3D-CNN for Facial Emotion Recognition in Videos

Jad Haddad, Olivier Lézoray, Philippe Hamel

3D-CNN for Facial Emotion Recognition in Videos

Jad Haddad[1,2], Olivier Lezoray[1], and Philippe Hamel[2]

[1] Université de Caen, GREYC, CNRS UMR 6072, Caen, France
[2] Zero To One Technology, Campus Effiscience, Colombelles, France

Abstract. In this paper, we present a video-based emotion recognition neural network operating on three dimensions. We show that 3D convolutional neural networks (3D-CNN) can be very good for predicting facial emotions that are expressed over a sequence of frames. We optimize the 3D-CNN architecture through hyper-parameters search, and prove that this has a very strong influence on the results, even if architecture tuning of 3D CNNs has not been much addressed in the literature. Our proposed resulting architecture improves over the results of the state-of-the-art techniques when tested on the CK+ and Oulu-CASIA datasets. We compare the results with cross-validation methods. The designed 3D-CNN yields a 97.56% using Leave-One-Subject-Out cross-validation, and 100% using 10-fold cross-validation on the CK+ dataset, and 84.17% using 10-fold cross-validation on the Oulu-CASIA dataset.

Keywords: Facial Emotion Recognition · Video · 3D-CNN · CK+ · Oulu-CASIA

1 Introduction

Facial emotion recognition has been gaining a lot of attention over the past decades with applications in cognitive sciences and affective computing. Ekman et al. have identified six basic facial expressions (anger, disgust, fear, happiness, sadness, and surprise) as basic emotional expressions that are universal and common among human beings [1]. Human emotions are complex to interpret, and building recognition systems is essential in human-computer interaction since affective information is a major component of human communication. Many approaches have been proposed for automatic facial expression recognition [2]. Most of the traditional non-deep approaches have focused on analyzing static images independently, thus ignoring the temporal relations of sequence frames in videos. However this temporal information is essential for tracking small changes in the face throughout the expression of an emotion. Recently, with the surge of deep learning, more promising results have been reported [3,4,6,5,7] tackling the automatic facial emotion recognition task using both geometric and photometric features.

2 Emotion recognition in videos

Predicting dynamic facial emotion expressions in videos has received a lot of attention. Many previous works have explored tracking geometric features of the face relying on the evolution of facial landmarks across frames [8], or have used e.g., the LBP-TOP approach [9]. Recently, deep learning techniques have emerged as they can provide enormous performance gains [7]. Several deep techniques can be considered for analyzing sequential data, the most prominents being Recurrent Neural Networks (RNN) [10], and Long Short-Term Memories (LSTM) [11]. Many works have been led on the combination of classical 2D Convolutional Neural Networks with RNNs or LTSMs to cope with the temporal aspect in emotion recognition in videos [12,13]. In these approaches, a RNN (or LSTM) takes the features extracted by a CNN over individual frames as inputs and encodes the temporal dynamics. Few works have been led on the use of 3D-CNN [14,15] as compared to the combination CNN-LSTM in the facial emotion recognition domain. However convolutional neural networks with 3D kernels (3D-CNNs) can have a superior ability to extract spatio-temporal features within video frames, as compared to 2D CNNs, even if combined with temporal networks. This is the line of the work we propose and we aim at designing an efficient 3D-CNN for emotion recognition in videos [16,3].

3 Our Approach

We consider 3D-CNNs to perform facial emotion recognition in videos. We base our study on [14], and regularize the feature extraction part of the network with batch normalization because of its success in reducing internal covariate shift [17]. We explore how we can optimize the structure and parameters of the network to obtain better performances. Unlike 2D convolution, when a 3D convolution is applied on a video sequence, the output is a 3D tensor. Therefore, 3D convolutions preserve the temporal aspect of a video sequence. In video contexts, facial expressions do not manifest themselves instantly, but instead, they are built up gradually across time until they reach their peak. Thus, a static approach would result in predictions that can vary a lot across the frames and lead to uninterpretable results. A 3D-CNN solves this issue since by nature it takes as input a group of sequential frames and analyzes them together to predict an emotion.

3.1 Deep 3D CNN

The full architecture of our proposed model to be optimized is presented in Figure 1, such that the white components are fixed, and the red components are to be explored. We suppose that an expression can be detected in 10 consecutive frames as in the state of the art. Our model takes as input a window of 10 RGB frames of size 112×112. We use the Adam optimizer and we set the learning rate to 0.0001, and set the maximum number of epochs to 16. We process batches

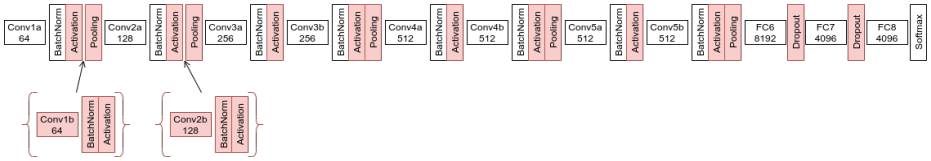

Fig. 1. Full architecture to optimize. Number of filters is denoted in each Conv box.

of size 30, and we report the results of the best epoch. The momentum of batch normalization layers is set to 0.1. We denote the kernel dimensions as (f, w, h) where f represents the temporal dimension, w represents the width dimension, and h represents the height dimension. All convolutional layers have a kernel of size (3, 3, 3), and a padding of (1, 1, 1). However, we will search in the following sections for the best temporal dimension size of the convolutional kernel. All pooling layers have a kernel size of (2, 2, 2) and a stride of (2, 2, 2), except for the first pooling layer, which has a kernel size of (1, 2, 2) and a stride of (1, 2, 2), and for the last pooling layer which has a kernel size of (1, 2, 2), a stride of (2, 2, 2), and a padding of (0, 1, 1).

3.2 Data Augmentation

Data augmentations have proved to increase the task performance of neural networks [18]. When a deep network has many parameteres, it can easily overfit when the size of the training dataset is small. Data augmentation overcomes this issue by artificially creating new samples by applying transformations to the training set. We explore two geometric and one photometric augmentation techniques:

1. **Flip:** flip horizontally.
2. **Rotation:** rotate by a random angle $\alpha \in [-30, 30]$.
3. **Linear contrast:** adjust contrast by scaling each pixel to $127 + \alpha * (v - 127)$ where v is the pixel value and α is a random multiplier $\in [0.22, 2.2]$.

3.3 Architecture Optimization

To find the best combination of all the options that we want to explore, we can do a grid search to explore every possible combination. Even though this technique yields the most accurate results, we would be facing a combinatorial explosion, which is computationally expensive. To tackle this problem, we used the Optuna [3] framework [19] to search for the best hyper-parameters combination. Optuna uses by default Tree-structured Parzen Estimator (TPE) [20], which is more efficient and much less computationally expensive than a grid search. TPE is a sequential model-based optimization (SMBO) approach. SMBO methods sequentially construct models to approximate the performance of hyperparameters

[3] `http://optuna.org/`

based on previous measurements, and then choose new hyperparameters to test, based on this model. On each trial, TPE fits for each parameter one Gaussian Mixture Model (GMM) $l(x)$ to the set of parameter values associated with the best objective values, and another GMM $g(x)$ to the remaining parameter values. Then it chooses the parameter value x that maximizes the ratio $l(x)/g(x)$. We're interested in exploring the following parameters:

- Type of pooling layer.
- Type of activation function.
- Optimizing using Lookahead (k=5, alpha=0.5) [21].
- Applying CLAHE [22] on input images with $clipLimit = 2$ and $tileGridSize = (8, 8)$, as illumination can vary a lot which can result in large intra-class variances, which we want to minimize [7].
- Normalizing images so that each pixel value $\in [-1, 1]$. Normalization increases the robustness of the the training efficacy of a neural network [23]
- Size of the temporal dimension in convolutional kernels and modifying the padding of the kernel so that we preserve the same shape of the temporal dimension.
- Weights initialization [24].
- Regularization using dropout between fully connected layers.
- Adding a second convolutional block in first two layer groups.
- Assigning weight to each class and pass it to cross-entropy loss.

The details of the hyper-parameters search are shown in Table 1. Optuna offers

Parameter	**Options**
Optimization	Lookahead+Adam/Adam
CLAHE	true/false
Normalization	true/false
Activation	ReLU/ELU/pReLU/ leaky ReLU/Mish [25]
Loss weights	true/false
Temporal size	1/3/5/7/9
Initializer	Xavier uniform/Xavier normal
Pooling layer	AvgPooling/MaxPooling
Second ConvLayer	true/false
Dropout	$[0, 1]$

Table 1. Hyper-parameters to explore.

pruning functionality for early stopping trials that are not promising. In this experiment, we prune trials that will yield a LOSO cross-validation accuracy less

than 96.5%, allowing us to iterate faster on the different combinations generated by the estimator.

4 Experiments

In this section we consider two state-of-the-art databases and show that a 3D-CNN with an efficient hyper-paprameter search can lead to very good results.

4.1 Evaluation on CK+

CK+ [26] contains 593 video sequences from 123 subjects. Among these videos, 327 sequences from 118 subjects are labeled with seven basic expression labels, i.e., anger, contempt, disgust, fear, happiness, sadness, and surprise. 266 of these video sequences are not annotated, and are discarded for the rest of the experiment. CK+ does not provide training/testing splits, therefore to compare our model's performance, we adopt the cross-validation technique. We use Leave-One-Subject-Out (LOSO) cross-validation technique as a metric to construct and optimize our network's architecture. However, to compare our results with previous works done on this dataset, we use 10-fold subject-independent cross-validation experiments as most of the state-of-the-art algorithms were evaluated in such a way. We constructed 10 subsets as described in several previous works [27,3], and compute the overall accuracy over 10 folds.

Preprocessing. We process the last 10 frames of each sequence by extracting the face using OpenCV's deep learning model for face detection. We then resize the cropped faces to the scale of 112×112, and rescale the pixels values so that each pixel $\in [0, 1]$. The majority of the video sequences are grayscale, therefore, we convert the few colored video sequences to grayscale RGB to preserve the consistency of the dataset.

Data Augmentation We perform different image augmentations empirically according to the representability of each class, knowing that geometric augmentations outperform photometric methods [18], we obtain a quasi-balanced dataset:

- Contempt: 1×Flip, 7×Rotation, 4×Linear contrast.
- Fear: 1×Flip, 4×Rotation, 4×Linear contrast.
- Sadness: 1×Flip, 4×Rotation, 3×Linear contrast.
- Anger: 1×Flip, 2×Rotation, 1×Linear contrast.
- Happy: 1×Flip, 1×Rotation, 1×Linear contrast.
- Disgust: 1×Flip, 1×Rotation, 1×Linear contrast.
- Surprise: 1×Flip, 1×Rotation, 0×Linear contrast.

Furthermore, we duplicate the last frame for video sequences having less than 10 frames.

Hyper-parameters Optimization More than 800 different configuration have been tested, we show the ones that have their LOSO above the pruning threshold and the best results are in the top right corner. After 600 trials, the TPE starts converging as shown in Figure 2, we observe that certain parameters contribute much in increasing the model's performance (e.g., CLAHE, Xavier uniform, $temporal\_size = 3$) and other parameters lower the model's performance (e.g. normalization, $temporal\_size \in \{1; 7; 9\}$, $dropout \in [0.5, 1]$, adding a second convolutional layer to the first two layer groups).

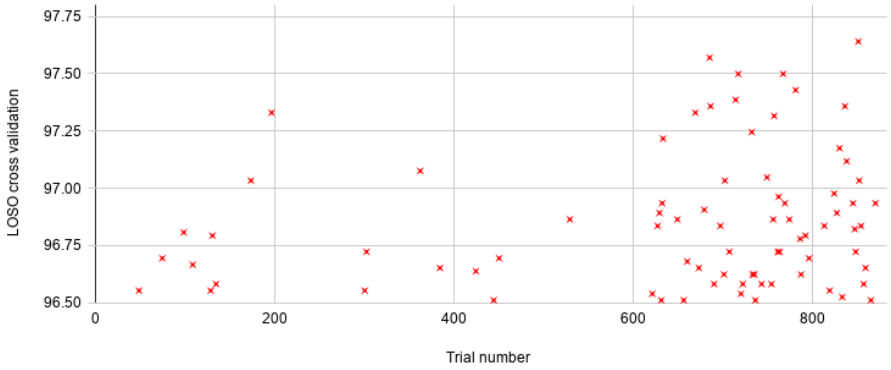

Fig. 2. Hyper-parameters search on CK+. Each red cross represents a different configuration.

Results The best trial of LOSO cross-validation on CK+ yielded **97.56%**. The hyper-parameters combination proposed by the TPE for this accuray is illustrated in Table 2 along with its confusion matrix in Figure 5, and the resulting architecture is illustrated in Figure 3.

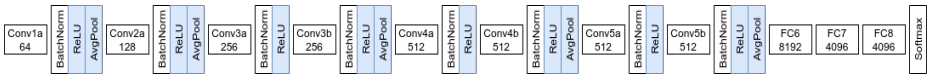

Fig. 3. Resulting architecture for CK+, number of filters is denoted in each Conv box.

We use the resulting architecture to evaluate the 10-fold subject-independent cross-validation, we achieve **100%**. This is so far the best result obtained on this dataset.

Fig. 4. Resulting architecture for Oulu CASIA, number of filters is denoted in each Conv box.

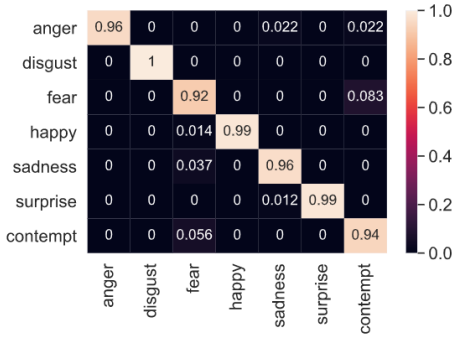

Fig. 5. Confusion matrix of LOSO for CK+.

Hyper-parameter	**Value**
Optimization	Adam
CLAHE	true
Normalization	false
Activation function	ReLU
Loss weights	false
Temporal size	3
Initializer	Xavier uniform
Pooling layer	AvgPooling
Second ConvLayer	false
Dropout	0.2511

Table 2. Hyper-parameters of the best CK+ LOSO trial.

4.2 Evaluation on Oulu-CASIA

We perform hyper-parameter tuning of the full architecture to obtain the optimized 3D-CNN architecture for the Oulu-CASIA dataset.

Oulu-CASIA [28] consists of six expressions (surprise, happiness, sadness, anger, fear and disgust) from 80 people between 23 to 58 years old. 73.8% of the subjects are males. Subjects were filmed by an NIR camera and a VIS camera which capture the same facial expression. All expressions are captured in three different illumination conditions: normal, weak and dark. Normal illumination means that good normal lighting is used. Weak illumination means that only computer display is on and subject sits on the chair in front of the computer. Dark illumination means near darkness. For this experiment, we only use video sequences captured in normal illumination condition.

Preprocessing. We process the last 10 frames of each sequence by extracting the face using OpenCV's deep learning model for face detection. We then resize the cropped faces to the scale of 112 × 112, and rescale the pixels values so that each pixel $\in [0, 1]$.

Data Augmentation We augment all the video sequences using 1 × Flip, 5 × Rotation, 2 × Linear contrast. Furthermore, we duplicate the last frame for video sequences having less than 10 frames.

Results The best trial of 10-fold cross-validation on Oulu CASIA yielded **84.17%**. The hyper-parameters combination proposed by the TPE for this accuray is illustrated in Table 3 along with its confusion matrix in Figure 6, and the resulting architecture is illustrated in Figure 3.

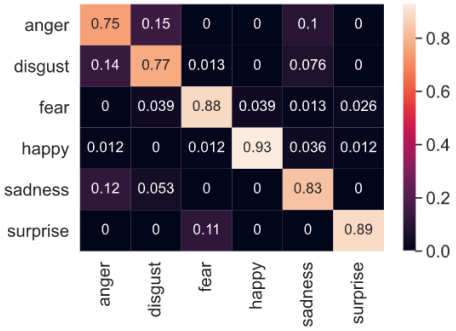

Fig. 6. Confusion matrix of 10-fold for Oulu-CASIA.

Hyper-parameter	**Value**
Optimization	Lookahead + Adam
CLAHE	false
Normalization	true
Activation function	Leaky ReLU
Loss weights	false
Temporal size	3
Initializer	none
Pooling layer	MaxPooling
Second ConvLayer	false
Dropout	0.4233

Table 3. Hyper-parameters of the best Oulu-CASIA 10-fold trial.

4.3 Importance of meta optimization

We evaluate the optimized architecture of CK+ on Oulu-CASIA and vice-versa to see the importance of having an architecture that is optimized to the dataset in question as opposed to having one architecture for both datasets. The p-value is used to determine the significance between the two architectures and therefore, the importance of having a different architecture optimized to each dataset. Tables 4 and 5 show that the p-value is significant. Thus, having an optimized architecture for each dataset is necessary to have a better accuracy.

4.4 Comparison with the state-of-the-art

We evaluate the accuracy of our proposed network architecture. Our approach improves the results of the state-of-the-art on CK+ according to [7] using Leave-One-Subject-Out cross-validation as shown in Table 6. The results of the state-of-the-art according to [2] using 10-fold cross-validation as shown in Table 7. Our approach yields to the best state-of-the-art results so far obtained on this dataset. Our model yields results in the range of the state-of-the-art on Oulu-CASIA using 10-fold cross-validation are shown in Table 8. Our model surpasses the results of [3], and yields similar results as the Spatio-temporal convolutional

	Fold												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average**	**P-value**	**Significance**
c1	100	100	100	100	100	100	100	100	100	100	100		
c2	93.9	100	84.8	78.7	87.8	93.9	93.9	96.8	87.5	90	90.76	0.00057	significant
c3	96.8	96.8	84.8	93.9	87.8	93.9	78.7	96.8	87.5	93.9	91.14	0.00067	significant

Table 4. P-value comparisons between architecture for the CK+ dataset. With **c1** being the optimized architecture for CK+ 3, **c2** the optimized architecture for Oulu-CASIA 4, and **c3** the optimized architecture for Oulu-CASIA 4 pre-trained on the Oulu-CASIA dataset

	Fold												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Average**	**P-value**	**Significance**
c2	97.5	97.5	80	82.5	82.5	80	72.5	97.5	75	77.5	84.25		
c1	81.2	81.2	70.8	75	72.9	72.9	83.3	72.9	72.9	87.5	77.08	0.03650	significant
c4	85	97.5	75	82.5	77.5	70	65	97.5	75	72.5	79.75	0.00597	significant

Table 5. P-value comparisons between architecture for the Oulu-CASIA dataset. With **c1** being the optimized architecture for CK+ 3, **c2** the optimized architecture for Oulu-CASIA 4, and **c4** the optimized architecture for CK+ 3 pre-trained on the CK+ dataset

(STC) used in [29]. We believe that better results could be obtained on this dataset with our approach by focusing more on the temporal aspect by using an additional LSTM, as done in [29]. Finally, our results show the benefits of: (i) using 3D-CNNs over traditionnal CNN or CNN-LSTM approaches, (ii) an efficient hyper-parameter search for a considered 3D-CNN architecture. Regarding this last point, if one looks at the final architectures shown in Figure 3 and 4, one can see that they look very similar. However the best hyper-paremeters are very different (see Tables 2 and 3). This is favor of our proposal that considers an efficient hyper-parameter space exploration.

Approach	**Accuracy** (%)
CNN (AlexNet)[30]	94.4
DAE (DSAE)[31]	95.79
Our approach	**97.56**

Table 6. LOSO results for CK+.

Approach	**Accuracy** (%)
LBP/Gabor + SRC[32]	98.09
DBN + MLP[33]	98.57
CNN[34]	98.62
FAN[27]	99.69
Our approach	**100**

Table 7. 10-fold results for CK+.

Approach	**Accuracy** (%)
FLT[3]	74.17
C3D[3]	74.38
FLT+C3D[3]	81.49
Our approach	84.17
STC[29]	84.72
LSTM (STC-NLSTM)[29]	**93.45**

Table 8. 10-fold results for Oulu-CASIA.

5 Conclusion

We have proposed 3D-CNNs for video-based facial expression recognition. 3D-CNNs can extract very subtle temporal features that enables to go beyond 2D-CNNs. However their design can be delicate and we have proposed to use an efficient hyper-parameter search to address this issue. The experiments have confirmed the benefit of our approach. The results on CK+ show that our network surpasses the actual state-of-the-art results on CK+ with 97.56% for Leave-One-Subject-Out cross-validation and 100% for 10-fold subject-independent cross-validation. Similarly, a rate of 84.17% on Oulu-CASIA for 10-fold subject-independent cross-validation. In future works we plan to combine the video modality with audio recording.

References

1. Ekman, P., Friesen, W.V.: Constants across cultures in the face and emotion. Journal of Personality and Social Psychology **17** (2), 124–129 (1971)
2. Huang, Y., Chen, F., Lv, S., Wang, X.: Facial expression recognition: A survey. Symmetry **11**(10) (2019)
3. Jung, H., Lee, S., Yim, J., Park, S., Kim, J.: Joint fine-tuning in deep neural networks for facial expression recognition. Proceedings of the IEEE International Conference on Computer Vision **2015 Inter**, 2983–2991 (2015)
4. Hasani, B., Mahoor, M.H.: Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural Networks pp. 30–40 (may 2017)
5. Sharma, G., Singh, L., Gautam, S.: Automatic Facial Expression Recognition Using Combined Geometric Features. 3D Research **10**(2) (2019)
6. Mollahosseini, A., Chan, D., Mahoor, M.H.: Going deeper in facial expression recognition using deep neural networks. 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016 (2016)
7. Li, S., Deng, W.: Deep Facial Expression Recognition: A Survey pp. 1–25 (2018)
8. Ghimire, D., Lee, J., Li, Z.N., Jeong, S.: Recognition of facial expressions based on salient geometric features and support vector machines. Multimedia Tools and Applications **76**(6), 7921–7946 (mar 2017)

9. Nigam, S., Singh, R., Misra, A.K.: Local Binary Patterns Based Facial Expression Recognition for Efficient Smart Applications. Springer International Publishing (2019)
10. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(5), 855–868 (2009)
11. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (nov 1997)
12. Li, T.H.S., Kuo, P.H., Tsai, T.N., Luan, P.C.: CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot. IEEE Access **7**, 93998–94011 (2019)
13. Jain, N., Kumar, S., Kumar, A., Shamsolmoali, P., Zareapoor, M.: Hybrid deep neural networks for face emotion recognition. Pattern Recognition Letters **115**, 101–106 (2018)
14. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. Proceedings of the IEEE International Conference on Computer Vision **2015 International Conference on Computer Vision, ICCV 2015**, 4489–4497 (2015)
15. Zhao, J., Mao, X., Zhang, J.: Learning deep facial expression features from image and optical flow sequences using 3D CNN. Visual Computer **34**(10), 1461–1475 (2018)
16. Teja Reddy, S.P., Teja Karri, S., Dubey, S.R., Mukherjee, S.: Spontaneous Facial Micro-Expression Recognition using 3D Spatiotemporal Convolutional Neural Networks. In: 2019 International Joint Conference on Neural Networks (IJCNN). vol. 2019-July, pp. 1–8. IEEE (jul 2019)
17. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. 32nd International Conference on Machine Learning, ICML 2015 **1**, 448–456 (2015)
18. Taylor, L., Nitschke, G.: Improving Deep Learning using Generic Data Augmentation (2017)
19. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining pp. 2623–2631 (2019)
20. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011 pp. 1–9 (2011)
21. Zhang, M.R., Lucas, J., Hinton, G., Ba, J.: Lookahead Optimizer: k steps forward, 1 step back pp. 1–16 (jul 2019)
22. Reza, A.M.: Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology **38**(1), 35–44 (2004)
23. Nawi, N.M., Atomi, W.H., Rehman, M.: The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks. Procedia Technology **11**(Iceei), 32–39 (2013)
24. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research **9**, 249–256 (2010)
25. Misra, D.: Mish: A Self Regularized Non-Monotonic Neural Activation Function (1) (aug 2019)

26. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. pp. 94–101. No. July, IEEE (jun 2010)
27. Meng, D., Peng, X., Wang, K., Qiao, Y.: Frame Attention Networks for Facial Expression Recognition in Videos. Proceedings - International Conference on Image Processing, ICIP **2019-Septe**(September), 3866–3870 (2019)
28. Zhao, G., Huang, X., Taini, M., Li, S.Z., Pietikäinen, M.: Facial expression recognition from near-infrared videos. Image and Vision Computing **29**(9), 607–619 (aug 2011)
29. Yu, Z., Liu, G., Liu, Q., Deng, J.: Spatio-temporal convolutional features with nested LSTM for facial expression recognition. Neurocomputing **317**, 50–57 (2018)
30. Ouellet, S.: Real-time emotion recognition for gaming using deep convolutional network features pp. 1–6 (2014)
31. Zeng, N., Zhang, H., Song, B., Liu, W., Li, Y., Dobaie, A.M.: Facial expression recognition via learning deep sparse autoencoders. Neurocomputing **273**, 643–649 (2018)
32. Zhang, S., Zhao, X., Lei, B.: Facial expression recognition using sparse representation. WSEAS Transactions on Systems **11**(8), 440–452 (2012)
33. Zhao, X., Shi, X., Zhang, S.: Facial expression recognition via deep learning. IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India) **32**(5), 347–355 (2015)
34. Breuer, R., Kimmel, R.: A Deep Learning Perspective on the Origin of Facial Expressions pp. 1–16 (2017)