balanced Bi-directional training scheme, we use the balance term to balance the forward training and reverse training. We compare the performance with balance term and without balance term, and try the different number of epochs, the result is shown in Table 3. Notice that β=1 denotes Bi-directional training without a balance term and β=0.5 denotes Bi-directional training with the balance term. We find that balanced Bi-directional training outperforms Bi-directional training in all different forward epochs settings. This is because this balanced training scheme makes the model focus more on the classification task, and also makes the model predict well in reversed direction due to the better classification result.

Table 3. Comparison between Bi-directional training and balanced Bi-directional training

forward epochs	train(β=1)	test(β=1)	train(β=0.5)	test(β=0.5)
2	68.15	42.74	**73.79**	**51.65**
10	61.9	46.77	**67.94**	**48.39**
50	83.87	**51.61**	**85.48**	43.55
100	81.84	45.16	**85.69**	**55.61**

Discussion and Analysis We compare the performance between our baseline and our BDNN. The result is shown in Table 4. We find BDNN is harder to converge in the training set. We also find BDNN has better generalization ability due to the advantage of Bi-directional training. However, the performance is still not satisfactory. Although models have a very high training accuracy, which fits well on the training set, the features in our dataset are not enough to predict stress in the test set. To do further analysis, we move to the video dataset in the following sections.

Table 4. Comparison between baseline and BDNN

model	train accuracy	test accuracy
baseline	95.77	51.61
BDNN	85.69	**55.61**

3.4 Experiments on Bi-BRCN using data v2

Model Evaluation We compare our model with BDNN, traditional 2DCNN, 3DCNN[6], and LRCN[2]. All models are trained in 40 epochs and are fine-tuned. The result is shown in Table 5. By seeing the table, BDNN has the worst performance as the spatial features are got by dimension reduction, lacking a lot of important features. 2DCNN has a better performance than BDNN as CNN can automatically learn useful spatial features and invariance. 3DCNN and LRCN further improve the performance compared to 2DCNN as they both additionally consider the temporal information. By experiment, we find 3DCNN costs much memory and we only can choose a small batch size. Finally, we can find our model outperforms all models on the test set, which proves the effectiveness of our model on the video dataset. In the following sections, we will do some ablation studies to better demonstrate the effectiveness of important components in our model.

Impact of different backbones We report our performance using different backbones in Table 6. All architectures are the same but the backbone. The first model's backbone is a manually designed CNN. Next two are classical backbones VGG [21] and ResNet[3]. We can find ResNet outperforms all other models as skip connection in ResNet can avoid gradient vanishing and is useful to train a very deep network. It also can prove the importance of the ResNet backbone of our model in video datasets.

Table 5. Comparison between different models

model	test accuracy
BDNN	55.6
2DCNN	58.4
3DCNN	60.84
LRCN	61.4
Ours	**68.4**

Table 6. Comparison with different backbones

model	test accuracy
CNN+Bi-LSTM	58.3
VGG+Bi-LSTM	65.4
ResNet+Bi-LSTM	**68.4**

Impact of different sequence models We employ different sequence models in our model, which are RNN [11], GRU [1], LSTM [4], and our Bi-LSTM. The result is shown in Table 7. By comparing the first three models we can find LSTM has the best performance as it has more gates to memorize the long-range sequence and can avoid gradient vanishing. But by experiments, we find GRU can achieve similar performance with LSTM while uses less memory and is faster than LSTM. So, LSTM and GRU both have their advantages and we can choose them according to our aim. Finally, we can see the performance of Bi-LSTM is improved by 7% compared to LSTM, which proves the effectiveness of our Bi-directional training scheme.

Table 7. Comparison with different sequence models

model	test accuracy
ResNet+RNN	53.2
ResNet+GRU	61.1
ResNet+LSTM	61.4
ResNet+Bi-LSTM	**68.4**

4 Conclusion and Future Work

In this paper, we study stress recognition problems using the Bidirectional Neural Network on the compressed ANUStressDB dataset and using Bi-directional Long-term Recurrent Convolutional Network on the video version dataset. We explore different modalities on stress recognition and show that bimodal input is helpful to improve performance. We also find Bi-directional training is harder to converge but has better generalization ability. Moreover, we extend the Bi-directional training scheme into LSTM and find it is helpful for modeling video sequence data. Extensive experiments show that our proposed Bi-LRCN can significantly improve performance than traditional sequence modeling-based models on video datasets.
Although our model achieves the best results on ANUStressDB dataset compared to other models, the performance is still not satisfactory enough. We summarize the problems as: (1) we simply treat input video as flat image sequence while ignoring the different granularity's information in video content. (2) the fusion method in our model is just 2 fully connected layers at the end of Bi-LSTMs part, which might do not fully exploit the feature representations. In the future, we will extend our architecture that combining both CNN, 3DCNN, and Bi-LSTM in a hierarchical way so that each part can focus on different granularity information. Also, a more reasonable fusion method will be explored.

References

1. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR **abs/1412.3555** (2014), http://arxiv.org/abs/1412.3555
2. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. CoRR **abs/1411.4389** (2014), http://arxiv.org/abs/1411.4389
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385

4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–80 (12 1997). https://doi.org/10.1162/neco.1997.9.8.1735
5. Irani, R., Nasrollahi, K., Dhall, A., Moeslund, T., Gedeon, T.: Thermal super-pixels for bimodal stress recognition. pp. 1–6 (12 2016). https://doi.org/10.1109/IPTA.2016.7821002
6. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1), 221–231 (2013). https://doi.org/10.1109/TPAMI.2012.59
7. Kosko, B.: Bidirectional associative memories. IEEE Transactions on Systems, Man, and Cybernetics **18**(1), 49–60 (1988). https://doi.org/10.1109/21.87054
8. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Inc., USA (1991)
9. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems **25** (01 2012). https://doi.org/10.1145/3065386
10. Kumar, S., Iftekhar, A.S.M., Goebel, M., Bullock, T., MacLean, M.H., Miller, M.B., Santander, T., Giesbrecht, B., Grafton, S.T., Manjunath, B.S.: Stressnet: Detecting stress in thermal videos. CoRR **abs/2011.09540** (2020), https://arxiv.org/abs/2011.09540
11. Mcculloch, W., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5**, 127–147 (1943)
12. Miller, G., Cohen, S., Ritchey, A.: Chronic psychological stress and the regulation of pro-inflammatory cytokines: A glucocorticoid-resistance model. Health psychology : official journal of the Division of Health Psychology, American Psychological Association **21**, 531–41 (12 2002). https://doi.org/10.1037/0278-6133.21.6.531
13. Nejad, A., Gedeon, T.: Bidirectional neural networks and class prototypes. In: Proceedings of ICNN'95 - International Conference on Neural Networks. vol. 3, pp. 1322–1327 vol.3 (1995). https://doi.org/10.1109/ICNN.1995.487348
14. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
15. Rumelhart, D., Hinton, G., Williams, R.: Leaning internal representations by back-propagating errors. Nature **323** (01 1986)
16. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997). https://doi.org/10.1109/78.650093
17. Sharma, N., Dhall, A., Gedeon, T., Goecke, R.: Modeling stress using thermal facial patterns: A spatio-temporal approach. pp. 387–392 (09 2013). https://doi.org/10.1109/ACII.2013.70
18. Ushiyama, T., Mizushige, K., Wakabayashi, H., Nakatsu, T., Ishimura, K., Tsuboi, Y., Maeta, H., Suzuki, Y.: Analysis of heart rate variability as an index of noncardiac surgical stress. Heart and vessels **23**, 53–9 (01 2008). https://doi.org/10.1007/s00380-007-0997-6
19. Vitetta, L., ANTON, B., CORTIZO, F., Sali, A.: Mind-body medicine: Stress and its impact on overall health and longevity. Annals of the New York Academy of Sciences **1057**, 492 – 505 (12 2005). https://doi.org/10.1111/j.1749-6632.2005.tb06153.x
20. Zhai, J., Barreto, A.: Stress recognition using non-invasive technology. pp. 395–401 (01 2006)
21. Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 (09 2014)