Facial Expression Recognition in the Wild - Deep Learning Model Analysis

Damith Chamalke Senadeera
Queen Mary University of London
d.c.senadeera@se21.qmul.ac.uk

1. Introduction

The facial expression recognition in the wild has become an important research topic as it plays a pivotal role in intelligent social interaction. As discussed in the previous assignment there are various restrictions faced while trying to perform facial expression recognition in real world settings such as substantial inter-class similarity, little intra-class similarity and different occlusion and pose variations in the captured images, etc. [25]. Avoiding these obstacles researchers have tried to train machine learning models in a supervised setting to recognize facial expressions in the wild using different techniques and recently deep learning-based techniques have gained much popularity due to its superior performances given a good amount of proper data.

Deviating a little on the data preparation step reported in the assignment 1, I divided the main data set utilized which has been generated synthetically using the methods mentioned in [12], [19], [11] and [22] based on an image set obtained from Aff-Wild2 database [10], [21], [14], [16], [20], [18], [15], [17], [26], [13] into 3 parts namely, the training, validation and testing sets which will consist of 70%, 20% and 10% of the main data set respectively as reported in table 1.

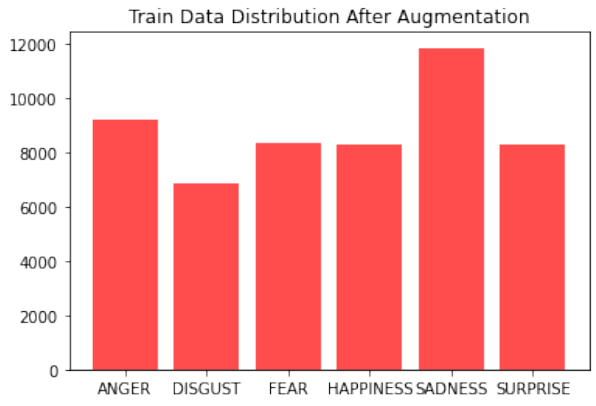

Figure 1. Training Data Distribution after Augmentation

Afterwards, all the pre-processing techniques which were mentioned in the Assignment 1 were applied and the training set was augmented according to the steps mentioned in the Assignment 1 per each class in the training set and the final augmented training data statistics can be observed in the figure1.

Text Label	No. of Images in training set	No. of Images in validation set	No. of Images in testing set
Anger	5759	1646	823
Disgust	1721	491	246
Fear	2088	597	298
Happiness	6379	1823	911
Sadness	9128	2608	1304
Surprise	5192	1483	742
Total	**30267**	**8648**	**4324**

Table 1. Training, Validation and Testing Set Image Distribution

2. Task 1: Main Machine Learning Model

As the main machine learning model, the Vision Transformer (ViT) Model [3] was selected as this is the state-of-the-art deep learning model which gave the best performance reported in the past work for the available standard facial expression recognition databases related to deep learning models according to the literature review done before.

The Vision transformer is a deep learning model which leverages attention mechanisms where significance of each component of the input data are weighed differentially [3]. The basic architectures of a Vision transformer model [5] is given in the figure 2.

In general, how the vision transformer works is that the input image is split into patches and those patches are flattened and used to create a lower dimensional linear embedding which after being incorporated with a positional embedding is inputted to the transformer encoder to obtain the required predictions.

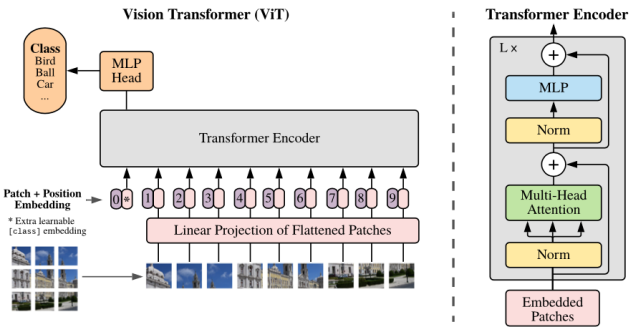

Figure 2. Vision Transformer Architecture

As the main model I have chosen a pretrained vision transformer model "vit-base-patch16-224-in21k" obtained from the Hugging Face Transformers Library [8] which was originally released by google research [3] consisting of 12 ViT layers which has been pre-trained on ImageNet-21k [23] data set consisting of 14 million images across 21 thousand classes where each image is of resolution 224*224 which were in turn divided into 16*16 fixed-size patches. The final fully connected layer of this pretrained model was replaced with a liner layer to output the 6 class labels relevant to the current dataset to perform transfer learning.

As the loss function, I used the PyTorch default Cross Entropy loss function but with class weights calculated according to the data distribution for each class label in the training data set to have a class balanced loss because still the training data set is somewhat imbalanced.

The images in the training set were pre-processed using the ViT feature extractor relevant to this pretrained model (to resize the images into 224*224 resolution and extract the 16*16 patches accordingly) on the fly and the model was trained using the Adam Optimizer starting with an initial learning rate of 0.001 and a batch size of 256 images per batch where all the learnable model parameters of the network were unfrozen and set to train.

As the main metric to measure the model's performance, F1 score was selected due to the imbalance data in the training set and after each epoch the F1 measure of the Validation set was checked and if the validation F1 measure has decreased more than 10% in the current epoch compared to the previous epoch, the learning rate was reduced by 20%.

In this manner the pre-trained ViT model was trained with the new training data set for 3 epochs and the results were evaluated and reported in table 2. The change of training loss, accuracy and F1 scores throughout training is plotted in the figure 3 .

The training and validation F1 score both has reached 99% and afterwards the trained model was evaluated on the separated testing set and results are reported in the table 3. It seems since the ViT model is pre-trained it's able to attain

Epoch	Training Loss	Training Accuracy (%)	Training F1 Score (Average) (%)	Val. Accuracy (%)	Val. F1 Score (Average) (%)
1	0.305	89.4	88.8	97.4	97.4
2	0.039	98.8	98.8	99.0	98.9
3	0.029	99.0	99.1	99.0	99.0

Table 2. Training details of the main model

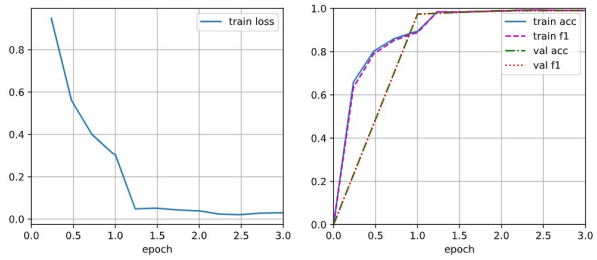

Figure 3. Loss and Metric Curves for the main model

Testing Accuracy (%)	Testing F1 Score (Average) (%)
99.0	99.0

Table 3. Testing details of the main model

Class	Test Accuracy (%)	Test F1 Score (%)
Anger	99.8	99.1
Disgust	98.8	98.1
Fear	99.7	99.8
Happiness	99.8	98.9
Sadness	98.8	99.2
Surprise	97.4	98.6

Table 4. Testing metrics for each class for the main model

this Accuracy and F1 Score with a small number of epochs. The individual Accuracies and F1 scores for each class label were also computed and can be observed in table 4. The confusion matrix related to this also can be observed in figure 4

It is observed that all the classes learn almost equally as the F1 scores and accuracies for each class labels are almost the same. Out of the class specific F1 scores, "Disgust" class has the lowest F1 score of 98.1%, which is just only 0.9% less than the average overall Test F1 score. Although

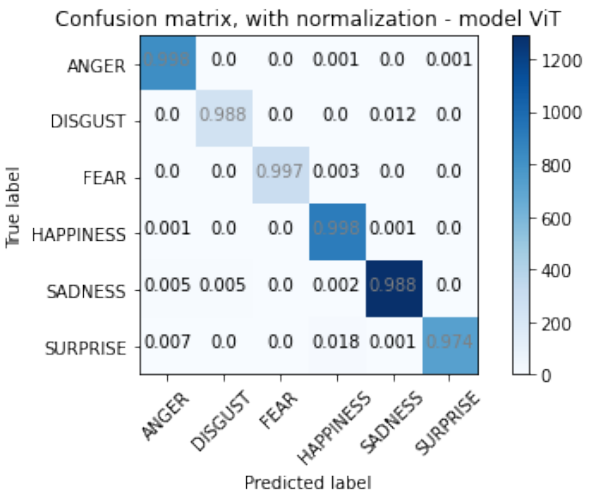

Figure 4. Confusion Matrix for testing data for the main model

it's not a very significant deviation, this might be due to the fact that "Disgust" had the least number of examples in the data set and due to the use of data augmentation and use of class balanced loss in training the effect might have been alleviated up to this extent.

3. Task 2: Baseline Machine Learning Model

As the baseline model, the ResNet-50 Architecture [7] was chosen as this Convolutional Neural Network architecture has also performed very well according to the literature review conducted in the previous assignment for facial expression recognition. Resnet-50 is a variant of the Resnet Architecture which constitutes of skip connections between layers in order to address the degradation problem when going deep with stacking layers, has 48 Convolution layers plus 1 MaxPool and 1 Average Pool layer [7].

In ResNet-50, skip connections are placed after 3 convolutional layers and the overall architecture can be observed in the figure 5.

As the baseline model I have chosen the pretrained ResNet-50 model from PyTorch - Torchvision Library [24] which has been pre-trained on ImageNet-1k [2] data set consisting of 1 million images across 21 thousand classes. The final fully connected layer of this pretrained model was replaced with a liner layer to output the 6 class labels relevant to the current dataset to perform transfer learning.

Similar to the main model, as the loss function, I used the PyTorch default Cross Entropy loss function but with class weights calculated according to the data distribution for each class label in the training data set to have a class balanced loss. The model was trained using the Adam Optimizer starting with an initial learning rate of 0.001 and a batch size of 256 images per batch where all the learnable

layer name	50-layer
conv1	$7 \times 7, 64$, stride 2
	3×3 max pool, stride 2
conv2_x	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	average pool, 2048-d fc

Figure 5. ResNet-50 Architecture

Epoch	Training Loss	Training Accuracy (%)	Training F1 Score (Average) (%)	Val. Accuracy (%)	Val. F1 Score (Average) (%)
1	0.855	69.1	66.7	78.6	78.4
2	0.443	85.4	85.1	87.1	88.0
3	0.316	89.4	89.1	89.2	89.8

Table 5. Training details of the baseline model

model parameters of the network were unfrozen and set to train.

As the main metric to measure the model's performance, F1 score was selected again due to the imbalance data in the training set and after each epoch the F1 measure of the Validation set was checked and if the validation F1 measure has decreased more than 10% in the current epoch compared to the previous epoch, the learning rate was reduced by 20%.

In this manner the pre-trained ResNet-50 model was trained with the new training data set for 3 epochs keeping the hyper-parameters as same as the main model as much as possible and the results were evaluated and reported in table 5. The change of training loss, accuracy and F1 scores throughout training is plotted in the figure 6.

The training and validation F1 scores both in this baseline has reached only 89% and afterwards the trained model was evaluated on the separated testing set and results are re-

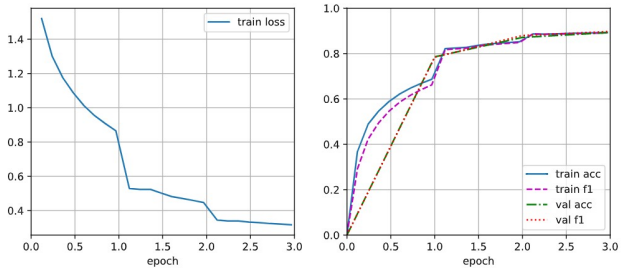

Figure 6. Loss and Metric Curves for the baseline model

Testing Accuracy (%)	Testing F1 Score (Average) (%)
89.5	89.9

Table 6. Testing details of the baseline model

Class	Test Accuracy (%)	Test F1 Score (%)
Anger	86.8	92.4
Disgust	91.1	95.1
Fear	90.3	85.5
Happiness	89.0	86.7
Sadness	88.3	88.8
Surprise	94.7	91.2

Table 7. Testing metrics for each class for the baseline model

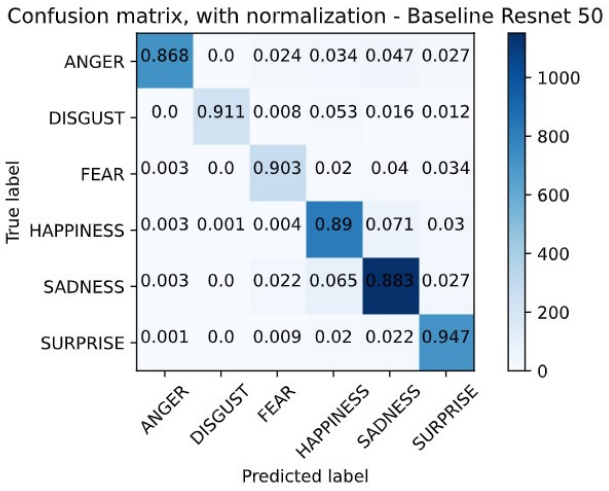

Figure 7. Confusion Matrix for testing set for the baseline model

Exp.	No. of Epochs	Test Accuracy (%)	Test F1 Score (%)
1	2	98.7	98.8
2	3	99.0	99.0
3	4	98.3	97.9
4	5	98.0	97.7

Table 8. Ablation for different epochs

ported in the table 6. In comparison with the main model with the similar training hyper-parameters, it seems the ResNet-50 model gives almost 10The individual Accuracies and F1 scores for each class label were also computed for the baseline model and can be observed in table 7. The confusion matrix related to this also can be observed in figure 7.

In general, the F1 scores for all the classes are lower than the test results from the main model and Class "Fear" has given the least F1 score of 85.5%. This might be due to the hyper-parameters used or else also due to the model's capacity to learn given the training dataset.

4. Task 3: Ablation Study

Since the main model performed well given the baseline model, the main model which used a ViT is selected for the ablation study. As the 3 hyper-parameters to tune in the model, number of epochs to train the model, Optimizer for the model and learning rate for the model were chosen.

Firstly, 4 experiments were conducted to see the test performance by varying the number of epochs the model is trained for. For these 4 experiments, the batch size was kept at 256 images per batch with Adam optimizer and a starting learning rate of 0.001 was used where all the learnable model parameters of the network were unfrozen and set to train. For each experiment accuracy and the F1 score for the testing data set was reported in the table 8

The 4 experiments with 4 different number of epochs showed that after 3 epochs the model's F1 score and accuracy didn't improve, but in turn it seems they tend to reduce somewhat. This might happen due to the reason that the ViT model is already pre-trained and when training for more epochs the pretrained knowledge might be getting distorted.

Afterwards, 3 different optimizers were chosen to experiment with namely, SGD (Stochastic gradient descent) which is the basic optimizer used to update the loss in neural networks [1], RMSProp (Root Mean Square Propagation) in which the learning rate is adapted to each of the parameters in the neural network [6] and Adam (Adaptive Moment Estimation) which is an updated version of RMSProp with momentum [9].

Along with them 3 different learning rates of 0.01, 0.001 and 0.00001 were tested. These learning rates were chosen to study how the different optimizers selected will behave

	Optimizer - SGD	Optimizer - RMSProp	Optimizer - Adam
Learning rate - 0.01	98.5 / 98.1	41.0 / 35.5	37.0 / 38.3
Learning rate - 0.001	74.8 / 70.1	72.2 / 72.9	99.0 / 99.0
Learning rate - 0.00001	14.2 / 7.5	99.5 / 99.5	99.7 / 99.7

Table 9. Testing metrics (Accuracy / F1 Score) for the ablation of Optimizers vs Learning Rates

when trained in a higher learning rate to a lower learning rate. The batch size was kept constant at 256 images per batch in all the experiments where all the learnable model parameters of the network were unfrozen and set to train and the models were trained for 3 epochs. The results obtained for these 9 experiments on the testing set (test accuracy % / test F1 score %) are tabulated in the table 9 and visualized in the figure 8

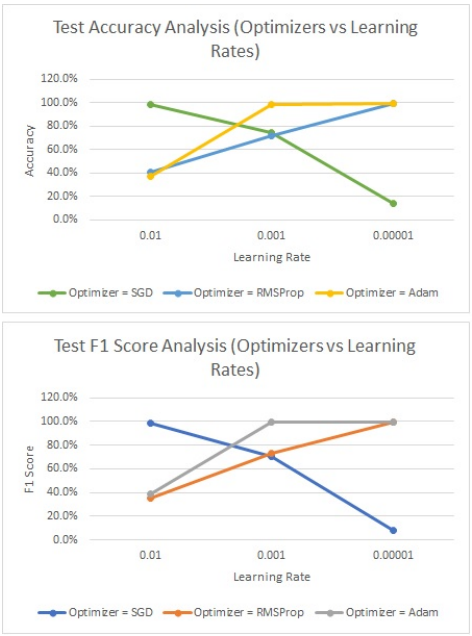

Figure 8. Ablation for Optimizers vs Learning Rates

From these experiments we can clearly see that when the learning rate for RMSProp and Adam optimizers decreases the accuracy and F1 score increased drastically and in contrast with SGD when the learning rate decreased the accuracy and the F1 score decreased. This was expected as for SGD when there is a low learning rate it should run a higher number of epochs to achieve the same performance as compared to a higher learning rate according to [1].

Exp	Learning Rate	Test Accuracy (%)	Test F1 Score (Average) (%)
1	0.00001	99.7	99.7
2	0.00002	99.8	99.7
3	0.00003	99.8	99.8

Table 10. Ablation for different small learning rates

Class	Test Accuracy (%)	Test F1 Score (%)
Anger	99.8	99.9
Disgust	99.2	99.5
Fear	100.0	99.9
Happiness	99.7	99.8
Sadness	99.9	99.8
Surprise	99.9	99.8

Table 11. Testing metrics for each class for the best model

Since Adam Optimizer with a learning rate of 0.0001 achieved the best testing F1 score and accuracy, keeping all the other hyper-parameters the same, 3 experiments were conducted to see how the model behaves in different lower learning rates of 0.00001, 0.00002 and 0.00003 for the Adam optimizer. The results obtained for these 3 experiments on the testing set (test accuracy % / test F1 score %) are tabulated in the table 10.

With these 3 experiments not much of a variation was observed in the testing F1 Score, but since the experiment with learning rate 0.00003 gave the best reported test F1 score of 99.8% out of all the conducted experiments this model was selected to be tested with new production data.

So according to the ablation study the best performing model was found out to be the ViT pre-trained model, later trained with Adam optimizer and with an initial learning rate of 0.00003 and a batch size of 256 images per batch where all the learnable model parameters of the network were unfrozen and set to train and which was trained for 3 epochs. The individual Accuracies and F1 scores for each class label were also computed for best performing model and can be observed in the table 11.

It seems that in the best case, predictions related to all the classes are performing in a near perfect manner.

5. Task 4: Production Data

The new production test set consists of 26,124 images distributed over the 6 basic facial expressions classes [4]

which the initial data set itself consisted of. The new production test set is also highly unbalanced and the image distribution over each class is depicted in the figure 9. Therefore, the main evaluation metric to be used in the analysis here also will be the F1 score.

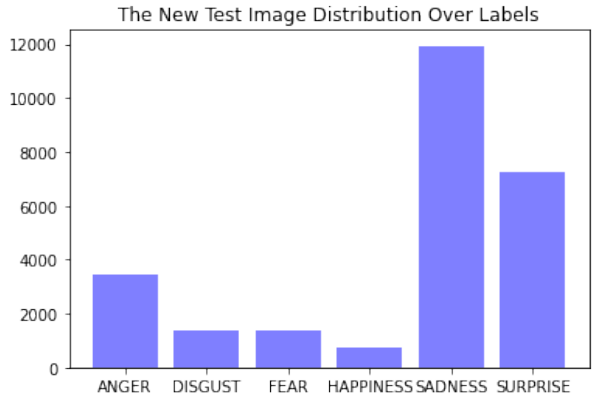

Figure 9. Production Data Distribution over Class Labels

The same pre-processing steps of face alignment, resizing and normalization was performed on this production test and it was noted that around 5% of the images in the production set were unable to be aligned as the facial landmarks were not identified using the MTCNN [27] algorithm (which was used in the Assignment 1 as well for alignment) in contrast where in the training set only around 3% of the images were unable to be aligned.

The ViT model identified during the ablation study in terms of the best average F1 Score of 99.8% for the testing set is used to test with the new production data set and the results are reported in the table 12.

Testing Accuracy (%)	**Testing F1 Score (Average) (%)**
97.1	94.6

Table 12. Testing details of the best model with production data

In the overall average F1 score there is a 5.2% decrease and in the Accuracy, there is a 2.7% decrease when this best performing model is tested with the new production data set. The individual Accuracies and F1 scores for each class label in the production data set were also computed and reported in table 13.

The largest decrease in F1 score can be observed in the "Disgust" class which reports 84.7% , depicting a 14.8% decrease compared to the testing F1 score with the original testing set. This is also the class which had the least number of data samples in the training set before augmentation. Therefore, a probable cause for this degradation of performance in this specific class might have been, when the data

Class	**Test Accuracy (%)**	**Test F1 Score (%)**
Anger	98.8	99.0
Disgust	85.6	84.7
Fear	99.6	94.3
Happiness	99.7	93.1
Sadness	96.9	97.6
Surprise	98.2	98.9

Table 13. Testing metrics for each class for the best model with production data

was augmented to increase the class size a data shift in the form of a covariate shift might have occurred introducing an undesired bias.

Also, when manually inspecting the production testing data set, it was observed that a vast majority of the face images are female images where as in in the training set the majority of the images present were male face images except in the "Happiness" class. So, the model might have learned a bias towards male face images in an unfair manner in this case for the specific classes.

This is also visible in the reported metrics for the specific classes in the production test set in table 13 where "Happiness" class has scored 99.7% accuracy. But the F1 score for "Happiness" class has again reduced to 93.1% because the female images from other classes might have got misclassified in the "Happiness" class as it seemed to contain female images at a considerable level in the training set. Also the production data set seem to contain an increased number of face images with occlusion variations (e.g.- hand in front of the face, some random text on the face, etc.) compared to the training set.

To mitigate this performance degradation in production test data set, the algorithmic biases learned by the model could be cleared by identifying the learned biases with respect to attributes such as gender and re-balancing the training data set with respect to those identified attributes causing the biases. Also, for the training set, the augmentation strategy could be tailored to incorporate the production of balanced data with respect to the identified attributes which may cause biases. Also, the model can be retrained periodically incorporating new data which is balanced and includes occlusion variations such as found in the new production test set if a data shift is identified on the run.

By following the above mentioned methodologies, it should be able to mitigate the performance degradation issues for new production data.

References

[1] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993. 4,

5

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arxiv 2020. *arXiv preprint arXiv:2010.11929*, 2010. 1, 2

[4] Paul Ekman. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16, 1999. 5

[5] google research. Google-research/vision-transformer. https://github.com/google-research/vision_transformer. 1

[6] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] Hugging-Face. Google/vit-base-patch16-224-in21k-hugging-face. https://huggingface.co/google/vit-base-patch16-224-in21k. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[10] Dimitrios Kollias. Abaw: Valence-arousal estimation, expression recognition, action unit detection & multi-task learning challenges. *arXiv preprint arXiv:2202.10659*, 2022. 1

[11] Dimitrios Kollias, Shiyang Cheng, Maja Pantic, and Stefanos Zafeiriou. Photorealistic facial synthesis in the dimensional affect space. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[12] Dimitrios Kollias, Shiyang Cheng, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Deep neural network augmentation: Generating faces for affect analysis. *International Journal of Computer Vision*, 128(5):1455–1484, 2020. 1

[13] Dimitrios Kollias, Mihalis A Nicolaou, Irene Kotsia, Guoying Zhao, and Stefanos Zafeiriou. Recognition of affect in the wild using deep neural networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1972–1979. IEEE, 2017. 1

[14] D Kollias, A Schulc, E Hajiyev, and S Zafeiriou. Analysing affective behavior in the first abaw 2020 competition. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, pages 794–800. 1

[15] Dimitrios Kollias, Viktoriia Sharmanska, and Stefanos Zafeiriou. Face behavior a la carte: Expressions, affect and action units in a single network. *arXiv preprint arXiv:1910.11111*, 2019. 1

[16] Dimitrios Kollias, Viktoriia Sharmanska, and Stefanos Zafeiriou. Distribution matching for heterogeneous multi-task learning: a large-scale face study. *arXiv preprint arXiv:2105.03790*, 2021. 1

[17] Dimitrios Kollias, Panagiotis Tzirakis, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, Björn Schuller, Irene Kotsia, and Stefanos Zafeiriou. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision*, pages 1–23, 2019. 1

[18] Dimitrios Kollias and Stefanos Zafeiriou. Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface. *arXiv preprint arXiv:1910.04855*, 2019. 1

[19] Dimitrios Kollias and Stefanos Zafeiriou. Va-stargan: Continuous affect generation. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 227–238. Springer, 2020. 1

[20] Dimitrios Kollias and Stefanos Zafeiriou. Affect analysis in-the-wild: Valence-arousal, expressions, action units and a unified framework. *arXiv preprint arXiv:2103.15792*, 2021. 1

[21] Dimitrios Kollias and Stefanos Zafeiriou. Analysing affective behavior in the second abaw2 competition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3652–3660, 2021. 1

[22] Andreas Psaroudakis and Dimitrios Kollias. Mixaugment & mixup: Augmentation methods for facial expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2375, 2022. 1

[23] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 2

[24] torchvision. Resnet50-torchvision. https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html. 3

[25] Fanglei Xue, Qiangchang Wang, and Guodong Guo. Transfer: Learning relation-aware facial expression representations with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3601–3610, 2021. 1

[26] Stefanos Zafeiriou, Dimitrios Kollias, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, and Irene Kotsia. Aff-wild: Valence and arousal 'in-the-wild'challenge. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1980–1987. IEEE, 2017. 1

[27] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 6