A Compact Embedding for Facial Expression Similarity

SAHAND AKRAMIPOUR[2,4] AND RABEE PARHIZKARI[1,3]

[1] *Sharif Univercity Of Technology*
[1] *Computer Vision Course by Hanieh Naderi*
[1] *Winter/Spring 2024*
[1] *Rabzi.Parhiz@gmail.com*
[2] *Sahandap@gmail.com*
[3] *400109413*
[4] *401110618*

Compiled June 30, 2024

This is the report for Vision project done by Rabee Parhizkari and Sahand Akramipour. Here is the link to our codes in github A Compact Embedding for Facial Expression Similarity

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

In this project, we explored various methods to enhance the results of the article "A Compact Embedding for Facial Expression Similarity" by Raviteja Vemulapalli and Aseem Agarwala. Our approach involved two primary strategies: improving the efficiency of the pretraining model and experimenting with different neural network architectures to find a more efficient one than the original.

2. PRETRAINING

We began by investigating different Python libraries for face detection. Our objective was to detect faces in our dataset and highlight them by blacking out the non-facial pixels. We tested two different methods: MTCNN, Cascad.

A. Viola-Jones

The Viola-Jones detector is one of the earliest face detection methods. It processes grayscale images by interpreting them as collections of Haar features, consisting of lighter and darker rectangles. These features are computed rapidly using integral images. The features are then fed to a cascade of AdaBoost classifiers, which are ensembles of decision trees arranged sequentially. Each feature is evaluated by the first classifier: if rejected, it is instantly discarded; if accepted, it proceeds to the next classifier. This cascade approach transforms face detection into a task of rejecting non-faces, resulting in quick detection. However, it has limitations, such as reduced accuracy with varying face sizes and lack of robustness under non-ideal conditions (e.g., non-frontal faces, poor lighting).

B. MTCNN

The Multi-Task Cascaded Convolutional Neural Network (MTCNN) is a modern tool for face detection, utilizing a 3-stage neural network detector. Initially, the image is resized multiple times to detect faces of different sizes. The P-network (Proposal) performs the first detection, followed by the R-network (Refine) which filters the detections to obtain precise bounding boxes. The O-network (Output) performs the final refinement of the bounding boxes. MTCNN also detects facial landmarks (e.g., eyes, nose, mouth corners) as an optional feature. The TensorFlow implementation of MTCNN works well, but the PyTorch version is faster, achieving about 13 FPS on full HD videos and up to 45 FPS on rescaled ones. MTCNN is highly accurate and robust, effectively detecting faces with different sizes, lighting conditions, and rotations. Though slower than Viola-Jones, it benefits from GPU acceleration.

3. EVALUATION

We evaluated the models in two steps. First, we applied the three face detection methods to all test images and measured their detection rates. Then, we trained our model using the results from these algorithms and compared the accuracy scores. After testing, we found that the Viola-Jones detector couldn't detect faces 28 percent of the time, MTCNN couldn't detect faces 16 percent of the time.

A. Comparison

You can see the comparison of different models in Fig 2

4. ARCHITECTURAL ENHANCEMENTS TO DENSENET

A. DenseNet Overview

DenseNet connects each layer to every other layer in a feed-forward manner. This dense connectivity enables feature reuse and mitigates the vanishing gradient problem.

Architecture:

Initial Convolution: A single convolutional layer. Dense Blocks: Three dense blocks, each consisting of either BasicBlock or BottleneckBlock. Transition Layers: Two transition layers downsample feature maps and reduce the number of channels. Global Pooling and Classifier: Includes batch normalization, ReLU activation, global average pooling, and a fully connected layer for classification.

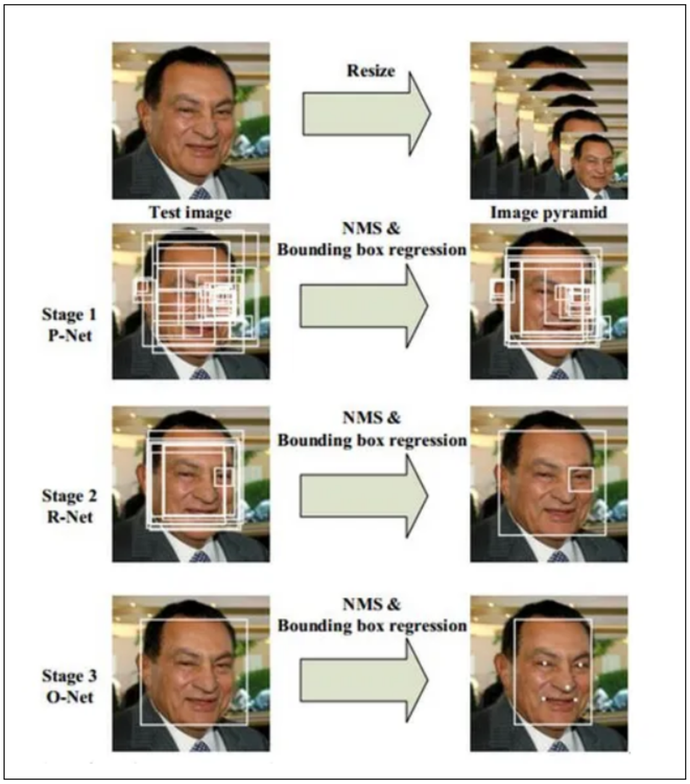

Fig. 1. MTCNN work visualization.

Feature	Viola-Jones	MTCNN
Speed	Very fast (>30 FPS), real-time	Fast (>10 FPS), real-time
Accuracy	Good	Very good
Robustness	Bad	Very good
Using GPU	Certain implementations (not OpenCV)	Yes, if available
Using color	No	Yes

Fig. 2. comparison between MTCNN and Viola$_{jones}$

B. AttentionDenseNet Overview:

AttentionDenseNet builds upon DenseNet by incorporating attention mechanisms. This architecture focuses on relevant parts of the feature maps, enhancing the network's representational power by weighting important features more heavily.

Architecture:

Initial Convolution: A single convolutional layer, similar to DenseNet. Dense Blocks: Three dense blocks, similar to DenseNet. Transition Layers: Two transition layers, similar to DenseNet. Attention Blocks: Two attention blocks inserted after each transition layer to refine feature maps based on relevance. Global Pooling and Classifier: Includes batch normalization, ReLU activation, global average pooling, and a fully connected layer, similar to DenseNet.

C. Detailed Comparison

1. BasicBlock and BottleneckBlock:

Both architectures use these blocks within their dense blocks, contributing to dense connectivity. 2. TransitionBlock:

Fig. 3. Image of comparison between MTCNN and Viola$_{jone}$

Identical in both architectures, performing downsampling and reducing feature map dimensions. 3. AttentionBlock:

Unique to AttentionDenseNet, this block enhances features by focusing on the most relevant parts of the input. It uses convolutional layers followed by batch normalization, ReLU activation, and a sigmoid function to generate attention maps, which weight the input feature maps. 4. DenseBlock:

Both networks use DenseBlock, which consists of multiple BasicBlock or BottleneckBlock units. The DenseBlock concatenates the input with its output to ensure dense connectivity. 5. Overall Network Structure:

DenseNet: Follows a simple structure with dense blocks separated by transition blocks. AttentionDenseNet: Enhances DenseNet by adding attention blocks after transition layers to refine features based on relevance.

D. LessDenseNet Overview

LessDenseNet introduces variability by alternating block types (BasicBlock and BottleneckBlock) based on the layer index. This approach offers a different learning strategy compared to the uniform block type approach of DenseNet, catering to different trade-offs in computational efficiency and model expressiveness.

Key Differences:

1. Dense Connectivity:

DenseNet: Maintains dense connectivity within each dense block, where each layer receives inputs from all preceding layers. LessDenseNet: Introduces a less dense connectivity pattern by alternating between BasicBlock and BottleneckBlock within each dense block, based on the layer index. 2. Block Structure:

DenseNet: Uses either BasicBlock or BottleneckBlock consistently throughout its dense blocks. LessDenseNet: Alternates block types (BasicBlock and BottleneckBlock) within each dense block, introducing a different learning strategy. 3. Dropout Usage:

DenseNet: Applies dropout uniformly across all layers where specified. LessDenseNet: Introduces an additional dropout rate and alternates its application between BasicBlock and BottleneckBlock within each dense block.

E. SEDenseNet Overview

SEDenseNet incorporates Squeeze-and-Excitation (SE) blocks within each block (BasicBlock or BottleneckBlock), recalibrating channel-wise feature responses adaptively by modeling interdependencies between channels.

Key Differences:

1. Attention Mechanism:

SEDenseNet: Incorporates SE blocks within each block to enhance feature learning and discriminative capabilities. DenseNet: Does not include SE blocks or any explicit attention mechanism. 2. Block Structure and Connectivity:

SEDenseNet: Retains dense connectivity and uses BasicBlock or BottleneckBlock uniformly. DenseNet: Maintains dense connectivity with uniform block types. 3. Initialization and Normalization:

SEDenseNet: Uses Kaiming normal initialization for convolutional layers and applies batch normalization uniformly. DenseNet: Likely follows similar initialization and normalization practices.

F. SkipDenseNet Overview

SkipDenseNet extends DenseNet by introducing skip connections between different dense blocks and additional regularization (dropout). This enhances feature propagation and gradient flow through the network.

Key Differences:

1. Skip Connections:

SkipDenseNet: Introduces skip connections between dense blocks, enhancing feature propagation and gradient flow. DenseNet: Does not incorporate skip connections between dense blocks. 2. Block Structure and Connectivity:

SkipDenseNet: Utilizes BasicBlock or BottleneckBlock within dense blocks similarly to DenseNet, but with additional skip connections. DenseNet: Follows a traditional DenseNet architecture with dense connectivity within each dense block. 3. Dropout Usage:

SkipDenseNet: Applies dropout within skip connections to regularize the network. DenseNet: Typically uses dropout within individual blocks but not explicitly in skip connections. 4. Initialization and Normalization:

SkipDenseNet: Uses a normal distribution for initializing convolutional layers and applies batch normalization uniformly. DenseNet: Likely follows similar initialization and normalization methods

G. Comparison

You can see the comparison of different models in Table 1

Feature / Model	DenseNet	AttentionDenseNet	LessDenseNet	SkipDenseNet	SEDenseNet
Block Type	BasicBlock / BottleneckBlock	BasicBlock / BottleneckBlock	BasicBlock / BottleneckBlock	BasicBlock / BottleneckBlock	BasicBlock / BottleneckBlock
Dense Connectivity	Yes	Yes	Yes	Yes	Yes
Skip Connections	No	No	No	Yes	No
Dropout Usage	Yes, within blocks	Yes, within blocks	Yes, configurable	Yes, within SkipBlock	Yes, within blocks
Attention Mechanism	No	Yes, SE module	No	No	Yes, SE module
Transition Blocks	Yes	Yes	Yes	Yes	Yes
Layer Count	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Global Average Pooling	Yes	Yes	Yes	Yes	Yes
Batch Normalization	Yes	Yes	Yes	Yes	Yes
Initialization	Kaiming Normal	Kaiming Normal	Kaiming Normal	Kaiming Normal	Kaiming Normal
Model Depth	Variable	Variable	Variable	Variable	Variable
Model Specifics	Modified, focus on comparison	Incorporates attention mechanism	Optimized for fewer parameters	Enhanced with skip connections	Incorporates attention mechanism

Fig. 4. copmaring different models

5. RELATED WORKS

Most of the existing research in the area of automatic facial expression analysis focuses on the following three topics: (i) Categorical model: Assigning discrete emotion category labels, (ii) FACS model: Detecting the presence/absence (and the strength) of various action units defined by FACS [12], and (iii) Dimensional model: Describing emotions using two or three dimensional models such as valence-arousal [45], pleasure-arousal-dominance [38], etc. Summarizing the vast amount of existing research on these topics is beyond the scope of this paper and we refer the readers to [27, 30, 35] for recent surveys on these topics. Expression datasets: Several facial expression datasets have been created in the past that consist of face images labeled with discrete emotion categories [4, 9, 10, 11, 16, 17, 31, 34, 40, 41, 43, 54, 55], facial action units [4, 34, 36, 37, 43], and strengths of valence and arousal [25, 27, 28, 40, 44]. While these datasets played a significant role in the advancement of automatic facial expression analysis in terms of emotion recognition, action unit detection and valence-arousal estimation, they are not the best fit for learning a compact expression embedding space that mimics human visual preferences. Expression embedding: A neural network was trained in [39] using an emotion classification dataset and category label-based triplet loss [46] to produce a 128-dimensional embedding, which was combined with an LSTM-based network for animating three basic expressions. Emotion labels do not provide information about within-class variations and hence a network trained with label-based triplets may not encode fine-grained expression information. The proposed FEC dataset addresses this issue by including expression comparison annotations for within-class triplets. A self-supervised approach was proposed in [26] to learn a 256-dimensional facial attribute embedding by watching videos, and the learned embedding was used for multiple tasks such as head pose estimation, facial landmarks prediction, and emotion recognition by training an additional classification or regression layer using labeled training data. However, as reported in [26], its performance is worse than existing approaches on these tasks. Different from [26], we follow a fully-supervised approach for learning a compact (16-dimensional) expression embedding. Triplet loss-based representation learning: Several existing works have used triplet-based loss functions for learning image representations. While majority of them use category label-based triplets [14, 19, 20, 32, 46, 48, 51, 56], some existing works [5, 50] have focused on learning fine-grained representations. While [50] used a similarity measure computed using several existing feature representations to generate groundtruth annotations for the triplets, [5] used textimage relevance based on Google image search to annotate the triplets. Different from these approaches, we use human raters to annotate the triplets. Also, none of these works focus on facial expressions.

6. PLOTS

Here are some plots for comparing different models:

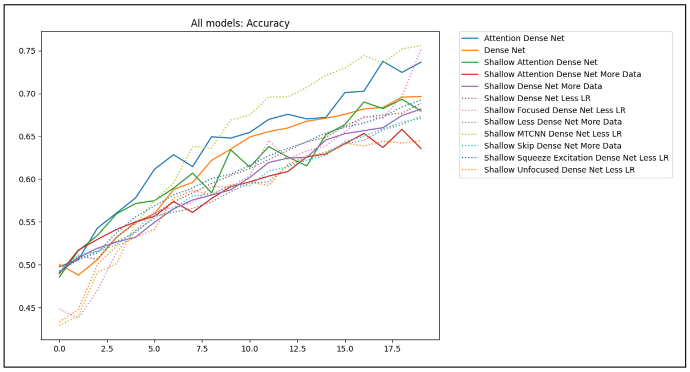

Fig. 5. all models accuracy

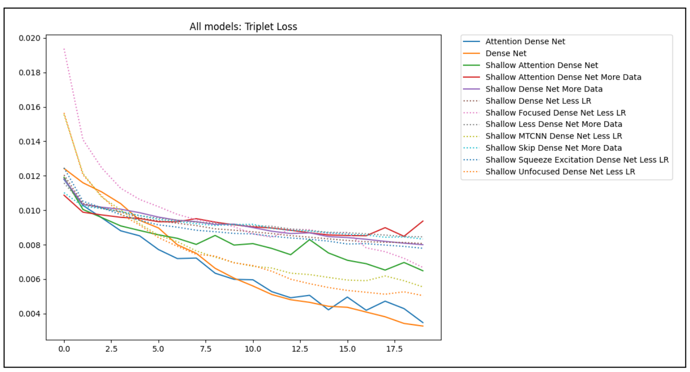

Fig. 6. all models loss.

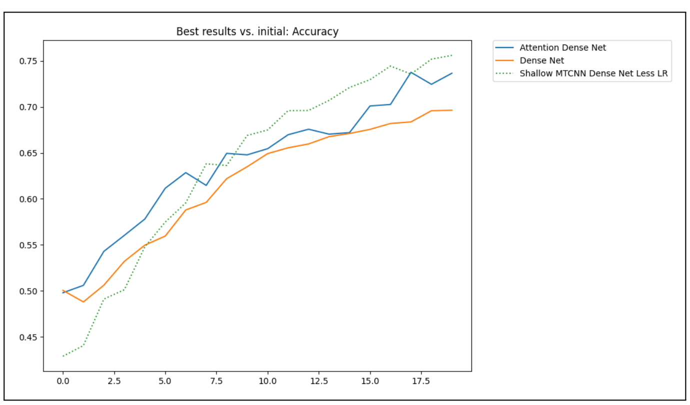

Fig. 7. best results accuracy

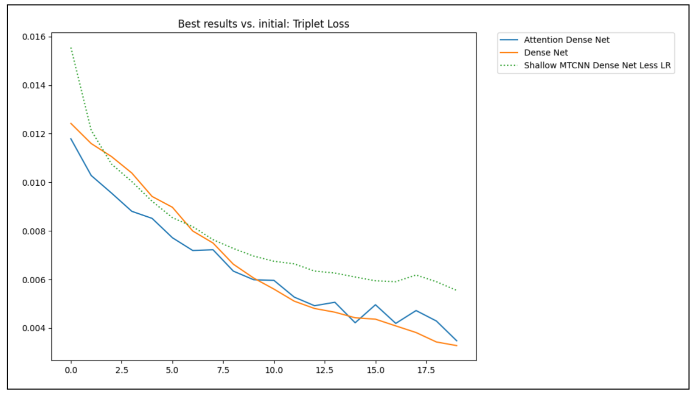

Fig. 8. best results loss.

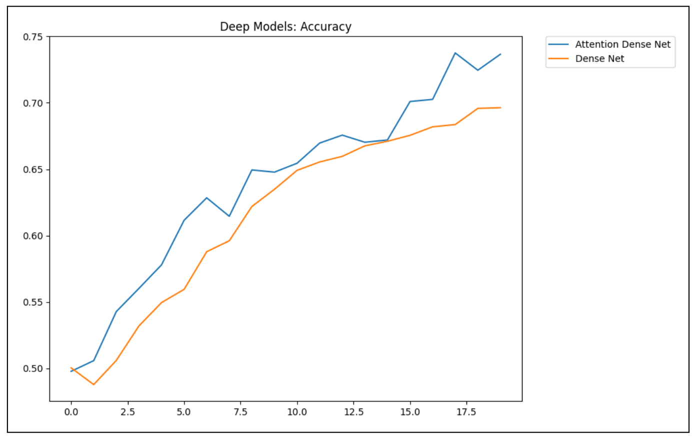

Fig. 9. deep models accuracy

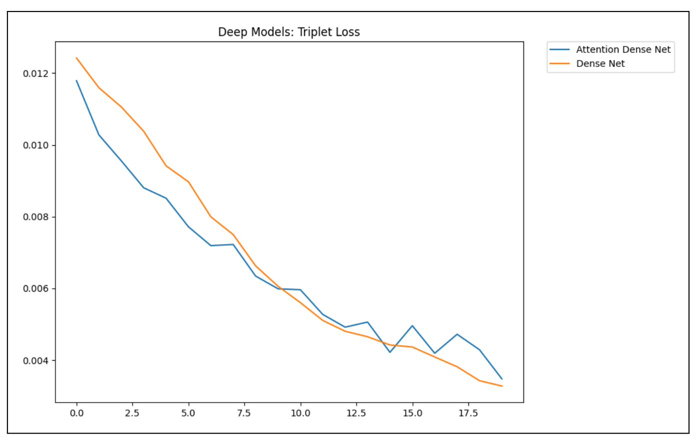

Fig. 10. deep models loss.

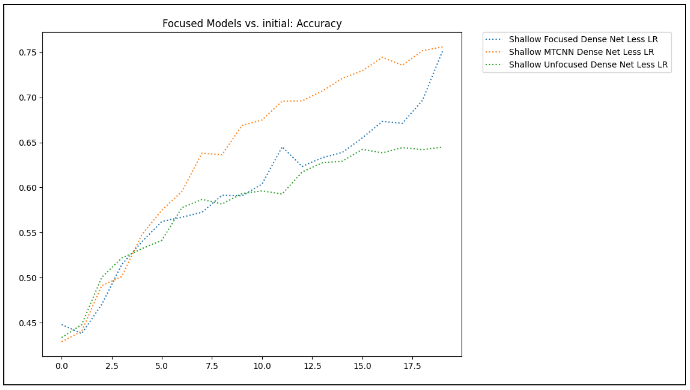

Fig. 11. focused models accuracy

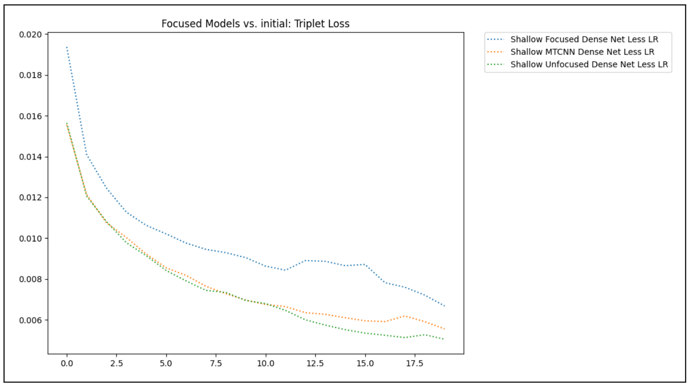

Fig. 12. focused models loss.

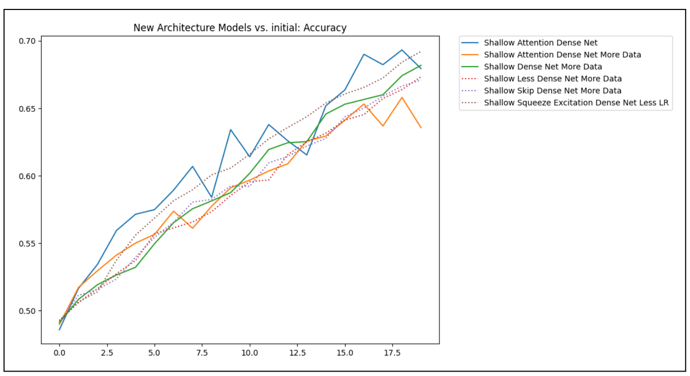

Fig. 13. new architecture models accuracy

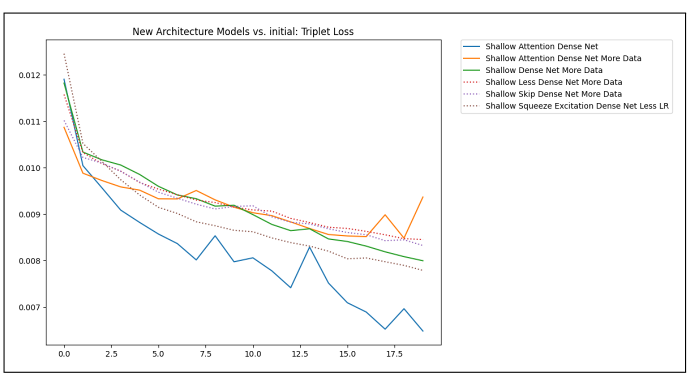

Fig. 14. new architecture models loss.

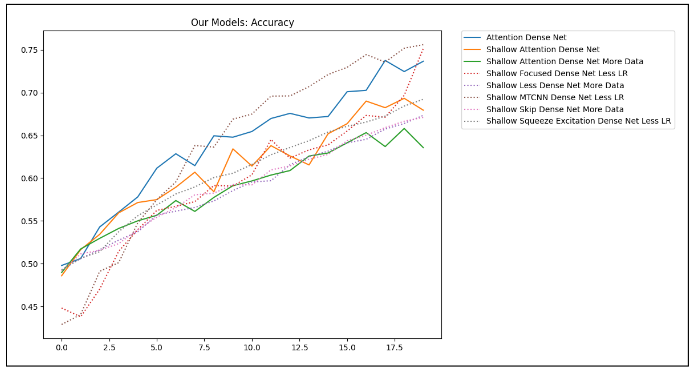

Fig. 15. our accuracy

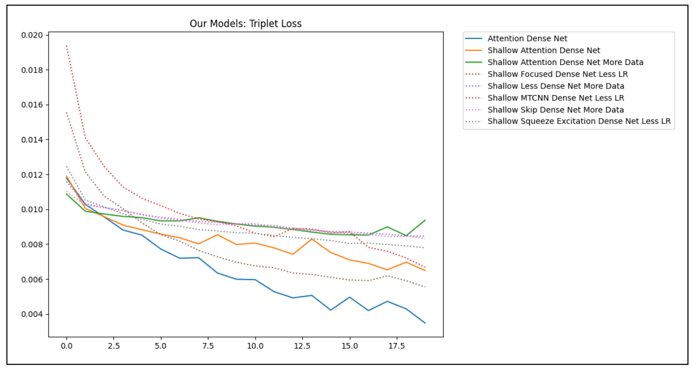

Fig. 16. our models loss.

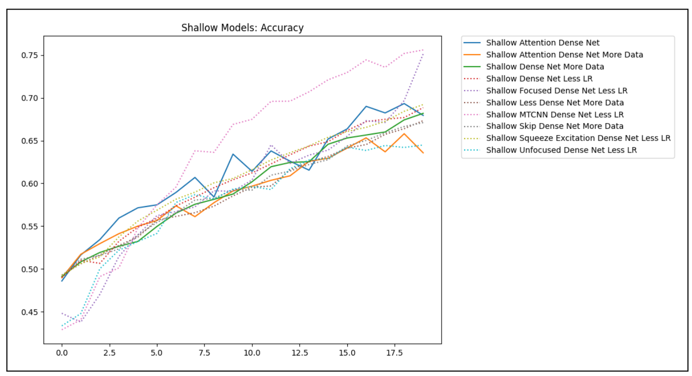

Fig. 17. shallow models accuracy

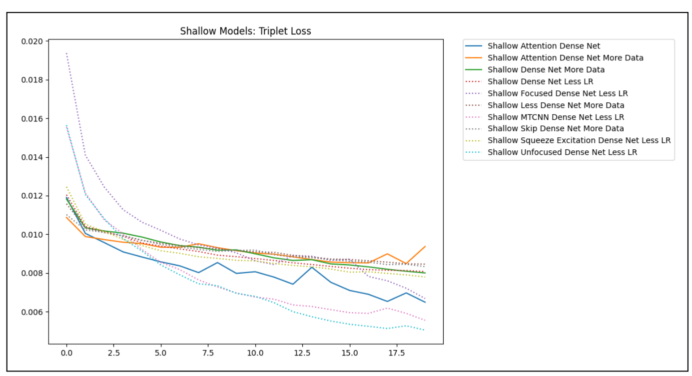

Fig. 18. shallow models loss.

References

[1] MegaFace challenge leaderboard: http://megaface.cs.washington.edu/results/facescrub.html. 4

[2] MS-Celeb-1M challenge leaderboard: http://www.msceleb.org/leaderboard/iccvworkshop-c1. 4

[3] Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 6, 9

[4] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martínez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *CVPR*, 2016. 2

[5] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135, 2010. 2

[6] C. A. Corneanu, M. Madadi, and S. Escalera. Deep structure inference network for facial action unit recognition. In *ECCV*, 2018. 5

[7] A. S. Cowen and D. Keltner. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *Proceedings of the National Academy of Sciences*, 114(38):E7900E7909, 2017. 3

[8] A. S. Cowen and D. Keltner. Clarifying the conceptualization, dimensionality, and structure of emotion. *Trends in Cognitive Sciences*, 22(4):274–276, 2018. 3

[9] A. Dhall, R. Goecke, S. Ghosh, J. Joshi, J. Hoey, and T. Gedeon. From individual to group-level emotion recognition: Emotiw 5.0. In *ICMI*, 2017. 2

[10] A. Dhall, R. Goecke, J. Joshi, M. Wagner, and T. Gedeon. Emotion recognition in the wild challenge 2013. In *ICMI*, 2013. 2

[11] A. Dhall, O. V. R. Murthy, R. Goecke, J. Joshi, and T. Gedeon. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *ICMI*, 2015. 2

[12] P. Ekman, W. V. Friesen, and J. C. Hager. *Facial Action Coding System - Manual*. A Human Face, 2002. 1, 2

[13] J. Fiss, A. Agarwala, and B. Curless. Candid portrait selection from video. *ACM Transactions on Graphics*, 30(6):128:1–128:8, 2011. 1

[14] W. Ge, W. Huang, D. Dong, and M. R. Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, 2018. 2

[15] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 5

[16] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. C. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. T. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64:59–63, 2015. 2

[17] R. Gross, I. A. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 2010. 2

[18] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016. 4

[19] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai. Triplet-center loss for multi-view 3D object retrieval. In *CVPR*, 2018. 2

[20] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 2

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 4

[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 4

[23] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The MegaFace benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016. 4

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[25] S. Koelstra, C. Mühl, M. Soleymani, J. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. DEAP: A database for emotion analysis using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, 2012. 2

[26] A. S. Koepke, O. Wiles, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *BMVC*, 2018. 2

[27] D. Kollias, P. Tzirakis, M. A. Nicolaou, A. Papaioannou, G. Zhao, B. W. Schuller, I. Kotsia, and S. Zafeiriou. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *CoRR*, abs/1804.10938, 2018. 2

[28] J. Kossaifi, G. Tzimiropoulos, S. Todorovic, and M. Pantic. AFEW-VA database for valence and arousal estimation in-the-wild. *Image and Vision Computing*, 65:23–36, 2017. 2

[29] A. Krizhevsky. Convolutional deep belief networks on CIFAR-10. *Unpublished manuscript*, 2010. 4

[30] S. Li and W. Deng. Deep facial expression recognition: A survey. *CoRR*, abs/1804.08348, 2018. 1, 2

[31] S. Li and W. Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing*, 28(1):356–370, 2019. 2

[32] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *CVPR*, 2016. 2

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 6

[34] P. Lucey, J. F. Cohn, T. Kanade, J. M. Saragih, Z. Ambadar, and I. A. Matthews. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *CVPR Workshops*, 2010. 2

[35] B. Martinez, M. F. Valstar, B. Jiang, and M. Pantic. Automatic analysis of facial actions: A survey. *IEEE Transactions on Affective Computing*, 2017. 1, 2

[36] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn. DISFA: A spontaneous facial action inten-

sity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013. 2, 5

[37] D. McDuff, R. E. Kaliouby, T. Senechal, M. Amr, J. F. Cohn, and R. W. Picard. Affectiva-MIT facial expression dataset (AM-FED): Naturalistic and spontaneous facial expressions collected in-the-wild. In *CVPR Workshops*, 2013. 2

[38] A. Mehrabian. *Basic dimensions for a general psychological theory: Implications for personality, social, environmental, and developmental studies*. Oelgeschlager, Gunn & Hain, 1980. 2

[39] H. Meng, T. Lin, X. Jiang, Y. Lu, and J. Wen. LSTM-based facial performance capture using embedding between expressions. *CoRR*, abs/1805.03874, 2018. 2

[40] A. Mollahosseini, B. Hassani, and M. H. Mahoor. AffectNet: A database for facial expression, valence, and arousal computing in the wild. *CoRR*, abs/1708.03985, 2017. 2, 3, 5, 8

[41] A. Mollahosseini, B. Hassani, M. J. Salvador, H. Abdollahi, D. Chan, and M. H. Mahoor. Facial expression recognition from world wild web. In *CVPR Workshops*, 2016. 2

[42] A. Nech and I. Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *CVPR*, 2017. 4

[43] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *ICME*, 2005. 2

[44] F. Ringeval, A. Sonderegger, J. S. Sauer, and D. Lalanne. Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. In *FG*, 2013. 2

[45] J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. 2

[46] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2, 3, 4, 5

[47] Z. Shao, Z. Liu, J. Cai, and L. Ma. Deep adaptive attention for joint facial action unit detection and face alignment. In *ECCV*, 2018. 5

[48] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 2

[49] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 4

[50] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[51] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. In *ICCV*, 2017. 2

[52] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 3

[53] O. Wiles, A. S. Koepke, and A. Zisserman. Self-supervised learning of a facial attribute embedding from video. In *BMVC*, 2018. 5, 8

[54] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. From facial expression recognition to interpersonal relation prediction. *International Journal of Computer Vision*, 126(5):550–569, 2018. 2

[55] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen. Facial expression recognition from near-infrared videos. *Image and Vision Computing*, 29(9):607–619, 2011. 2

[56] B. Zhuang, G. Lin, C. Shen, and I. D. Reid. Fast training of triplet-based deep binary embedding networks. In *CVPR*, 2016. 2