Exploration of analyzing relationship between sequential data of audience and level of depression videos through recurrent neuron network

Xiaoxiang Kong
Research School of Computer Science
Australian National University
Canberra ACT 2601
u6828507@anu.edu.au

Abstract. It is widely acknowledged that people can sense others' grievance and even depression by observing the expression and movements [1]. During the experiment described in the paper, 12 normal observers watched the videos of various depression level and their Galvanic Skin Response (GSR), Skin Temperature (ST) and Pupillary Dilation(PD) of binoculus were recorded by sensors and cameras. The data was re-formalized to sequential format so that a recurrent neuron network can be applied to detect the characteristics. In this paper, 2 different models are implemented to detect the relationship between the mentioned statistics and depression level of a video. The models are compared with each other after they process the properly pre-processed data. Various tricks such as different methods of padding, shuffling, cross validation are applied to promote the performance. After that, both models are evaluated, and some extra assumptions and queries are mentioned by the author. Finally, some conclusions are gained.

Keywords: Depression level, Recurrent Neuron Networks, multi-classes, 1-fold validation, SGD, Adam, Padding

1 Introduction

The main task of this paper is to adequately use the new version of datasets(previously used in [2]) to construct an appropriate recurrent neuron network, which can properly predict depression levels of various videos based on sequential 4-dimension physical signs (GSR, Left-PD, Right-PD, ST). Before the prediction starts, the original data needs to be properly and carefully pre-processed based on the mechanism of current model . During the process of prediction, different tricks are implemented and compared to find the best method of training current model. The validation set was prudently chosen because the model is expected to predict depression level while someone is watching the movie. In the end, the author tried to verify some queries and assumptions based on the phenomenon occurred in the process.

1.1 Dataset

The dataset comes from the statistics of a set of sensors and cameras in the watching depression video experiment [1]. The 12 participants watching the German potential depression patients do not understand German language and male-female ratio of them is 1:1. It means that the effect of needless semantic information and different genders were already rigorously avoided. The 3 physical signals come from different equipment. *Galvanic Skin Response (GSR)* is also known as skin conductance or electrodermal activity responses, which reflects a person's electricity flow through the skin [3]. It is caused by variation of amount of sweat on the skin. *Skin Temperature (ST)* could be understood by its name, and it changes when vasodilatation of peripheral blood vessels occurs. ST is considered to be negatively correlated with unpleasant emotions [4]. *Pupillary Dilation (PD)* mainly reflects the pupil size, which was bigger after positive or negative stimuli [5] and in this dataset, it was divided into 2 parts to record the corresponding data of both eyes. The devices to collect these signs were properly set up to record change of them. During the candidates are watching the movie, their physical signs (GSR, PD, ST) are recorded all the time to form the sequential data in current dataset. The levels of depression videos were manually divided in to 4 levels (0, 1, 2, 3) in advance based on the depression scores from 0 to 63 and the actual depression level of each video was derived by the Beck Depression Inventory – II (BDI-II) [6]. There are 16 videos whose ratio of depression levels is 1:1:1:1. The average length of one dataset is about 2000. However, almost every dataset has some missing values(nan or zero), which may be caused by sample frequency of the devices in the lab or accidentally failure during the original experiments.

1.2 Task description and Model setup

The task is to construct a high-performance classifier based on the knowledge of traditional neuron network and recurrent neuron network. The original data is sequential because it was observed in the duration of each video. each entity in every dataset represents the statistics (GSR, left-PD, right=PD, ST) of one candidate watching one movie. Hence, this format of data is suitable for recurrent neuron network (RNN), which takes sequential data as input and tries to process it using its recurrent construction. More specifically, traditional RNN has many disadvantages when it faces such long sequential data. The derivatives of its weights in very early time steps are possibly exploding or vanishing because they need to be multiplied by many times when we perform backpropagation (BP) algorithm to update the weights. Except for that, as the depth of network grows, the whole model is becoming easier to trapped into a local optimal solution. Hence, the author decided to use LSTM to process the sequential data. Compared with naïve RNN, LSTM can control the derivative of current state of memory with respect to that of previous one by several gates. After process the sequential data, the model needs another neuron network to deal with the output of the LSTM. This neuron network takes the specific output of LSTM as the input, which contains the information of the sequential data, then return a vector whose number of dimensions equals to the categories of the labels (Here is 4 because the level is 0, 1, 2, 3). Besides, for contrast purpose, a traditional 3-layer neuron network is also constructed and trained for this task. Following the fundamental mechanism, the number of input neuron network equals to the number of input features (GSR, Left-PD, Right-PD, ST). To make the most of the datasets, I decided to use the mean value of the sequential data in one single video to represent the information of that video duration. When the neuron networks executed backward propagation, I attempted both SGD (random gradient descent optimization algorithm) and Adam optimizer for comparison of the final output. Since the number of layers is only 3 (with 1 hidden layer), gradient disappearance or gradient eruption is not a disturbing problem. The output of each model contains 4 values, they will present the probabilities of belonging to corresponding class after they are processed by SoftMax function. This function is usually used in multi-class classification task because it can normalize a set of float numbers based on their original value (It means that the order of the value will not be changed) and ensure that the sum of them equals to 1, which is suitable to representing the probability of each class. The model selects the class holding the largest probability as the classification result.

To express the constructions of models more clearly, a corresponding figure (Fig. 1) is used to demonstrate it.

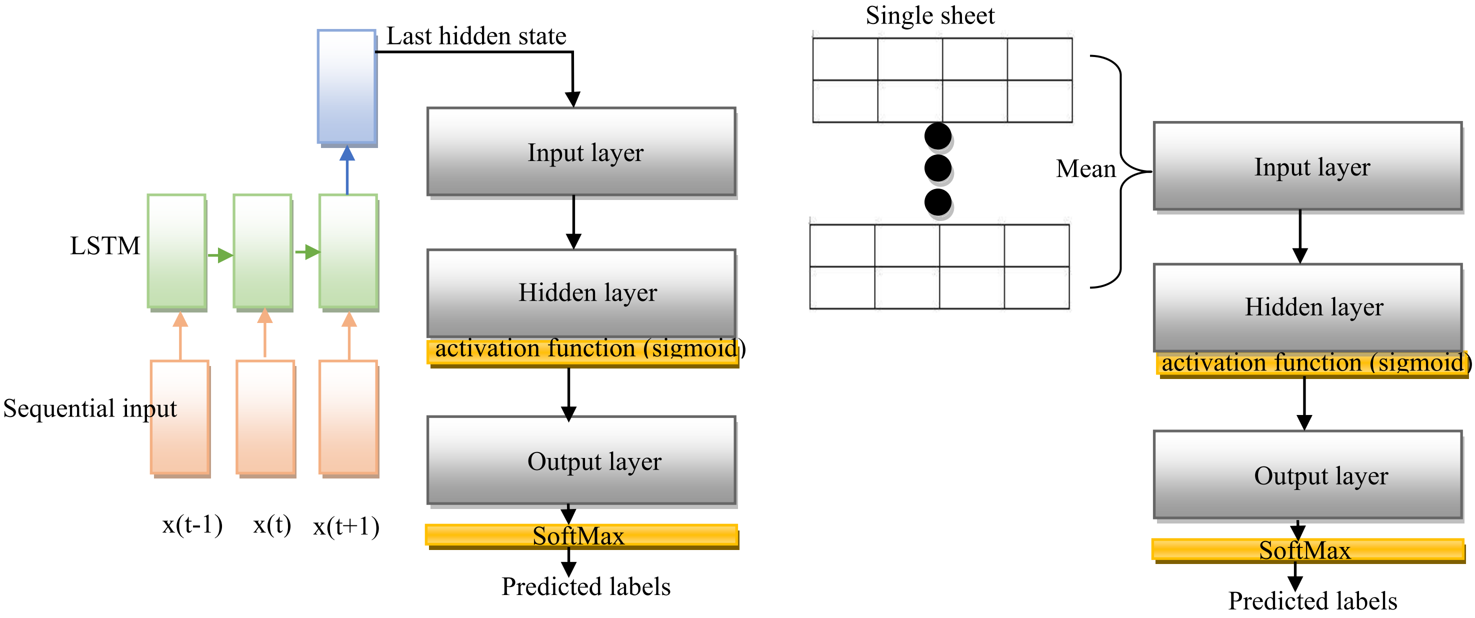


Fig. 1. Two different models of this paper. The first one is the combination of LSTM and a 2-layer neuron network, while the second one is a traditional 2-layer neuron network which takes the mean of a dataset as an input in one epoch.

2 Method

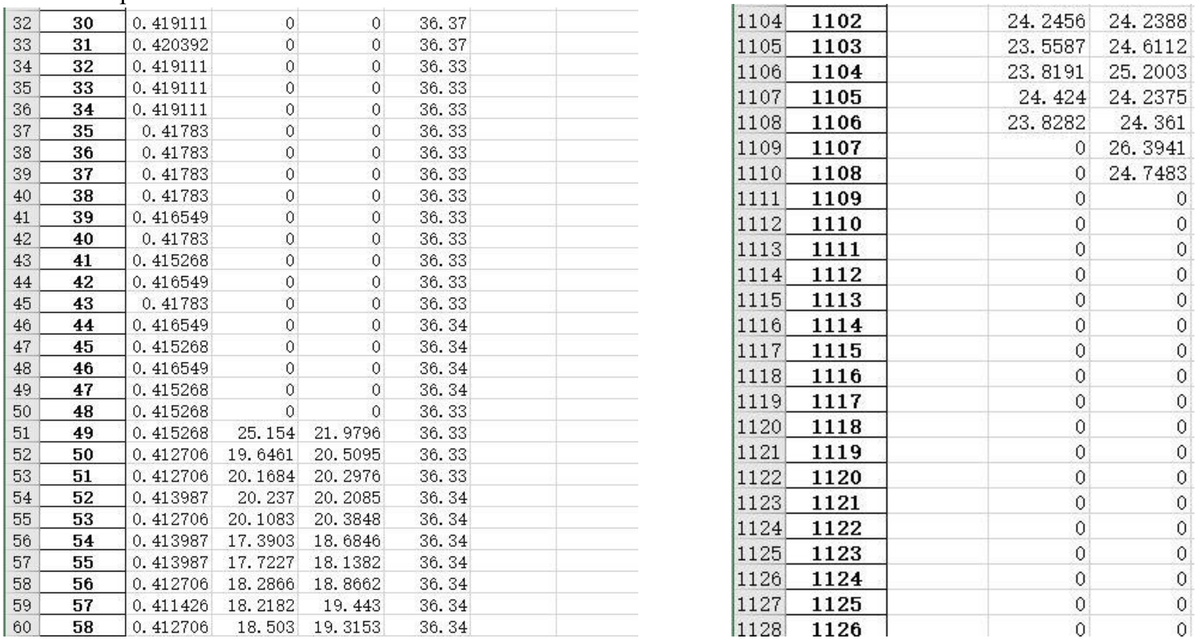
The software operating environment is python 3.8; IDE: PyCharm 2021.1.1; Operating system: Windows 10.
The hardware basis of this paper is: Memory: 16.0GB; CPU: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz

2.1 Data preprocess

The layout of each sheet of each dataset is similar. The entities in the sheet represent the 4 physical signs during that candidate is watching that movie along with the timeline. Hence, it is natural to split them based on the participant, too.

Referring to the method adopted in the experiment [1], I implemented 1-participant-fold validation in total datasets. It is a type of cross-validation. The original idea of cross-validation is to randomly partition data into k equal-sized subsets. During each whole training and validating process, one of the subsets will be considered as validation set and the rest would be training set. In this task, 12 participants watched 16 videos and totally produced 12*16=192 sheets. It is reasonable of us to choose one participant's all data as the validation set and the rest as training set. Because if the depression recognizer based on this trained neuron network is applied to judge the depression level by the statistics from an audience, this audience's physical statistics is unlikely to appear in our network's training set. So, it is essential for the network to learn the reflection from a human caused by the depression video rather than remember the pattern of current individual's physical characteristics.

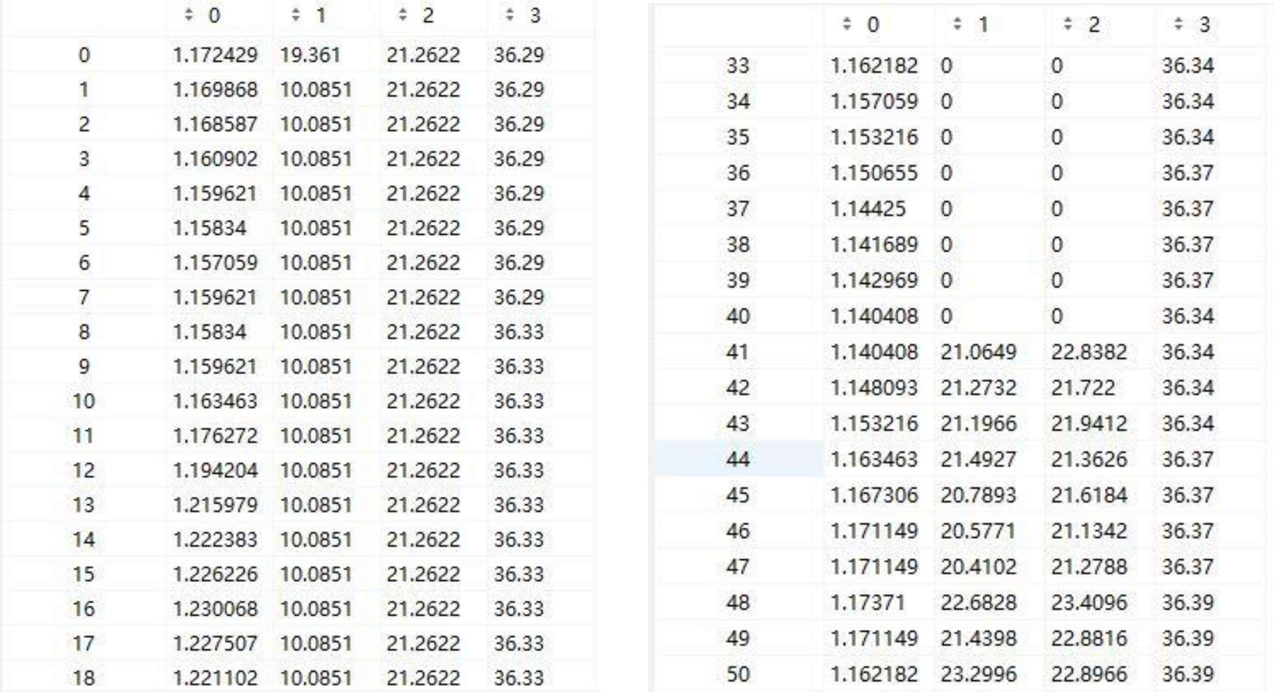
However, the original content of datasets looks not perfect because there are many zeros or missing values (nan) with random distribution. After carefully observe the datasets, I did not sum up the rules of missed signs and not infer any acceptable reason of this phenomenon. Because of the necessity of making the most of current dataset and forcing the neuron network to learn as much as possible, I decided to appropriately prune and pad the datasets. To prune them properly, I implemented a discriminator to delete the empty frame whose all signs are either 0 or nan in sheets of every dataset. After the pruning process, the pre-processor pads the data based on 2 different strategies—zero padding and nearest neighbor padding. Zero padding means that keeping all zero values in the rest frames and change nan value to zero, too. It ensures that the length of each frame is the same and it did not add other information to the datasets. However, in the process of the second model (traditional NN model), since the mean of all the frames in one sheet will be taken as an input to that model, zero padding bounds to significantly change the mean value randomly. Hence, in the second model, I adopt the nearest padding method, which means that padding the missing value with the nearest known value in another frame. Nearest padding will not obviously affect the mean of the dataset, meanwhile, it can also be considered as a remedial method of missing values caused by accidental sensor failure. Hence, the LSTM+NN model uses both padding methods, while the NN model uses nearest padding only. All the datasets have been pruned before they attend the process of both models.



32	**30**	0.419111	0	0	36.37		
33	**31**	0.420392	0	0	36.37		
34	**32**	0.419111	0	0	36.33		
35	**33**	0.419111	0	0	36.33		
36	**34**	0.419111	0	0	36.33		
37	**35**	0.41783	0	0	36.33		
38	**36**	0.41783	0	0	36.33		
39	**37**	0.41783	0	0	36.33		
40	**38**	0.41783	0	0	36.33		
41	**39**	0.416549	0	0	36.33		
42	**40**	0.41783	0	0	36.33		
43	**41**	0.415268	0	0	36.33		
44	**42**	0.416549	0	0	36.33		
45	**43**	0.41783	0	0	36.33		
46	**44**	0.416549	0	0	36.34		
47	**45**	0.415268	0	0	36.34		
48	**46**	0.416549	0	0	36.34		
49	**47**	0.415268	0	0	36.34		
50	**48**	0.415268	0	0	36.33		
51	**49**	0.415268	25.154	21.9796	36.33		
52	**50**	0.412706	19.6461	20.5095	36.33		
53	**51**	0.412706	20.1684	20.2976	36.33		
54	**52**	0.413987	20.237	20.2085	36.34		
55	**53**	0.412706	20.1083	20.3848	36.34		
56	**54**	0.413987	17.3903	18.6846	36.34		
57	**55**	0.413987	17.7227	18.1382	36.34		
58	**56**	0.412706	18.2866	18.8662	36.34		
59	**57**	0.411426	18.2182	19.443	36.34		
60	**58**	0.412706	18.503	19.3153	36.34		

1104	**1102**		24.2456	24.2388
1105	**1103**		23.5587	24.6112
1106	**1104**		23.8191	25.2003
1107	**1105**		24.424	24.2375
1108	**1106**		23.8282	24.361
1109	**1107**		0	26.3941
1110	**1108**		0	24.7483
1111	**1109**		0	0
1112	**1110**		0	0
1113	**1111**		0	0
1114	**1112**		0	0
1115	**1113**		0	0
1116	**1114**		0	0
1117	**1115**		0	0
1118	**1116**		0	0
1119	**1117**		0	0
1120	**1118**		0	0
1121	**1119**		0	0
1122	**1120**		0	0
1123	**1121**		0	0
1124	**1122**		0	0
1125	**1123**		0	0
1126	**1124**		0	0
1127	**1125**		0	0
1128	**1126**		0	0

Fig. 2. Views of original frames in one sheet of one dataset with missing values.



	0	1	2	3
0	1.172429	19.361	21.2622	36.29
1	1.169868	10.0851	21.2622	36.29
2	1.168587	10.0851	21.2622	36.29
3	1.160902	10.0851	21.2622	36.29
4	1.159621	10.0851	21.2622	36.29
5	1.15834	10.0851	21.2622	36.29
6	1.157059	10.0851	21.2622	36.29
7	1.159621	10.0851	21.2622	36.29
8	1.15834	10.0851	21.2622	36.33
9	1.159621	10.0851	21.2622	36.33
10	1.163463	10.0851	21.2622	36.33
11	1.176272	10.0851	21.2622	36.33
12	1.194204	10.0851	21.2622	36.33
13	1.215979	10.0851	21.2622	36.33
14	1.222383	10.0851	21.2622	36.33
15	1.226226	10.0851	21.2622	36.33
16	1.230068	10.0851	21.2622	36.33
17	1.227507	10.0851	21.2622	36.33
18	1.221102	10.0851	21.2622	36.33

	0	1	2	3
33	1.162182	0	0	36.34
34	1.157059	0	0	36.34
35	1.153216	0	0	36.34
36	1.150655	0	0	36.37
37	1.14425	0	0	36.37
38	1.141689	0	0	36.37
39	1.142969	0	0	36.37
40	1.140408	0	0	36.34
41	1.140408	21.0649	22.8382	36.34
42	1.148093	21.2732	21.722	36.34
43	1.153216	21.1966	21.9412	36.34
44	1.163463	21.4927	21.3626	36.37
45	1.167306	20.7893	21.6184	36.37
46	1.171149	20.5771	21.1342	36.37
47	1.171149	20.4102	21.2788	36.37
48	1.17371	22.6828	23.4096	36.39
49	1.171149	21.4398	22.8816	36.39
50	1.162182	23.2996	22.8966	36.39

Fig. 3. View of padded frames by nearest-method and those by zero-method.

The specific values of each attribute are also carefully observed. I do not think there are many large gaps between different. Besides, the dimension of each sample is small (4). Hence this time PCA or GA algorithm selecting partial attributes to attend process is not necessary.

As the last step before the training datasets are taken as the inputs to attend the training process, they need to be randomly shuffled with their corresponding labels so that the model is prevented from potential misunderstanding that some kinds of videos appear with other kinds of videos. The method of shuffler is in shuffler.py.

2.2 Strategies of model processing

2.2.1 LSTM+NN classifier with nearest padding/zero padding with SGD/Adam

This kind of neuron networks is the first one that the author used to construct the classifier. SGD can reduce the calculated amount of the device because it processes BP algorithm based on sub-samples each time. The setup of SGD can control the momentum of that algorithm, which leads to quicker convergence. Because if the direction of current gradient decent is the same as the previous one, the momentum will prompt the parameters to descend more quickly. Otherwise, it will slow down that to avoid shock. In this sub-experiment, I chose the normal validation process—chose a fixed ratio of various-level depression videos ignoring which participants they are from. Adam means adaptive moment estimation. Compared to SGD, Adam uses the first moment and the second moment of gradient to adaptively adjust the learning rate of each parameter [7]. Adam has been chosen as the best optimizer in nowadays Machine Learning tasks.

2.2.2 Traditional NN classifier with nearest padding with SGD

2.2.3 Splicing LSTM+NN classifier with nearest padding/zero padding

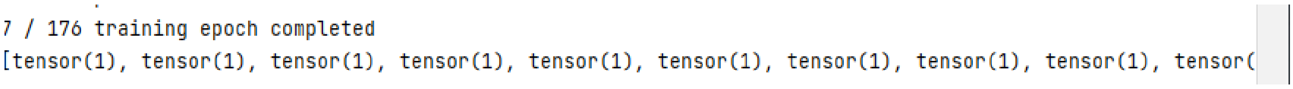
This scheme was put forward by me after all the results of schemes above were gained. It means that use multi "final" hidden states of LSTM to be the input of the NN component so that the model can grasp more sequential information of current dataset. To realize it, the model needs to set length of a single sequence. The model goes through the current dataset and generalize one final state in one iteration. In each iteration, the LSTM take len_seq frames as inputs. When the LSTM finish processing current dataset, the final hidden states are integrated to represent the information of this duration of current candidate's watching. It is a pity that since the limitation of time, I did not implement this new model.

3 Results and Discussion

The result of each scheme is not as good as expected. All of them do not tend to be convergent even though I adopted a series of tricks to avoid. More specifically, the loss jumped from about 1.20 to 1.80 during both models' training processes. The highest validation accuracy of all the schemes is only 31.25%, and it can not be kept in the subsequent training epochs.

3.2 Discussion and Conclusion

Although the outcome is depressed, I still find some curious facts in it. When I printed the predicted labels of validation set in every epoch of training, I found that the model always tends to think the 16 videos in validation share the same depression level, like Fig. 4.

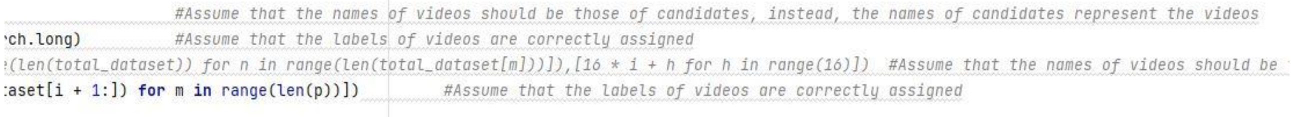


```
7 / 176 training epoch completed
[tensor(1), tensor(1), tensor(1), tensor(1), tensor(1), tensor(1), tensor(1), tensor(1), tensor(1), tensor(
```

Fig. 4 printed outcome of validation set shows that the classifier always tends to think the validation videos have the same depression level.

Naturally, I made an assumption that the names of videos and names of candidates were wrongly exchanged, and the videos in the validation set should from a same label rather than a same people.

Hence, I re-wrote the method of label generalization in data pre-processing to treat the number of candidates as the depression level of video. Unfortunately, the result was still bad. In this situation, the precision in validation set was either 100% or 0. Although this assumption was disproved, I still reserved the relevant code(Fig. 5).



```
                     #Assume that the names of videos should be those of candidates, instead, the names of candidates represent the videos
ch.long)             #Assume that the labels of videos are correctly assigned
(len(total_dataset)) for n in range(len(total_dataset[m]))]),[16 * i + h for h in range(16)])  #Assume that the names of videos should be
aset[i + 1:]) for m in range(len(p))])            #Assume that the labels of videos are correctly assigned
```

In conclusion, current construction of LSTM cannot learn the sequential information of this dataset. A more powerful model needs to be found.

References

[1] Richard Jones, Xuanying Zhu, Tom Gedeon and Sabrina Caldwell, "Detecting emotional reactions to videos of depression".

[2] Xiaoxiang Kong, "Exploration of analyzing relationship between physical signs of audience and depression videos through neuron networks".

[3] R. M. Stern, W. J. Ray, and K. S. Quigley, Psychophysiological recording. Oxford University Press, USA, 2001.

[4] N. Sharma and T. Gedeon, "Modeling stress recognition in typical virtual environments," in Proceedings of the 7th international conference on pervasive computing technologies for healthcare, 2013, pp. 17–24.

[5] B. Laeng, S. Sirois, and G. Gredebäck, "Pupillometry: a window to the preconscious?," Perspect. Psychol. Sci., vol. 7, no. 1, pp. 18–27, 2012.

[6] A. T. Beck, R. A. Steer, and G. K. Brown, "Beck depression inventoryII," San Antonio, vol. 78, no. 2, pp. 490–498, 1996.

[7] Diederik P. Kingma and Jimmy Lei Ba, "Adam: A Method for Stochastic Optimization", ICLR, 2015

[8] https://piazza.com/class/kkuwnbmtk6ut5?cid=394