Emotion Detection with Ensemble-based CNN

Yun Cheng
Carnegie Mellon University
Pittsburgh, PA 15213
yuncheng@andrew.cmu.edu

Zhiyi Kuang
Carnegie Mellon University
Pittsburgh, PA 15213
zkuang@andrew.cmu.edu

Yuxin Pei
Carnegie Mellon University
Pittsburgh, PA 15213
yuxinp@andrew.cmu.edu

Abstract

Emotion detection, or facial emotion recognition, refers to the process of detecting human emotions from facial expressions. While there have been several state-of-the-art deep learning networks for emotion detection, we aimed to combine new methods in recent research to achieve better results for Facial Expression Recognition Challenge 2013 (FER2013). In this paper, we explored transfer learning and support vector machine classification layer for fine-tuning pre-trained deep convolutional neural network (CNN) models. Based on that, we developed an ensemble model that achieved a test accuracy of 72.9%, which shows a large performance gain from individual networks. Finally, we used confusion matrices and occlusion-based saliency maps for better interpretability and analysis of our models' behavior that led to this significant improvement.

1 Introduction

Facial expression is one of the most powerful, natural, and universal signals for human beings to convey their emotional states and intentions [2]. Numerous studies were conducted on emotion detection because of its practical usage in preceding drowsiness of drivers, medical treatment, market research, and building effective artificial intelligence that can read, interpret, and respond to human emotions [7].

Anger, Disgust, Fear, Happiness, Sadness, and Surprise are considered as the *prototypical facial expressions* [5]. However, in datasets such as Facial Expression Recognition Challenge 2013 (FER2013), head pose, face location, and illumination have introduced large variations and created new challenges for accurate emotion detection. As a solution, the convolutional neural network (CNN) has proven to be effective and robust in overcoming these challenges [5]. In this project, our goal is to exploit CNN models to classify face images from FER2013 into these seven facial expression categories.

In this paper, we present our work of performing transfer learning on state-of-the-art CNN models, including residual and VGG neural networks. We performed an ensemble method on individual networks and were able to achieve significant performance gain. Finally, we performed error analysis using techniques of confusion matrices and occlusion-based saliency maps for further analysis and better interpretability of the results.

2 Data

FER2013 is our primary dataset for training and testing our models.

2.1 Dataset

FER2013 consists of 35887 images of faces. Each image is 48x48 gray-scale and is classified into seven categories (Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral). The faces are more or less

Preprint. Under review.

centered. The training set consists of 28709 examples, and the public test set consists of 3589 examples. There is variability in age, illumination, pose, and expression intensity, and FER2013 is thus a more challenging dataset: the human accuracy on this dataset is around 65.5%.

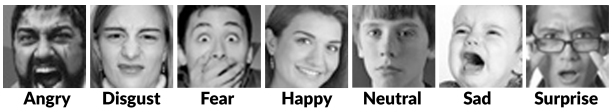


Figure 1: FER2013 dataset samples

2.2 Preprocessing

To account for the variability of image quality in the FER2013 dataset and improve the accuracy of our model, we combined the typical pipeline for data preprocessing with face normalization methods [5].

2.2.1 Facial Landmarks Registration and Affine Transformation

The facial features of each image are extracted using the library Dlib, and affine transformations are used to align them. Each input image is rotated such that the eyes are aligned on a horizontal line. The image is scaled so that the face is centered and the size of the face is identical.

2.2.2 Illumination Normalization

To remedy the effect of poor lighting and shadows on facial feature recognition, the intensity of each pixel in every image is normalized to have a mean of 0 and a norm of 100.

2.2.3 Pose Normalization

As demonstrated in Figure 2, frontalization synthesizes frontal facing views of faces appearing in single unconstrained photos. 2D coordinates of local facial features (using the face detector Dlib) are projected to a 3D reference surface that approximates the shape of input faces. Soft symmetry of faces is used to fill in the missing information in such projection and produce front-facing views of photos [3].

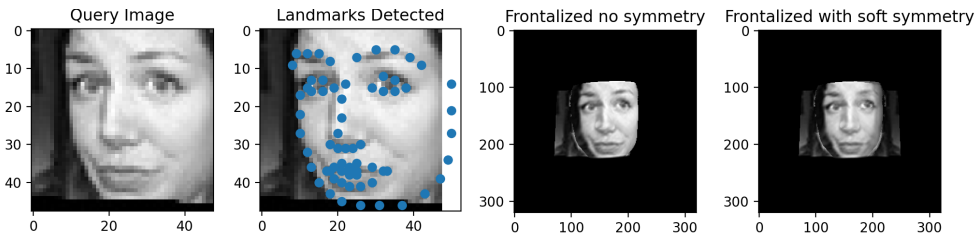


Figure 2: Frontalization process example

3 Related Work

Largely due to its practical usage, facial expression recognition has been a popular field of research for a long time. As shown in Figure 3, before 2013, majority of the methods focused on shallow learning (e.g., local binary patterns (LBP) [7], LBP on three orthogonal planes (LBP-TOP) [10], non-negative matrix factorization (NMF) [11] and sparse learning [12]) for FER. As GPU computing power improved significantly, and the size of datasets increased, researchers gradually transferred to deep learning as the main method of facial expression analysis. Several studies in the FER literature found that CNN is robust to face location changes and scale variations and behaves better than

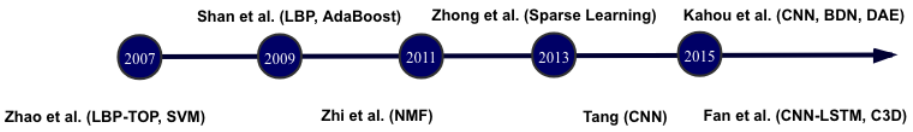


Figure 3: Evolution of facial expression

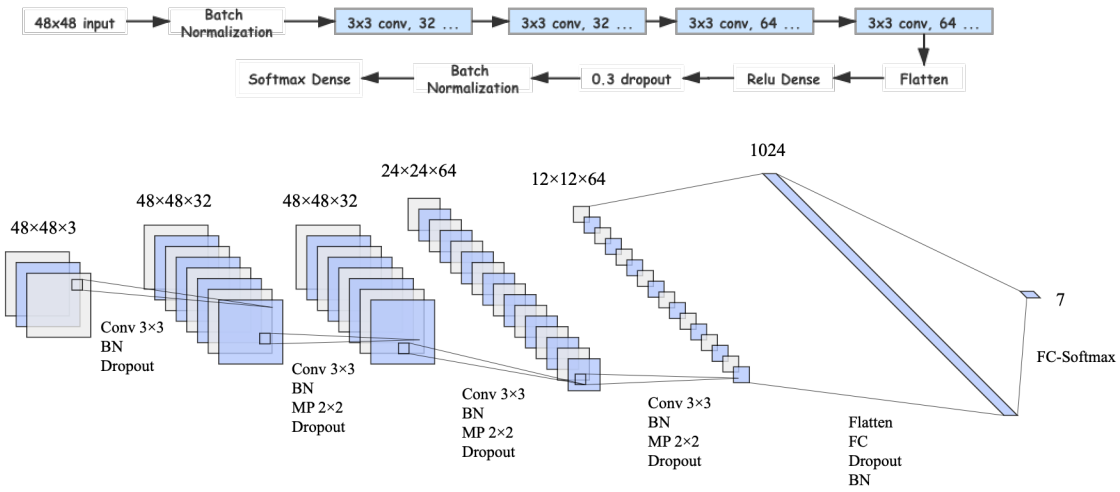


Figure 4: Baseline Model (BN: batch normalization; MP: max pooling, FC: fully-connected layer)

the multilayer perceptron (MLP) in the case of previously unseen face pose variations [5]. Some well-known CNN models in the field include AlexNet, VGGNet, GoogleNet, and ResNet. We included VGGNet and ResNet in our Transfer Learning method, and a more detailed explanation of these two models is provided in Section 4.

4 Methods

4.1 Baseline Model

To practice processing data and setting up models, we constructed a baseline model for this problem. The baseline model consists of a vanilla CNN using two 3x3x32 same-padding, ReLU filters, interleaved with 2x2 MaxPool layers, batch normalization, and 50% dropout layers (Figure 4). It achieved an accuracy of 67.5% for both softmax (Section 4.4.1) and support vector machine (Section 4.4.2) classification.

4.2 Transfer Learning

Since FER2013 is a relatively small and unbalanced dataset, training a deep neural network from scratch is challenging. Therefore, we turned to transfer learning for better feature extraction and performed fine-tuning on pre-trained models.

4.2.1 Residual Neural Networks

Residual neural networks have proven to be effective and efficient in image classification tasks [4]. We explored the most popular ResNet18, ResNet50, and SENet50 in the experiment. We obtained weights of these three models pre-trained on VGGFace2, a much larger dataset for face and pose recognition [1]. We froze all layers of the pre-trained model except for the last five layers. Then we replaced the last output layer with two fully-connected layers of size 4096 and 1024 with 50% dropout and a classification layer. We experimented with both softmax (Section 4.4.1) and support vector machine (Section 4.4.2) for the final classification and achieved accuracy around 60% (Section 5). For fine-tuning, all models are trained for 100 epochs with an initial learning rate of 0.01, batch size 128, and weight decay 0.0001. The loss is optimized with stochastic gradient descent with a momentum of 0.9. We adopted the strategy of halving the learning rate if the validation accuracy did not improve for more than 10 epochs.

4.2.2 VGG16

VGG16 is another pre-trained model that we explored. VGG16 is a convolutional neural network pre-trained on the ImageNet dataset, which consists of over 14 million images with more than 1000 classes and is much larger than the FER2013 dataset.
We froze all 16 pre-trained layers and added two fully-connected layers of size 4096 and 1024 respectively with a 50% dropout rate and a final classification layer. We experimented with both softmax (Section 4.4.1) and support vector machine (see Section 4.4.2) for the final classification layer. After 120 epochs of training with stochastic gradient descent optimizer with 0.001 learning rate and 0.9 momentum, we achieved 68.3% test accuracy for the VGG16 model with softmax classification layer and 66.8% for the VGG16 model with SVM classification layer.

4.3 Ensemble

We performed ensembling by summing up the prediction results from individual networks and choosing the label with the maximum probability as the prediction result of the ensemble model. We experimented with ensembling four models with softmax classification and four models with SVM classification and achieved an accuracy around 72% (Section 5). The best ensemble model we obtained consists of 6 models (VGG16 with softmax, VGG16 with SVM, ResNet50 with softmax, ResNet50 with SVM, and SeNet50 with softmax, and baseline model with softmax classification). The ensemble modeling has made significant improvement from individual performance (Section 5).

4.4 Classfication Objectives

4.4.1 Softmax Classification

Given the set of input $\{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^M$ and the corresponding label $\{y_i\}_{i=1}^N$, the softmax layer outputs the predicted probability for the j-th class for the sample vector $\mathbf{x}_i$ and weight vector $\mathbf{w}$ as

$$\mathbb{P}(y_i = j \mid \mathbf{x}_i) = \frac{e^{\mathbf{x}_i^T \mathbf{w}_j}}{\sum_{k=1}^7 e^{\mathbf{x}_i^T \mathbf{w}_k}}$$

Denote $\hat{\mathbf{y}}$ as the output of the network, the categorical cross-entropy loss is

$$\ell(\mathbf{w}) = -\sum_{i=1}^N \sum_{k=1}^7 y_k^{(i)} \log\left(\hat{y}_k^{(i)}\right)$$

4.4.2 Multi-class SVM Classification

Although a softmax cross-entropy loss is most widely used in training CNNs in face recognition tasks, the conventional softmax layer can suffer from high inter-class similarity and intra-class variation that characterize FER2013 [5]. Thus, inspired by the work of Tang in 2015, we replaced the softmax objective with the SVM objective [8]. The unconstrained optimization problem with categorical hinge loss can be written as

$$\ell(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^N \max(1 - \mathbf{w}^T\mathbf{x}_i, 0)$$

where C is the regularization constant. Differentiating the SVM objective with respect to the penultimate layer activation $\mathbf{h}$ gives

$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{h}_i} = -C\mathbf{w} \cdot \mathbb{I}_{\{1 > \mathbf{w}^T \mathbf{h}_i\}}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function.

5 Results

We reported our test accuracy on the FER2013 test set in Table 1 above. Our ensemble model of baseline, VGG16, ResNet50, and SeNet50 with softmax classification layer achieved 72.2% test accuracy. The ensemble model of baseline, VGG16, ResNet50, and SeNet50 with SVM classification layer achieved 71.7% test accuracy. The best ensemble model that we obtained consists of 6 models (VGG16 with softmax, VGG16 with SVM, ResNet50 with softmax, ResNet50 with SVM, and SeNet50 with softmax, and baseline model with softmax classification). It achieved a test accuracy as high as 72.9%. The best performance of the individual network is achieved by ResNet50 at an accuracy of 68.5%. Thus, we saw a significant improvement made by the ensemble modeling from individual performance. For reference, the highest reported test accuracy on the FER2013 dataset in the literature is 75.2% [6].

Model	Depth	Test Accuracy (softmax)	Test Accuracy (SVM)
Baseline	5	67.5%	67.5%
VGG16	16	68.3%	66.9%
ResNet18	18	43.4%	38.7%
ResNet50	50	68.5%	69.1%
SeNet50	50	62.4%	63.5%
Ensemble	-	72.2%	71.7%
Ensemble (best)	-	**72.9%** (softmax+SVM)	

Table 1: Summary of model accuracy

Potentially due to the complexity and depth of the pre-trained models, there were signs of overfitting in the training process. The model Resnet50 quickly overfits the training dataset at the beginning of 20 epochs even though we added a 50% dropout rate. One potential reason for the jittery validation loss curve is the relatively small size of our dataset. Another reason is that the learning rate of the SGD optimizer is too high (we set it to be 0.001, with momentum 0.9). We could improve upon our current result by varying the batch size and adding in the regularization term. We could also perform data augmentation to increase the size of our dataset and achieve a more robust model.

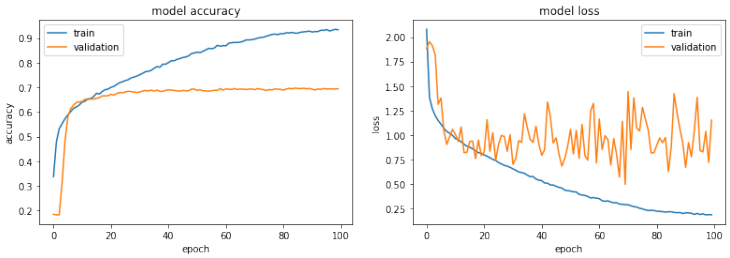
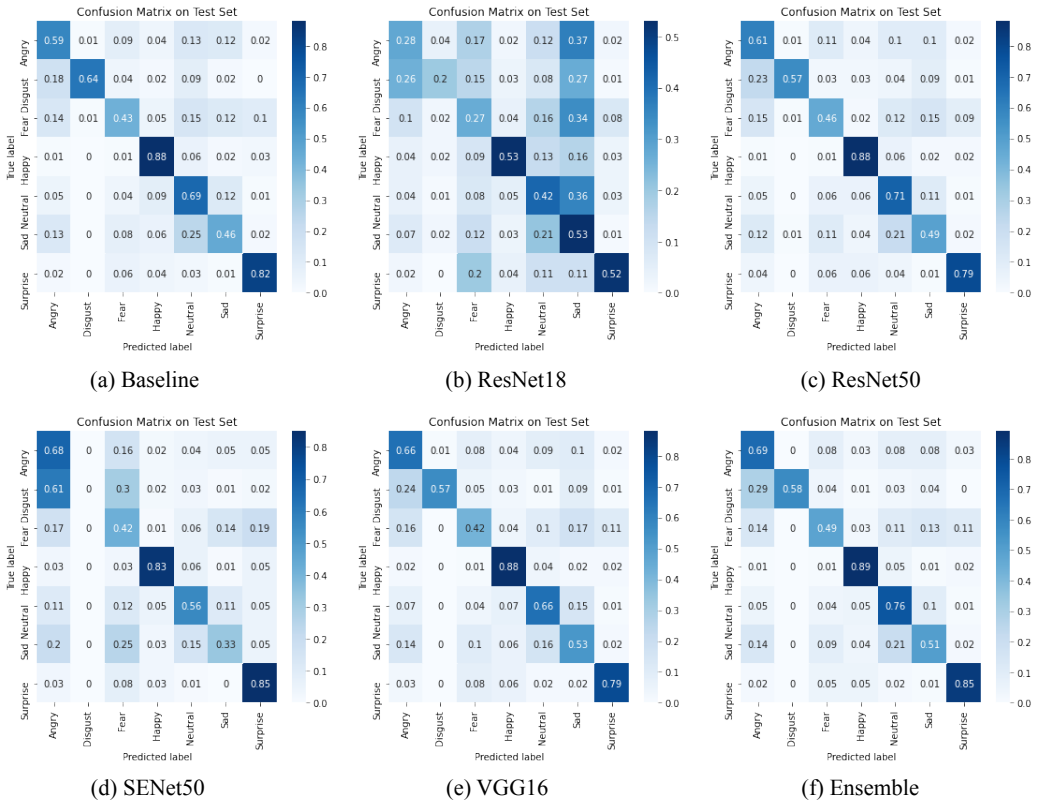


Figure 5: Resnet50 with softmax output accuracy and loss curve

6 Discussion and Analysis

Our ensembling method has proven to be able to obtain significant performance gain. However, during the experiment, we found ensembling some combination of networks (Baseline, ResNet50,



(a) Baseline (b) ResNet18 (c) ResNet50

(d) SENet50 (e) VGG16 (f) Ensemble

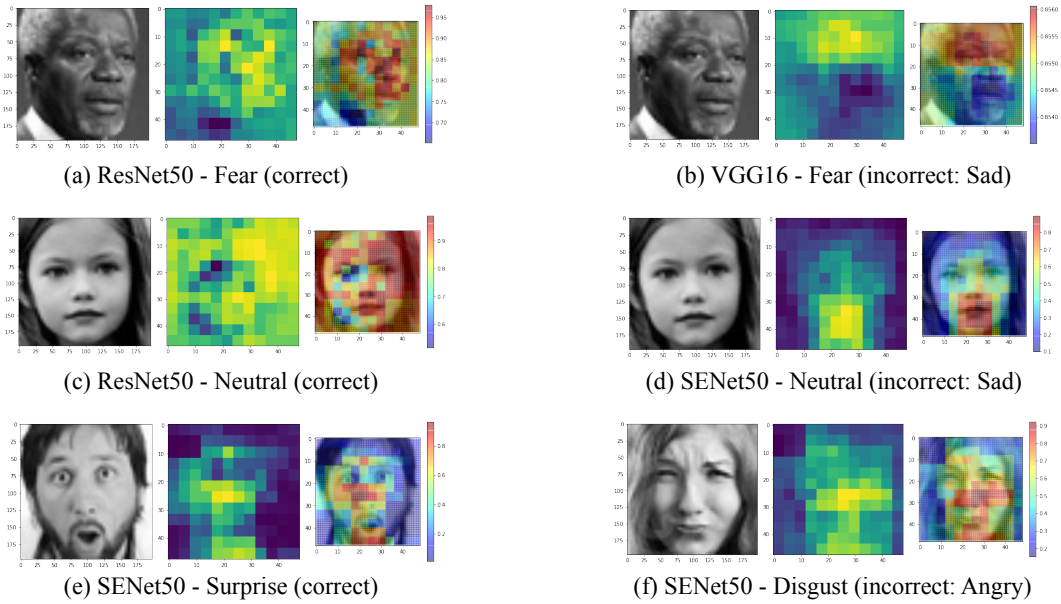
Figure 6: Confusion matrices of 5 models and the ensemble model (baseline, ResNet50, SENet50, VGG16)

SENet50, VGG16) effective while some others that exclude ResNet50 or SENet are not. Thus, it is worth exploring the reason behind this improvement. For error analysis, we relied on confusion matrices for misclassification rates for all of our models. Then we focused on the misclassified examples and used occlusion-based saliency maps to analyze and interpret the behavior of a specific network model.

6.1 Confusion Matrix

A confusion matrix is a common table layout for visualizing the classification performance of supervised learning algorithms. Since the multi-class classification problem in this experiment has 7 categories, all of our confusion matrices are $7\times$ 7 with rows representing predicted class and columns representing the true label. A confusion matrix with large numbers (close to one) along and small numbers (close to zero) anywhere else indicates high classification accuracy of a model.

Figure 6 shows the confusion matrices of all 5 models and the ensemble model (baseline, ResNet50, SENet50, VGG16). We noticed that the Happy category achieves the highest accuracy score across all models and the Fear and Sad categories achieve the lowest. We also made an interesting observation that ResNet50 complements with SENet50 in categories like Neutral: ResNet50 achieves an accuracy of 0.71% while SENet has only 0.56% accuracy. On the other hand, SENet50 scored 0.68% in classifying Angry faces for which the baseline model had a less desirable performance at 0.59% accuracy. The confusion matrix of the ensemble model shows that it achieved an accuracy of at least the max percentage of individual models in most of the categories and even improved 1-2%. We think this explains why ensembling this combination of the baseline model, ResNet50, SENet50, and VGG16 turned out to be more effective than others.



(a) ResNet50 - Fear (correct)

(b) VGG16 - Fear (incorrect: Sad)

(c) ResNet50 - Neutral (correct)

(d) SENet50 - Neutral (incorrect: Sad)

(e) SENet50 - Surprise (correct)

(f) SENet50 - Disgust (incorrect: Angry)

Figure 7: Saliency maps superimposed on original face images to highlight different facial regions weighed by models, which allows better interpretability of which features are more prone to errors

6.2 Occlusion-based Saliency Map

We used occlusion-based saliency maps to identify the facial region that our models are most sensitive to in classification. We systematically occluded fixed-size regions of the input image and observe the output probability p_i of the correct class of the modified image by the model [9]. Then we recorded $1 - p_i$ by the occluding region i on a heatmap. We subtracted the probability from 1 to have higher value represent higher sensitivity since occluding that region leads to smaller probability of outputting the correct class. Finally, we normalized the values on the heatmap to $[0, 1]$ and visualized the result using a colormap. Figure 7 shows sample occlusion-based saliency maps of correctly classified and misclassified images.

First, all of our models have primary focuses on eyes (including eyebrows), mouth, cheekbones, and chin bun. These are consistent with how humans detect emotions by observing the change in other's eyes and mouth etc.

Then we looked at the Fear and Sad categories that our models generally achieved low performance. By observing and comparing examples like (a) and (b) in Figure 7, we hypothesized that models that extract global features of faces tend to perform better than models that pay "attention" to specific local features in classifying negative emotions. We realized that it is more challenging to classify negative emotions since there are 4 of them (Angry, Disgust, Fear, Sad) versus only 1 (Happy) positive emotion. Indeed, from the perspective of a human observer, Angry, Disgust, Sad, and Fear faces are likely to have lower eyebrows and corners of the mouth drawn downwards. Angry and Surprise faces often have open mouths and bulged eyes. Therefore, as in (b) of Figure 7, models extracting only eyes feature is likely to confuse a Fear face with a Sad face.

We were particularly interested in pairs of models that "complement" each other in performance in some categories as indicated by the confusion matrices. Again, we attributed the misclassification to the difference in detecting global and local features. For example, as (c) and (d) in Figure 7 show, SENet50 only focused on the downward mouth corners while ResNet50 paid extra attention to eyes and cheekbones.

SENet50 appeared to be good at extracting local facial features, and we wondered if this affected classification of all categories of emotion. While local features were suboptimal in most categories, SENet had the best grasp of Surprise faces. We think a more conclusive correlation can be made

between local feature detection and misclassification only after such an experiment being performed on more network models. We will leave this as a part of future work.

7 Conclusion and Future Work

We explored the shallow CNN model (baseline) and deep neural networks including ResNet18, ResNet50, SENet50, and VGG16. Our ensemble model has proven to be effective in significantly improving the performance from 68.5% (maximum accuracy achieved by individual model) to 72.9%. We evaluated each of our models using the confusion matrix and occlusion-based saliency map. These techniques have provided better interpretability of the behaviors of the models. We explained the improvement of the ensemble model with individual models complementing each other, and we found models attending to global facial features generally perform better. Meanwhile, we recognized that it is more challenging to differentiate negative emotions due to more categories than positive and neutral emotions.

In terms of future work, we wish to improve data augmentation and try some new network models and losses. One problem in the training process is that deep networks like SENet50 quickly overfit the training data. Data augmentation can mitigate this issue by slightly modifying existing data to produce new synthetic data and acting as a regularizer. We would also like to extend our work using Siamese Network (SNN) and triplet loss. SNN contains two or more "identical" subnetworks sharing the same weights and parameters. It is more robust to class imbalance. It can be further nicely ensembled with other supervised classifiers since SNN learns semantic similarity, which is very different from the features learned by conventional classifiers. Triplet loss is a type of contrastive loss that emphasizes inter-class difference. It aims to minimize the distance of embeddings between similar instances and maximize the distance otherwise under some metric.

References

[1] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age, 2018.

[2] C. Darwin and P. Prodger. The expression of the emotions in man and animals, 1998.

[3] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. *CoRR*, abs/1411.7964, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *IEEE Transactions on Affective Computing*, page 1–1, 2020.

[6] Christopher Pramerdorfer and Martin Kampel. Facial expression recognition using convolutional neural networks: State of the art, 2016.

[7] C. Shan, S. Gong, and P. W. McOwan. "facial expression recognition based on local binary patterns: Acomprehensive study," image and vision computing, 2009.

[8] Yichuan Tang. Deep learning using linear support vector machines, 2015.

[9] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

[10] G. Zhao and M. Pietikainen. "dynamic texture recognition using local binary patterns with an application tofacial expressions," ieee transactions on pattern analysis and machine intelligence, 2007.

[11] R. Zhi, M. Flierl, Q. Ruan, and W. B. Kleijn. "graph-preserving sparse nonnegative matrix factorizationwith application to facial expression recognition," ieee transactions on systems, man, and cybernetics, part b(cybernetics), 2011.

[12] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. "learning active facial patches for expression analysis," in computer vision and pattern recognition (cvpr), 2012.