MixAugment & Mixup:
Augmentation Methods for Facial Expression Recognition

Andreas Psaroudakis
National Technical University of Athens, Greece
andreaspsaroudakis@gmail.com

Dimitrios Kollias
Queen Mary University of London, UK
d.kollias@qmul.ac.uk

Abstract

Automatic Facial Expression Recognition (FER) has attracted increasing attention in the last 20 years since facial expressions play a central role in human communication. Most FER methodologies utilize Deep Neural Networks (DNNs) that are powerful tools when it comes to data analysis. However, despite their power, these networks are prone to overfitting, as they often tend to memorize the training data. What is more, there are not currently a lot of in-the-wild (i.e. in unconstrained environment) large databases for FER. To alleviate this issue, a number of data augmentation techniques have been proposed. Data augmentation is a way to increase the diversity of available data by applying constrained transformations on the original data. One such technique, which has positively contributed to various classification tasks, is Mixup. According to this, a DNN is trained on convex combinations of pairs of examples and their corresponding labels.

In this paper, we examine the effectiveness of Mixup for in-the-wild FER in which data have large variations in head poses, illumination conditions, backgrounds and contexts. We then propose a new data augmentation strategy which is based on Mixup, called MixAugment. According to this, the network is trained concurrently on a combination of virtual examples and real examples; all these examples contribute to the overall loss function. We conduct an extensive experimental study that proves the effectiveness of MixAugment over Mixup and various state-of-the-art methods. We further investigate the combination of dropout with Mixup and MixAugment, as well as the combination of other data augmentation techniques with MixAugment.

1. Introduction

The human emotion constitutes a conscious subjective experience that can be expressed in various ways. During the past decade, with the rapid development in the field of Artificial Intelligence, scientists have conducted numerous studies to develop systems and robots that will be capable of perceiving automatically people's feelings and behaviors [25,29]. An ultimate goal is the creation of digital assistants that will display a human-centered character and interact with users in the most natural way possible. It is a very complex and demanding task, since expression recognition in real world conditions is not easy and straightforward to do [20,33].

Over the last decade, Deep Neural Networks have emerged as a method to solve any computer vision task [50, 51,57]. DNNs in order to work and generalise well, need to be trained on large and diverse databases [31,38]. Nevertheless, in multiple applications, the collection of new data and their corresponding annotation is not always an easy or possible task to do (eg it is a quite time consuming and costly process). In the FER domain, RAFD-DB [42,43], Affect-Net [44] and Aff-Wild2 [15,21–24,27,30,32,34,36,37,58] are the most widely used in-the-wild databases. Additionally, despite DNNs' considerable power, the networks are prone to overfitting. This means that they often tend to memorize the input data or learn the noise and not the real data distribution, thus failing to generalize successfully when faced with data that are (considerably) different to the input ones.

One possible solution would be to expand the training set by adding new samples (although as we previously mentioned that is not always feasible due to inavailability of existing large in-the-wild datasets). Another way of extending the training set is by adding artificial samples that have been produced using 3D methods [1, 8, 18, 19, 53] or Generative Adversarial Networks (GANs) [4,9,35,45,60]. However, in this case, the generated samples must be realistic to the human eye, which still remains a very challenging task under investigation. In various application fields other problems may arise as well (eg for creating human faces the identity of the human should be preserved).

Another approach is to use data augmentation techniques, i.e., methods that produce new samples, by utilizing those that are already available and exist in the training set. Data augmentation is a way to increase the diver-

sity of available data by applying constrained transformations on the original data. A fairly recent technique of this kind, which has positively contributed to various tasks, is Mixup [59]. According to that, a DNN is trained on convex combinations of pairs of examples and their corresponding labels. By doing so, the distribution of the available data is extended and the generalization ability of the network improves. This principle has already been applied in some particular fields but has hardly ever been tried out in human affect estimation problems, especially in "in-the-wild" conditions with variations in head poses, illumination conditions, backgrounds and contexts.

In this paper, we examine the effectiveness of Mixup for 7 basic expression classification (categorical model [6]) by utilizing the Real-world Affective Faces Database (RAF-DB) [43], a large-scale facial expression database with around 30K great-diverse facial images downloaded from the Internet. We further propose a new data augmentation strategy that is based on Mixup, which we call MixAugment; according to this a DNN is trained on a combination of virtual examples and real examples. The overall loss function of DNN training consists of the loss of the real examples and the loss of the virtual examples. Finally we examine the effect of dropout [48] when used in combination with Mixup and our proposed MixAugment. Useful conclusions are drawn from the experimental study and the foundations are laid for future extensions.

2. Related Work

Mixup [59] constitutes a simple but powerful data augmentation routine that has already been applied in various tasks in Computer Vision, Natural Language Processing (NLP) and the audio domain. Some indicative examples pertain to medical image segmentation [5], sentence classification [2, 10, 49], audio tagging [55], audio scene classification [56] and image classification [12, 16, 17, 26, 28].

Regarding expression recognition, Mixup has been tried out only in very limited scenarios. In particular, this data augmentation technique was applied for the first time in speech expression recognition (SER) data to alleviate the issue of small existing datasets in the field. In [39] a framework that combined Mixup with a Generative Adversarial Network was proposed so as to improve the generation of synthetic samples. Specifically, they utilized this routine to train a GAN for synthetic expression feature generation and also for learning expression feature representation. To prove the effectiveness of the proposed framework, they showed results for SER on synthetic feature vectors, augmentation of the training data with synthetic features and encoded features in compressed representation. The results indicated that the proposed network can successfully learn compressed expression representations and can also produce synthetic samples that enhance performance in within-

corpus and cross-corpus evaluation.

Apart from the lack of many large in-the-wild datasets in the field of SER, another problem is affiliated with the common difference between the training and test data distributions. SER systems can achieve high accuracy when these two sets are identically distributed, but this assumption is often violated in practice and the systems' performance declines against unforeseen data shifts. In [41], the authors proved that the use of Mixup enhances the robustness to noise and adversarial examples in DNN Architectures. As a result, the generalization ability of the models improves and the DNNs perform better against unseen real-time situations. Moreover, the evaluations on the widely used IEMOCAP and MSP-IMPROV datasets showed that Mixup is a better augmentation technique for SER compared to the popular speed perturbation [40].

Jia and Zheng [14] tried to solve the problems of naturalness, robustness, fidelity and expression recognition accuracy in the process of expression speech synthesis. For that purpose, they designed an expression speech synthesis method based on multi-channel time–frequency generative adversarial networks (MC-TFD GANs) and Mixup. The comparative experiments were carried out on the IEMO-CAP corpus. The results showed that the mean opinion score (MOS) and the unweighted accuracy (UA) of the speech generated by the synthesis method were improved by 4% and 2.7%, respectively. The proposed method was superior in subjective evaluation and objective experiments, proving that the speech produced by this model had higher reliability, better fluency and emotional expression ability.

In terms of expression recognition from facial images, a published work in which Mixup is utilized, is from [46]. In this paper, the researchers made use of Mixup to improve the generalization of a proposed DNN, named eXnet. The model was trained and evaluated on FER-2013, CK+, and RAF-DB benchmark datasets. The experimental results showed that the model trained with Mixup technique witnessed an increase in accuracy of about 1%.

3. Methodology

3.1. Mixup

Mixup [59] is a simple and data-agnostic data augmentation routine that trains a DNN on convex combinations of pairs of examples and their labels. In other words, Mixup constructs virtual training examples $(\tilde{x}, \tilde{y})$ as follows:

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j$$
$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \qquad (1)$$

where x_i and x_j are two random raw inputs (i.e., images), y_i and $y_j \in \{0, 1\}^7$ are their corresponding one-hot label encodings and $\lambda \sim \mathrm{B}(\alpha, \alpha) \in [0, 1]$ (i.e., Beta distribution) for $\alpha \in (0, \infty)$.

Therefore, Mixup extends the training distribution by incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. By doing so, it regularizes the DNN (while training) to favor linear behavior in-between training examples. The implementation of Mixup training is straightforward, and introduces a minimal computation overhead.

In the studied case, the training samples are aligned facial images and the labels are one-hot encoding vectors corresponding to one of the 7 basic expressions. When training a DNN with the Mixup technique, the mixup loss function is the categorical cross entropy (CCE) of the virtual (v) samples defined as:

$$\mathcal{L}_{CCE}^{v} = \mathop{\mathbb{E}}_{\bar{y}, \tilde{y}}[-\tilde{y} \cdot \log \bar{y}] \qquad (2)$$

where $\bar{y}$ is the predicted probability of the sample $\tilde{x}$; $\tilde{x}$ and $\tilde{y}$ are given in Eq. 1.

An example of Mixup implementation on facial images is illustrated in Figure 1. An image that corresponds to a "happy" facial expression is linearly mixed with another one that demonstrates a "sad" expression, in a 60:40 ratio. The resulting image depicts a human face, that combines facial characteristics from the two initial images. Its label, which is written above the constructed image, states that this virtual sample belongs to class "happy" by 60% and to class "sad" by 40%.
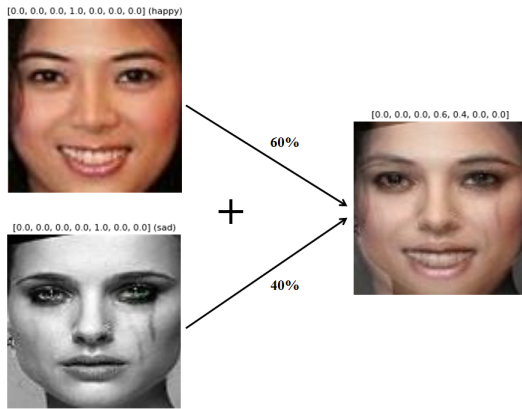

Figure 1. Construction of virtual example with the Mixup technique on samples taken from RAF-DB

3.2. MixAugment

In the typical Mixup data augmentation routine, randomly selected pairs of images are linearly interpolated and then fed into the DNN for training. However, "in-the-wild" facial databases contain a lot of images with large variations in head poses, gazes and angles. As a result, when mixing randomly selected images, it is possible for two images with

different head poses to be combined. An indicative example is illustrated in Figure 2, where a "happy" facial expression is mixed with a "sad" reaction, in a 50:50 ratio.

Figure 2. Example of mixing images with different head poses

As one can see, the resulting image does not resemble a real human face. Such cases may hinder training and learning of DNNs. To cope with this problem, we propose a simple approach name MixAugment. According to this, during each training iteration, the DNN is trained concurrently on both real (r) and virtual (v) examples. Specifically, in each training iteration, the DNN is fed with both x_i and x_j, and the generated image $\tilde{x} = \lambda x_i + (1-\lambda)x_j$ (of Eq. 1). In this scenario, the loss function is:

$$\begin{aligned} \mathcal{L}_{total} &= \mathcal{L}_{CCE}^{v} + \mathcal{L}_{CCE}^{r_i} + \mathcal{L}_{CCE}^{r_j} \\ &= \mathbb{E}[-\tilde{y} \cdot \log \bar{y} - y_i \cdot \log \bar{y_i} - y_j \cdot \log \bar{y_j}] \\ &= \mathbb{E}\big[-[\lambda y_i + (1-\lambda)y_j] \cdot \log \bar{y} - y_i \cdot \log \bar{y_i} - y_j \cdot \log \bar{y_j}\big] \\ &= \mathbb{E}[-y_i \cdot \log (\bar{y_i}\bar{y}^{\lambda}) - y_j \cdot \log (\bar{y_j}\bar{y}^{1-\lambda})] \quad (3) \end{aligned}$$

where $\bar{y}$ is the predicted probability of the sample $\tilde{x}$; $\tilde{x}$ and $\tilde{y}$ are given in Eq. 1; y_i and y_j are the labels of two (random) images (mentioned in Eq. 1) and $\bar{y_i}$ and $\bar{y_j}$ are their corresponding predicted probabilities (the indices in the expectations are omitted for simplicity).

As can be seen in Eq. 3 we merge the mixup loss with the classification loss to enhance the classification ability on both raw samples and mixup samples. This is different from the original design of Mixup [59] where the authors replaced the classification loss with the mixup loss.

4. Experimental Studies

4.1. Database

All the experiments are carried out utilizing the Real-world Affective Faces Database (RAF-DB), a large-scale facial expression database with around 30K great-diverse facial images downloaded from the Internet. Based on
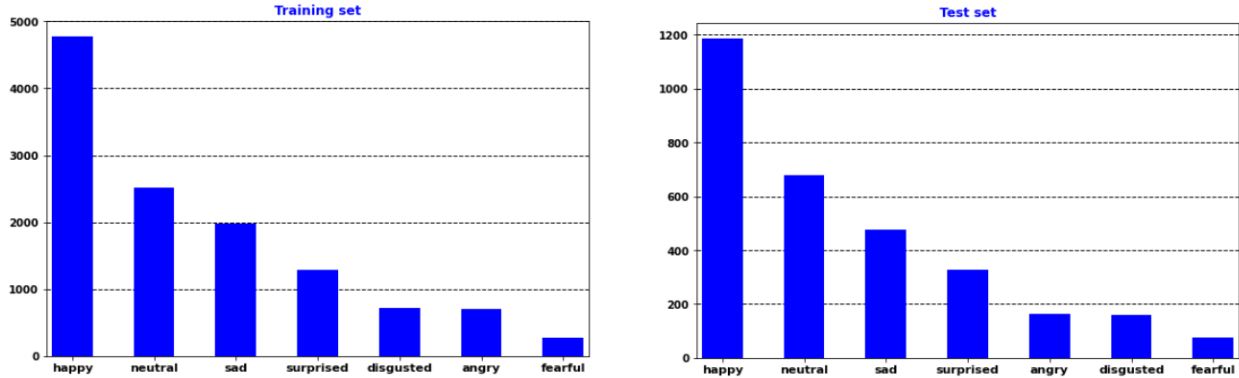
Figure 3. Class distribution of the RAF-DB training and test sets

the crowdsourcing annotation, each image has been independently labeled by about 40 annotators. Images in this database are of great variability in subjects' age, gender and ethnicity, head poses, lighting conditions, occlusions, (e.g. glasses, facial hair, self-occlusion). RAF-DB includes two subsets: (i) single-label subset, which consists of images annotated in terms of the seven basic expressions (surprise, fear, disgust, happiness, sadness, anger and neutral); (ii) multi-label subset, which consists of images annotated in terms of twelve compound expressions. In our experiments, we use the single-label subset. The database has been split into a training set (consisting of around 12,200 images) and a test set (consisting of around 3,100 images) where the size of training set is four times larger than the size of the test set; expressions in both sets have a near-identical distribution, as illustrated in Figure 3. It is worth mentioning that the two sets are imbalanced, with the expression "happy" having by far the largest number of samples and the class "fearful" being the least popular in both cases.

4.2. Performance Metric

For the evaluation of our models we make use of three different performance metrics: i) Accuracy, ii) Average Accuracy (mean diagonal of the normalized confusion matrix) and iii) macro F1-score (harmonic mean of Precision and Recall). Accuracy is defined as the percentage of correct predictions among the total number of predictions. It is the most common evaluation metric, however not preferred in imbalanced classification problems. Our dataset is imbalanced, therefore, to have a superior insight, we should take into account some additional metrics. The Average Accuracy (aka macro Recall) and macro F1-score are useful, since they give equal importance to each class, in contrast to Accuracy which gives equal importance to each sample, thus favoring majority classes. During the training phase, we monitor these metrics, and if no improvement is observed over the test set, we apply early stopping and keep the best configuration.

4.3. Pre-processing

Data pre-processing consists of the steps required for facilitating extraction of meaningful features from the data. In a typical expression recognition problem with facial images, the usual steps are face detection and alignment, image resizing and image normalization. We experimented with using two different face detectors to extract bounding boxes around each face and detect 68 facial landmarks. In the first version of the database (the public release), a face detector from the dlib library has been used, while in the other case the detector is the RetinaFace [3]. The alignment step is the same for both versions. Out of all 68 located landmarks, we focus on 5 - corresponding to the location of the left eye, right eye, nose and mouth in a prototypical frontal face - as rigid, anchor points. Then, for every image, the respective 5 facial landmarks are extracted and affinity transformations between the coordinates of these 5 landmarks and the coordinates of the 5 landmarks of the frontal face are computed; these transformations are imposed to the whole new frame for the alignment to be performed. All resulting images are resized to $100 \times 100 \times 3$ or $112 \times 112 \times 3$. Finally, all cropped and aligned images' pixel intensity values are normalized to the range $[0, 1]$.

4.4. Training Implementation Details

Table 1 demonstrates all implementation details pertained to the training session. Where dropout [48] was applied, its value was 0.5. In the following, to not clutter the presented results, we present results only for the publicly-released dataset version (1st version); the same conclusions have been drawn when utilizing the other version (2nd).

4.5. Results

Utilize Mixup vs Vanilla Case We start our experiments by training a ResNet50 [11], pretrained on ImageNet, for 100 epochs, when applying and not applying dropout (vanilla case). We also train the exact same

Table 1. Training parameters with their corresponding values

Parameters	Values
Image size	$100 \times 100 \times 3$ (1st version) $112 \times 112 \times 3$ (2nd version)
Batch size	32
Loss function	Categorical cross entropy
Optimizer	Adam
Learning rate	$10^{-3}, 10^{-4}$
Dropout rate	0.5
Number of epochs	100

Table 2. ResNet50 trained with Mixup and without Mixup (i.e., vanilla case)

Mixup	Dropout	Accuracy	F1-score	Aver. Acc.
No	✗	83,31	75,25	73,46
	✓	83,21	75,16	74,07
$\alpha = 0.1$	✗	**84,06**	**75,51**	**74,38**
	✓	83,25	74,88	73,60
$\alpha = 0.2$	✗	82,33	73,95	73,38
	✓	83,12	74,62	74,14
$\alpha = 0.6$	✗	83,15	74,81	72,73
	✓	83,47	75,23	74,29
$\alpha = 1$	✗	82,55	74,01	73,29
	✓	82,67	74,69	73,58

model (same weight initialization) with Mixup, for $\alpha \in \{0.1, 0.2, 0.6, 1\}$. Table 2 illustrates the results of these experiments. Let us not that large values of the hyperparameter ($\alpha \in \{4, 8\}$) in Mixup lead to underfitting and not good network performance. In Table 2, it can be seen that the best configuration (i.e., when the model trained with Mixup achieves its higher performance in all 3 studied metrics) is when $\alpha = 0.1$ and no dropout has been used. In this case the model trained with Mixup outperforms by at least 0.3% in all studied evaluation metrics the model trained without Mixup. In terms of the use of dropout, Table 2 shows that its addition sometimes contributes positively, whereas some other times seems to contribute negatively.

Utilize MixAugment vs Vanilla Case Similar as before, we use the same model (ResNet50 pretrained on ImageNet) with the same training parameters and compare its performance when the proposed MixAugment is and is not used. Table 3 illustrates that performance comparison for $\alpha \in \{0.1, 0.2, 0.6, 1\}$ and when dropout is and is not applied. Similarly as in the Mixup case, we noticed that for large values of the hyperparameter ($\alpha \in \{4, 8\}$) Mixup leads to underfitting and and not good network performance. In Table 3, it can be seen that the best configuration is when $\alpha = 0.1$ and no dropout has been used. In this case, the

model trained with MixAugment outperforms by at least 1.7% in all studied evaluation metrics the model trained without MixAugment. In terms of the use of dropout, Table 3 shows that its addition sometimes contributes positively, whereas some other times seems to contribute negatively. Finally, one can observe in Tables 2 and 3, that in both cases (when Mixup or MixAugment have been used), best results across all metrics have been obtained when $\alpha = 0.1$ and no dropout has been used. We can deduct that optimal results cannot be achieved when dropout is used in addition to Mixup or MixAugment.

Table 3. ResNet50 trained with the proposed MixAugment and without MixAugment (i.e., vanilla case)

MixAugment	Dropout	Accuracy	F1-score	Aver. Acc.
No	✗	83,31	75,25	73,46
	✓	83,21	75,16	74,07
$\alpha = 0.1$	✗	**85,04**	**77,30**	**75,32**
	✓	83,96	76,03	73,77
$\alpha = 0.2$	✗	84,19	76,57	74,74
	✓	84,39	76,64	75,13
$\alpha = 0.6$	✗	84,13	75,04	73,38
	✓	84,26	76,46	74,38
$\alpha = 1$	✗	83,74	75,43	73,87
	✓	83,51	75,58	74,36

Next, in Table 4 we present the model's confidence for the correct and wrong predictions under two settings: i) when the model is trained with the proposed MixAugment and ii) when it is trained without MixAugment (this is the vanilla case). As illustrated in Table 4, our proposed technique helps the network make right decisions with higher confidence and wrong decisions with less assuredness, which is obviously desirable.

Table 4. Prediction confidence with MixAugment and without MixAugment (i.e. vanilla case)

Confidence	mean		median	
Type of predictions	correct	wrong	correct	wrong
No MixAugment (Vanilla)	96,37	92,66	98,82	99,69
MixAugment	**98,77**	**84,24**	**100,0**	**90,75**

In addition, it would be interesting to examine the performance of MixAugment separately on each one of the 7 basic expression categories across various metrics. For that purpose, in Table 5, we present the Precision, Recall and F1-score for each class when the model is trained with and without MixAugment. It can be seen that when the network is trained with MixAugment, in the vast majority of the categories, there is a substantial rise in all the aforementioned metrics. It is worth mentioning that the highest growth is observed for the minority class "fearful". Particularly, Pre-

Table 5. Comparison between ResNet50 trained with the proposed MixAugment and without MixAugment (i.e., vanilla case) for each one of the 7 classes

Class	Precision		Recall		F1-score		Samples
	No MixAugment (Vanilla)	MixAugment	No MixAugment (Vanilla)	MixAugment	No MixAugment (Vanilla)	MixAugment	
surprised	83,49	↑ **85,14**	79.94	↑ **83,59**	81,68	↑ **84,36**	329
fearful	69,09	↑ **78,85**	51,35	↑ **55,41**	58,91	↑ **65,08**	**74**
disgusted	62,60	↑ **65,89**	51,25	↑ **53,12**	56,36	↑ **58,82**	160
happy	92,33	↑ **92,73**	93,50	↓ 92,57	92,91	↓ 92,65	1185
sad	81,01	↑ **83,37**	80,33	↑ **82,85**	80,67	↑ **83,11**	478
angry	79,47	↓ 75,08	74,07	↓ 71,60	76,68	↓ 75,08	162
neutral	78,30	↓ 77,73	85,44	↑ **86,76**	81,72	↑ **82,00**	680

cision, Recall and F1-score increase by approximately 10%, 5% and 6% respectively when MixAugment is used. This is a really promising result, since there are many classification tasks in which the classes with the smallest number of samples are of paramount importance (e.g. medical classification problems).

Utilize MixAugment vs Utilize Mixup vs Vanilla Case To summarise the main presented results and to illustrate the difference in ResNet50's performance when the model is trained with the proposed MixAugment, when it is trained with Mixup and when it is trained without any of these, we have created Table 6. It can be observed that Mixup improves the model's performance and MixAugment further improves its performance. Compared to Mixup, our technique further improves all three evaluation metrics for at least 1%. Finally, it is notable to mention that when MixAugment is used in network training, the convergence is faster compared to the cases when Mixup is used or when neither of the two is used.

Table 6. ResNet50 trained with MixAugment, with Mixup and without any of the two

Method	Accuracy	F1-score	Aver. Acc.
Vanilla	83,31	75,25	73,46
Mixup	84,06	75,51	74,38
MixAugment	**85,04**	**77,30**	**75,32**
MixAugment + Flipping	**86,06**	**78,24**	**76,28**

Finally, let us mention a final experiment that we conducted. When training the model (ResNet50) with the proposed MixAugment we further performed flipping, which resulted in further boosting the model's performance by around 1% in each studied metric (Accuracy, F1-score and Average Accuracy).

Utilize MixAugment with other DNNs We further used our proposed MixAugment when training other widely used DNNs, such as VGG16 [47], DenseNet121 [13] and EfficientNet [52]. We noticed the same observations as before (i.e., as in the case of ResNet50 described previously). In more detail, the performance of these networks trained with MixAugment outperformed -in all 3 studied metrics- the performance of the networks trained with Mixup, which outperformed -over all metrics- the performance of the corresponding vanilla networks.

Utilize MixAugment vs State-of-the-Art In the previous cases, our model (ResNet50) was only pre-trained on ImageNet. It is known that if the model is further pre-trained on a similar task to the studied one (which is FER), then its performance further increases. Therefore we first pre-trained ResNet50 on AffectNet and then trained it with MixAugment and flipping. In Table 7 we compare its performance to the performance of various state-of-the-art methods. It can be observed that our approach outperforms all state-of-the-art methods in the accuracy metric and shows a slightly worse performance than two state-of-the-art methods (FaceBehaviorNet (Residual) [24]) and VGGFACE [19]) in the average accuracy metric.

Table 7. Performance comparison between state-of-the-art methods and ResNet50 trained with MixAugment

Method	Accuracy	Aver. Acc.
RAN [54]	86,90	-
Ad-Corre [7]	86,96	-
mSVM + DLP-CNN [42, 43]	-	74,20
MT-ArcRes [34]	-	75,00
MT-ArcVGG [34]	-	76,00
FaceBehaviorNet (VGG) [23]	-	71,00
FaceBehaviorNet (Residual) [24]	-	**78,00**
VGGFACE [19]	-	77,50
pre-train, MixAugment + Flipping	**87,54**	77,30

The FaceBehaviorNet (Residual) [24] is a multi-task

learning network that has been trained on over 5M of images. The VGGFACE [19] is a network that has been trained on an augmented training set consisting of the training set of RAF-DB plus 13,000 other synthetic/generated images (more than the training size of RAF-DB); therefore our method is expected to perform worse than such methods.

5. Conclusion and Future Work

In this paper, at first, we examine the effectiveness of Mixup for in-the-wild Facial Expression Recognition. Mixup is a data augmentation technique in which a DNN is trained on convex combinations of pairs of examples and their corresponding labels. Taking into account that in in-the-wild FER people display high variations in head poses, illumination conditions, backgrounds and contexts, we have proposed a variation of Mixup, called MixAugment in which the network is trained on a combination of virtual examples (generated by Mixup) and real examples; in our approach both examples contribute to the overall loss function. We have conducted a large experimental study that includes: performance comparison between models trained with Mixup, MixAugment or without any of the two versus state-of-the-art methods; ablation studies; testing the combination of Mixup or MixAugment and dropout; testing the combination of MixAugment and other data augmentation techniques such as flipping. The experimental study proves that models perform better when our proposed MixAugment is used during training.

In our future work, we aim to extend and apply the proposed MixAugment technique to other small and large-scale "in-the-wild" datasets, as well as to other affect recognition tasks, such as valence-arousal estimation and action unit detection.

References

[1] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F Cohen. Bringing portraits to life. *ACM Transactions on Graphics (TOG)*, 36(6):196, 2017. 1

[2] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*, 2020. 2

[3] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020. 4

[4] Hui Ding, Kumar Sricharan, and Rama Chellappa. Exprgan: Facial expression editing with controllable expression intensity. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1

[5] Zach Eaton-Rosen, Felix Bragman, Sebastien Ourselin, and M Jorge Cardoso. Improving data augmentation for medical image segmentation. 2018. 2

[6] Paul Ekman. Facial action coding system (facs). *A human face*, 2002. 2

[7] Ali Pourramezan Fard and Mohammad H Mahoor. Ad-corre: Adaptive correlation-based loss for facial expression recognition in the wild. *IEEE Access*, 10:26756–26768, 2022. 6

[8] Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM Transactions on Graphics (TOG)*, 35(4):128, 2016. 1

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[10] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019. 2

[11] 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. 4

[12] Junlin Hou, Jilan Xu, Rui Feng, Yuejie Zhang, Fei Shan, and Weiya Shi. Cmc-cov19d: Contrastive mixup classification for covid-19 diagnosis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 454–461, 2021. 2

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6

[14] Ning Jia and Chunjun Zheng. Emotion speech synthesis method based on multi-channel time–frequency domain generative adversarial networks (mc-tfd gans) and mixup. *Arabian Journal for Science and Engineering*, pages 1–14, 2021. 2

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

[16] Dimitrios Kollias, Anastasios Arsenos, Levon Soukissian, and Stefanos Kollias. Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 537–544, 2021. 2

[17] Dimitrios Kollias, N Bouas, Y Vlaxos, V Brillakis, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and S Kollias. Deep transparent prediction through latent representation analysis. *arXiv preprint arXiv:2009.07044*, 2020. 2

[18] 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

[19] 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, 6, 7

[20] Dimitris Kollias, George Marandianos, Amaryllis Raouzaiou, and Andreas-Georgios Stafylopatis. Interweaving deep learning and semantic techniques for emotion analysis in human-machine interaction. In *2015 10th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 1–6. IEEE, 2015. 1

[21] 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

[22] 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

[23] 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, 6

[24] 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, 6

[25] Dimitrios Kollias, Athanasios Tagaris, and Andreas Stafylopatis. On line emotion detection using retrainable deep neural networks. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE, 2016. 1

[26] Dimitrios Kollias, Athanasios Tagaris, Andreas Stafylopatis, Stefanos Kollias, and Georgios Tagaris. Deep neural architectures for prediction in healthcare. *Complex & Intelligent Systems*, 4(2):119–131, 2018. 2

[27] 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

[28] Dimitris Kollias, Y Vlaxos, M Seferis, Ilianna Kollia, Levon Sukissian, James Wingate, and S Kollias. Transparent adaptation in deep medical image diagnosis. In *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pages 251–267. Springer, 2020. 2

[29] Dimitrios Kollias, Miao Yu, Athanasios Tagaris, Georgios Leontidis, Andreas Stafylopatis, and Stefanos Kollias. Adaptation and contextualization of deep neural network models. In *Computational Intelligence (SSCI), 2017 IEEE Symposium Series on*, pages 1–8. IEEE, 2017. 1

[30] Dimitrios Kollias and Stefanos Zafeiriou. Aff-wild2: Extending the aff-wild database for affect recognition. *arXiv preprint arXiv:1811.07770*, 2018. 1

[31] Dimitrios Kollias and Stefanos Zafeiriou. A multi-component cnn-rnn approach for dimensional emotion recognition in-the-wild. *arXiv preprint arXiv:1805.01452*, 2018. 1

[32] Dimitrios Kollias and Stefanos Zafeiriou. A multi-task learning & generation framework: Valence-arousal, action units & primary expressions. *arXiv preprint arXiv:1811.07771*, 2018. 1

[33] Dimitrios Kollias and Stefanos Zafeiriou. Training deep neural networks with different datasets in-the-wild: The emotion recognition paradigm. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018. 1

[34] Dimitrios Kollias and Stefanos Zafeiriou. Expression, affect, action unit recognition: Aff-wild2, multi-task learning and arcface. *arXiv preprint arXiv:1910.04855*, 2019. 1, 6

[35] 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

[36] 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

[37] 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

[38] Dimitrios Kollias and Stefanos P Zafeiriou. Exploiting multi-cnn features in cnn-rnn based dimensional emotion recognition on the omg in-the-wild dataset. *IEEE Transactions on Affective Computing*, 2020. 1

[39] Siddique Latif, Muhammad Asim, Rajib Rana, Sara Khalifa, Raja Jurdak, and Björn W Schuller. Augmenting generative adversarial networks for speech emotion recognition. *arXiv preprint arXiv:2005.08447*, 2020. 2

[40] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, and Julien Epps. Direct modelling of speech emotion from raw speech. *arXiv preprint arXiv:1904.03833*, 2019. 2

[41] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, and Björn W Schuller. Deep architecture enhancing robustness to noise, adversarial attacks, and cross-corpus setting for speech emotion recognition. *arXiv preprint arXiv:2005.08453*, 2020. 2

[42] Shan Li and Weihong Deng. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Transactions on Image Processing*, 28(1):356–370, 2019. 1, 6

[43] Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2852–2861, 2017. 1, 2, 6

[44] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. 1

[45] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation:

Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 818–833, 2018. 1

[46] Muhammad Naveed Riaz, Yao Shen, Muhammad Sohail, and Minyi Guo. Exnet: An efficient approach for emotion recognition in the wild. *Sensors*, 20(4):1087, 2020. 2

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2, 4

[49] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. Mixup-transformer: dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*, 2020. 2

[50] Athanasios Tagaris, Dimitrios Kollias, and Andreas Stafylopatis. Assessment of parkinson's disease based on deep neural networks. In *International Conference on Engineering Applications of Neural Networks*, pages 391–403. Springer, 2017. 1

[51] Athanasios Tagaris, Dimitrios Kollias, Andreas Stafylopatis, Georgios Tagaris, and Stefanos Kollias. Machine learning for neurodegenerative disorder diagnosis—survey of practices and launch of benchmark dataset. *International Journal on Artificial Intelligence Tools*, 27(03):1850011, 2018. 1

[52] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6

[53] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 1

[54] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020. 6

[55] Shengyun Wei, Kele Xu, Dezhi Wang, Feifan Liao, Huaimin Wang, and Qiuqiang Kong. Sample mixed-based data augmentation for domestic audio tagging. *arXiv preprint arXiv:1808.03883*, 2018. 2

[56] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu. Mixup-based acoustic scene classification using multi-channel convolutional neural network. In *Pacific Rim conference on multimedia*, pages 14–23. Springer, 2018. 2

[57] Miao Yu, Dimitrios Kollias, James Wingate, Niro Siriwardena, and Stefanos Kollias. Machine learning for predictive modelling of ambulance calls. *Electronics*, 10(4):482, 2021. 1

[58] 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

[59] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 3

[60] Yuqian Zhou and Bertram Emil Shi. Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 370–376. IEEE, 2017. 1