Facial Action Unit Recognition in the Wild with Multi-Task CNN Self-Training for the EmotioNet Challenge

Philipp Werner, Frerk Saxen, and Ayoub Al-Hamadi
Neuro-Information Technology Group, Otto von Guericke University Magdeburg, Germany
{Philipp.Werner, Frerk.Saxen, Ayoub.Al-Hamadi}@ovgu.de

Abstract

Automatic understanding of facial behavior is hampered by factors such as occlusion, illumination, non-frontal head pose, low image resolution, or limitations in labeled training data. The EmotioNet 2020 Challenge addresses these issues through a competition on recognizing facial action units on in-the-wild data. We propose to combine multi-task and self-training to make best use of the small manually / fully labeled and the large weakly / partially labeled training datasets provided by the challenge organizers. With our approach (and without using additional data) we achieve the second place in the 2020 challenge – with a performance gap of only 0.05% to the challenge winner and of 5.9% to the third place. On the 2018 challenge evaluation data our method outperforms all other known results.

1. Introduction

The challenge was run on the EmotioNet database [2], which comprises (1) a *training set* of about 944k samples, which were automatically labeled with 12 facial Action Units (AUs), (2) an optimization set (*opt set*) of about 25k samples, which were manually labeled with 23 AUs – the same AUs that appear in the test set (listed in Section 3) –, and (3) a *validation* and a *test set* of about 107k and 218k images respectively, which were manually labeled with the 23 AUs and used to evaluate the approaches of the challenge participants. Each participant had five submissions on the validation and one submission on the test set. The used performance measure, called *final ranking score*, is the mean of the accuracy and the F1-score.

Our approach for recognizing AUs involves two ideas that are novel in this context: (1) **Multi-task learning**, which here means using two output neurons per AU, one for each of the training subsets. Even if labels of two training subsets have the same intended meaning, like a specific AU, there may be labeling biases or differences in labeling quality, especially if some data have been labeled by an algorithm. Using multi-task learning may help to better cope with these issues and still benefit from all available data. (2) **Self-training** [9] means that a teacher model is trained on a labeled dataset and used to predict pseudo-labels on a larger unlabeled (or in our case weakly / partially labeled) dataset. Afterwards, a student model is trained using both datasets (with manual labels and pseudo-labels). Introducing noise in the training of the student model (e.g. by data augmentation and dropout) facilitates to learn beyond the teacher's knowledge [9].

Figure 1. Example images of the EmotioNet dataset. Left: Alignment for expression AUs. Right: Alignment for pose AUs.

2. Methods

Preprocessing: We use the face detection, landmark localization, and head pose estimation of OpenFace [1] (following suggestions of [8]). To reduce the number of faces not detected (for which we output AU absence in the challenge evaluation), we additionally run RetinaFace [4] and the landmark localization of Bulat and Tzimiropoulos [3] on the images for which OpenFace failed. We then apply the OpenFace face registration approach, which is based on a stable subset of 68 landmarks, without masking out context. We use two different "zooms" as illustrated in Fig. 1: The one with more facial details is used for the expression

This work was funded by the German Federal Ministry of Education and Research (BMBF), projects HuBA (03ZZ0470) and Easy Cohmo (03ZZ0443G). The sole responsibility for the content lies with the authors.

AUs, the other with more context is used for the head pose AUs (which are defined relatively to the body, not the camera view). Both have the resolution 240×240, but are fed into two distinct CNN models.

As we will see in the experiments, the manually labeled optimization dataset is a better source for supervised learning than the larger but automatically (weakly) and incompletely labeled training set. Thus, and to get more validation attempts for selecting and tuning models, we split the *opt set*: 20% of the images are used as a validation set (called *opt-val set*) and the remaining 80% are used as training data (called *opt-train set*).

Convolutional Neural Network (CNN) Architectures: We use three architectures: (1) a self designed straight forward CNN which we call *OwnNet*, (2) *MobileNetV3* (large) [5], and (3) *EfficientNet-B0* [6]. With all networks we use an input resolution of 224×224×3. OwnNet-w has a variable width factor w and seven 2D convolution layers (Conv), each followed by batch normalization and ReLU. The first Conv has $4w$ output channels. After each Conv until the fifth, we apply a 3×3 max-pooling with stride 2 and double the output channels of the next Conv. The last two Conv have $128w$ output channels and are followed by global average pooling. In all networks we use a dropout of 0.5 in front of the final dense output layer, which is activated with the sigmoid function. For the head pose networks (AU 51-56) we additionally feed the three head orientation angles from OpenPose (or the mean pose to fill missing values) into a dense layer (1024 neurons, ReLU) and concatenate its outputs with the CNN outputs before the final dense layer. MobileNetV3 and EfficientNet-B0 models are fine-tuned starting from the pretrained model provided in the TensorFlow SLIM and TPU repositories, respectively. The OwnNet models are trained from scratch using the default Xavier initialization.

Training: We only train multi-label CNNs, basically with 17 AU labels / output neurons for the expression CNNs and 6 for the pose CNNs. However, in what we call multi-task in the following, we have one output per AU *and dataset*, e.g. we have one output for AU 1 of the *opt-train set* and one for AU 1 of the *training set* if we train with both datasets. For prediction we always use the outputs trained with the *opt-train set*, which is more accurately labeled, but following the multi-task idea, the performance can benefit from adding the huge 944k samples *training set* because it helps to learn better features. With a batch of N samples and a CNN with M outputs the loss is calculated as:

$$L(y, \hat{y}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \lambda_m \cdot w_m(y_{n,m}) \cdot l(y_{n,m}, \hat{y}_{n,m}), \quad (1)$$

with y being the target label, $\hat{y}$ the prediction, λ_m a label-specific weight, and $l(y, \hat{y})$ the binary cross-entropy. The λ_m-values are tuned to adjust the training speed of the different AUs in order to avoid that some AUs are already overfitting while others are still underfitted. For each label there is a class-dependent weighting function $w_m(y)$, which zeros the loss for missing labels (unknown class) and reduces the negative impact of the class imbalance, which is common in AU recognition [7]. For this purpose, $w_m(y)$ weights down the majority class samples and weights up the minority class samples following the imbalancing damping idea of [7] (with $\alpha = 0.5$ for the expression AUs and $\alpha = 0.7$ for the pose AUs). All weights are normalized to not increase the average gradient length. The loss is optimized with stochastic gradient descent. We use a batch size of 16 and assemble the batches by equally sampling from the datasets used for training. We apply early stopping with a fixed number of epochs (OwnNet 500k, MobileNetV3 150k, EfficientNet-B0 300k) and start with a learning rate of 0.1 (expression CNN) / 0.01 (pose CNN), reducing it by a factor of 0.33 after half and three quarters of the iterations. For data augmentation we use random cropping, horizontal flipping, brightness and contrast adjustments, cutout, as well as occasional downscaling and grayscale conversion. Additionally, label smoothing (0.2) and weight decay (4e-6) are used.

Self-Training: Inspired by [9] we use self-training to benefit from the large 944k weakly / partly labeled *training set*: We first train a model on the fully labeled *opt-train set* and apply it to predict pseudo-labels on the *training set*. Afterwards we train a second model using both, the *opt-train set* (with manual labels) and the *training set* (with pseudo-labels). This way, the first model acts as a teacher and the second as a student. In a second iteration, the student model can be used to update the pseudo-labels and train a second generation student model.

Ensemble: To improve results further we combine the predictions of several well-performing student models in heterogeneous ensembles. The models differ regarding the pseudo-labels used for training and the CNN architectures. We fuse the predictions by calculating the mean of the models' output scores before rounding the resulting scores for the final decision.

3. Experiments

Image Alignment: To analyze the impact of more details vs more context, we trained one common OwnNet8 model for all 23 AUs with the close-up view alignment and one with the more-context view (left and right in Fig. 1). The expression AUs performed better with the close-up

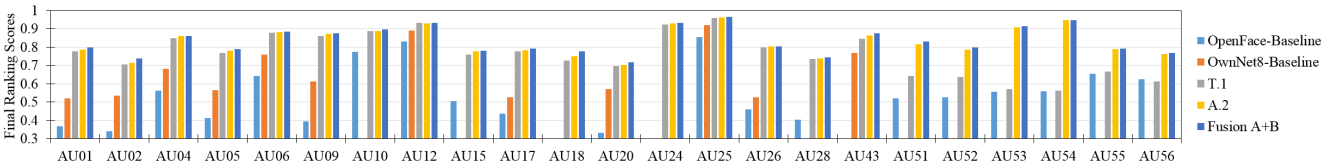

Figure 2. Per-AU final ranking scores on the *opt-val set*: OpenFace-Baseline and OwnNet8-Baseline (see text), the best teacher model T.1 (only trained on optimization set), the best student model A.2, and the fusion model A+B (mean score of all A and B models).

#	Model	Score
T.0	OwnNet3 teacher (*opt-train* only)	0.7531
-	↳ OwnNet8 student	0.7647
-	↳ EfficientNet-B0 student	**0.7661**
T.1	OwnNet8 teacher (*opt-train* only)	0.7637
B.1	↳ MobileNetV3 student	0.7680
A.1	↳ EfficientNet-B0 student	**0.7699**
-	↳ EfficientNet-B0 student	0.7694
-	↳ MobileNetV3 student	0.7671
-	↳ OwnNet8 student	0.7666
-	MobileNetV3 teacher (*opt-train* only)	0.7518
T.2	MobileNetV3 teacher (*opt-train* and *training set*)	0.7603
-	↳ MobileNetV3 student	0.7623
-	↳ EfficientNet-B0 student	0.7666
B.2	↳ EfficientNet-B0 student	0.7674
A.2	↳ OwnNet8 student	**0.7706**
-	↳ OwnNet8 student (no multi-task)	0.7634
-	EfficientNet-B0 teacher (*opt-train* only)	0.7602
T.3	EfficientNet-B0 teacher (*opt-train* and *training set*)	0.7609
A.3	↳ OwnNet8 student	**0.7684**
B.3	↳ MobileNetV3 student	0.7680
-	Fusion A (mean score of A.1, A.2, A.3)	0.7767
-	Fusion B (mean score of B.1, B.2, B.3)	0.7754
-	Fusion A+B (mean score of all A and B)	**0.7800**

Table 1. Final ranking scores on the *opt-val set*. Indentation and arrows show the teacher-student relation. T.* are identifiers of the teacher models, A.* of the best student models in the category, B.* of the second best student models. All student models have been trained on the *opt-train set* (with manual labels) *and* the *training set* (with pseudo-labels generated by the teacher model) with multi-task learning (if not denoted differently).

view (mean: 0.784 vs 0.772) and the head pose AUs with the more-context view (mean: 0.554 vs 0.528). Thus, we trained two CNNs in the following as mentioned in Sec. 2: one with with close-up view images for expression AUs and one with more-context view for pose AUs.

Multi-Task Self-Training: Table 1 shows validation results obtained on the *opt-val set*. Some early teacher models (OwnNet T.0 and T.1) were trained on the *opt-train set* only, without using the 944k samples of the official *training set*. After using multi-task learning for the student models, we also trained teacher models with multi-task learning (using the *opt-train set* and the *training set* with the labels provided by [2]). These performed better than the respective teacher models trained on only the *opt-train set* (compare T.2 and T.3 with the respective line above). All student models outperform their respective teacher models, except the second generation student models learning from the pseudo-labels provided by the first generation student A.1. So the self-training generally improves the results at least for the first iteration. Comparing A.2 with the row below, which has been trained without multi-task using the same output neurons for the pseudo-labels of the 944k *training set* and the manually labeled *opt-train set*, we see that multi-task learning is beneficial in combination with self-training, as the pseudo-labels are still less accurate than the manual labels.

Ensemble: The last three rows of Table 1 list the results of combining the outputs of several models. All individual models are outperformed by the three tested ensembles. The fusion of all A and B models performs best.

Per-AU comparison: The challenge task was to recognize 23 Action Units (AUs). Fig. 2 shows the per-AU results of several models, including two baselines: The OpenFace-Baseline tests the expression AU output as provided by OpenFace [1]. The pose AUs (51-56) were predicted with an RBF-SVM trained on the head orientation angles provided by OpenFace. The OwnNet8-Baseline is similar to T.1, but trained with the 944k *training set* and the labels automatically created by [2] (instead of the smaller *opt-train set* with manual labels). The comparison (1) of OpenFace with the others shows the benefit of our CNN approach compared to OpenFace's classical approach consisting of feature extraction (HOG + landmarks / head pose) followed by SVM; (2) of OwnNet8-Baseline with T.1 shows that training with less but high quality labels in this case is better than relying only on many more but weakly labeled samples; and (3) of T.1 and A.2 shows that especially the head pose AUs (51-56) significantly benefit from self-training. Fusion A+B consistently improves results compared to A.2, but the benefits differ significantly between AUs.

EmotioNet Challenge Results: Table 2 summarizes the final ranking scores obtained on the official EmotioNet 2020 and 2018 Challenge validation and test sets. For our final submission, we retrained all student models involved

Model / Participant	Challenge 2020		Challenge 2018	
	Val. Set	Test Set	Val. Set	Test Set
Our models				
- Teacher T.0	0.7143	-	0.7754	-
- Teacher T.1	0.7213	-	0.7828	-
- Student A.1	0.7324	-	0.7873	-
- Fusion A+B	0.7448	-	0.8011	-
- *Fusion A+B**	**0.7452**	**0.7301**	**0.8014**	**0.7734**
Competitors 2020				
- TAL	**0.7460**	**0.7306**	-	**0.7722**
- UCAS-NTU	0.6363	0.6711	-	0.7377
- UCAS-alibaba	-	0.6053	-	0.6428
Best results 2018				
- PingAn-GammaLab	-	-	**0.7855**	**0.7553**
- VisionLabs	-	-	0.6788	0.6718
- MIT	-	-	0.5995	0.6711
- Univ. of Washington	-	-	0.6645	0.6300

Table 2. Final ranking scores obtained on the validation and test set of the EmotioNet 2020 Challenge (and its 2018 predecessor): Our results, results of the best 2020 competitors, and of the best 2018 challenge participants.

Participant	Challenge 2020		Challenge 2018	
	Accuracy	F1	Accuracy	F1
TAL	**0.9147**	0.5465	**0.9499**	0.5945
Univ. of Magdeburg	0.9124	**0.5478**	0.9458	**0.6009**
UCAS-NTU	0.9013	0.4410	0.9485	0.5268
PingAn-GammaLab	-	-	0.9446	0.5659
VisionLabs	-	-	0.9207	0.4229
MIT	-	-	0.9298	0.4125

Table 3. Accuracies and F1-scores of the best participants on the test set of the EmotioNet 2020 and 2018 Challenges. We (Univ. of Magdeburg) perform better in F1, TAL is better in accuracy.

in the A+B fusion with the whole *opt set* instead of the *opt-train* subset to benefit from 5k additional manually labeled samples. This model is denoted "Fusion A+B*" in Table 2. The table also includes our prior validation attempts and the best results of the other challenge participants. Our "Fusion A+B*" model achieved the second place in the 2020 challenge, with a test performance that is only 0.05% worse than the winner (TAL) but 5.9% better than the third place (NCAS-NTU). On the 2018 challenge data (12 AUs; without AU 10, 15, 18, 24, 28, and 51-56), we outperform all other known results, including the 2018 challenge winner (PingAn-GammaLab) and the 2020 challenge winner (TAL). Table 3 compares the mean accuracies and F1 performances of the best 2020 and 2018 challenge participants.

4. Conclusion

In this paper we described our approach for facial Action Unit (AU) recognition in the wild, which uses: (1) self-training, (2) multi-task learning, (3) an heterogeneous ensemble involving three CNN architectures, (4) a weighted loss for handling data imbalance, and (5) a more-detail face alignment for expression AUs and a more-context face alignment for head pose AUs. With this approach we reached the second place in the EmotioNet 2020 Challenge (with only a small margin of 0.05% to the winner), without using additional training data next to EmotioNet dataset provided by the challenge organizers. Further, we achieved the best result reported so far on the EmotioNet 2018 Challenge data.

Several experiments showed that self-training can improve AU recognition if a large amount of unlabeled data is available. This is promising for future works, since acquiring FACS labels is expensive and unlabeled data is available virtually indefinitely. We can also recommend to apply multi-task learning if multiple datasets are combined and the datasets' AU labels differ regarding their quality.

References

[1] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. P. Morency. OpenFace 2.0: Facial behavior analysis toolkit. In *FG*, 2018. 1, 3

[2] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez. EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild. In *CVPR*, 2016. 1, 3

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 1

[4] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-stage Dense Face Localisation in the Wild. *arXiv:1905.00641 [cs.CV]*, may 2019. 1

[5] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for MobileNetV3. In *ICCV*, 2019. 2

[6] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML*, 2019. 2

[7] P. Werner, F. Saxen, and A. Al-Hamadi. Handling Data Imbalance in Automatic Facial Action Intensity Estimation. In *BMVC*, 2015. 2

[8] P. Werner, F. Saxen, and A. Al-Hamadi. Landmark based head pose estimation benchmark and method. In *ICIP*, 2017. 1

[9] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with Noisy Student improves ImageNet classification. *arXiv:1911.04252v2 [cs.LG]*, 2020. 1, 2