Facial Expression Recognition
Improving on the State of the Art

Abbas Mammadov **20200844** · **Kaleb Mesfin Asfaw** **20200805** · **Zahra Bayramli** **20200812** · **Tivan Varghese George** **20200829**

School of Computing
Korea Advanced Institute of Science and Technology

1 Introduction

Humans have many avenues through which they transfer information; one such avenue being facial expressions. Facial expressions can reveal a lot of hidden information, including a person's emotional state, which can be used within contexts such as psychological assessment or legal interrogation.

Considering the importance of human-computer interaction, a facial-expression emotion classifier becomes increasingly more desirable. Even though there already exist several high-accuracy models that distinguish simple expressions and emotions in simple and controlled settings, there is still room to improve in models that aim to identify complex emotions based on facial expressions, to a high-degree in a real world setting. As such, in this project, we aim to create an efficient model that classifies emotion based on facial expressions in a real world setting. This means that our goal is to create a model that can accurately classify emotion independent of age, ethnicity, and appearance of a person. This model can potentially be of great service in the medical, psychiatric, legal, and leisure industries due to the planned generality of its classifications.

2 Related Works

There is plenty of existing research in the FER domain. In particular, a recent paper by In-Kyu Choi, Ha-eun Ahn and Jisang Yoo [8] "Facial expression classification using deep CNN" introduces a model with accuracy of 93.95% and short execution time. They have proposed Facial expression recognition using CNN and used 10 different datasets to be classified into 6 expressions. They have evaluated the accuracy of the proposed deep neural network architecture in two different experiments such as cross validation and cross database. In the cross validation experiment, the K-fold cross validation technique has been used, which ensures the generalizability of classifiers.

Another paper by Pramerdorfer and Kampel [9] describes the approaches taken by six current papers and ensembles their networks to achieve 75.2% test accuracy on the FER2013 dataset.

One of these 6 papers is "Learning Social Relation Traits from Face Images" by Z. Zhang, P. Luo, C.-C. Loy, and X. Tang [10]. They have formulated a new network architecture with a bridging layer to learn a rich face representation. The method achieves performance of 75.10% accuracy, through fusing data from multiple sources.

3 Datasets

We have used the FER2013 [2] dataset which was collected automatically by the Google image search API. This large-scale data, which is available on Kaggle, has been studied well and played a crucial

,

role in various research papers. The dataset consists of 35887 grayscale images of faces, which have been normalized to the $48x48$ pixel scale. In FER2013, images of 7 facial expressions has been distributed as follows: Angry (4953), Disgust (547), Fear (5121), Happy (8989), Sad (6077), Surprise (4002), and Neutral (6198). We divided the dataset into train, validation, and test sets with ratio $80 : 10 : 10$ (train - 28709, validation - 3589, test - 3589). During the training, we did data augmentation to increase the amount of data by adding slightly modified copies of existing images.

In order to increase the accuracy of our model, we have also added 2 more auxiliary datasets, CK+ [4], and affectnet [3]. The CK+ dataset contains samples from subjects of varying age (18-50 years old), ethnicity, and gender. The samples are of two different resolutions (both of which are vastly higher than that of the FER2013 dataset), but each subject seems to be in a similar setting. There are both black and white as well as color samples in this dataset. The affectnet dataset contains pictures with subjects of varying gender, age, and ethnicity in several real-world settings (different lighting, backgrounds, etc.). The pictures also have varying resolution. The variety of the samples from both datasets will greatly increase the accuracy, and hence usability, of our model in real life settings.

4 Methods

Resnet-50
First, we decided to tackle this problem by using ResNet-50 with various settings. ResNet-50 is a convolutional neural network (CNN) with 50 layers depth. Our various trial-and-error attempts lead us to a conclusion that Resnet-50 layers, being best suitable for this task, operate better with a batch size of 128 and using Adam optimizer, rather than Stochastic Gradient Descent. We have got 62.1% accuracy using a ResNet-50 model when we trained it using the SGD optimizer. This accuracy has increased to 64.9% when we use the Adam optimizer. We have also trained it on the ResNet 50 model where we have added 3 fully connected layers, and 3 dropouts, and got 62.21% accuracy.

VGG-19
VGG-19 is another pre-trained model that we have explored and used in our problem. Being shallower than ResNet-50, it is a convolutional neural network, with 19 layers, pretrained on more than a million images from ImageNet database. It has generally been used in diverse and complex classification tasks. In our implementation, we have replaced the last layer with a fully connected layer of 4096 inputs and 7 outputs to make it suitable with our multi-class classification. We have trained it with 50 epochs, Adam Optimizer, and got a 60.9% accuracy on the testset.

EfficientNet-B7
Efficient Net is one of the most common CNN models which are used on such facial expression related deep learning projects. We have used PyTorch's pretrained efficientNetb7 model. Since we are dealing with a 7 class classification, just like we did for the other models, we have modified its last layer to have an output of 7. We have tried training it using both SGD, and Adam optimizers, and we found out that Adam's optimizer gives better accuracy. After training it for 50 epochs, we have got an accuracy of 65% on the test dataset.

5 Experiments, Results, and Discussion

As we stated in our model section, we used ResNet-50 as our first pre-trained model (baseline). ResNet-50 in PyTorch has been pretrained on more than one million images of the 'ImageNet' dataset.

5.1 *Trial - 1: Using only FER2013*

As our first stage of our experiment, we have just used only the FER2013 dataset. Due to this dataset we were exposed to some factors which prevented us from achieving high accuracy. To mention some, this FER2013 dataset consists of about $35,000$ images, which is a bit smaller compared to the well known huge sized datasets that contributed to the boosting of accuracy of many models. It's also not well balanced. For instance, the number of the images labeled as "disgust" is almost incomparable with respect to the other emotions (more on this, later). Moreover, the fact that the dataset is entirely composed of grayscale images poses an obstacle to our model to perform well on the real world images.

Preprocessing
The dataset which is available on kaggle contains a csv file of the numerical pixel values of the images. Since we want to visualize, and apply some image augmentation technique, we have converted the pixel values to jpg images using pytorch's inbuilt functions. However, for model building, we have converted the images into pixels of normalized tensors.

Model Training
After Separating our dataset into train, validation and test sets with proportions of $80\%, 10\%, and 10\%$ respecitvely, for this model, we have only used Resnet50 (by modifying the last layer to have output of 7), and VGG19. The best accuracy we got was 0.61 on the validation dataset by training our model with batch size of32, and epoch 50. Then, to maximize our accuracy, we utilized the concept of transfer learning here, and we have added 2 fully connected layers, each with sizes of 1024 and 4096 on our 'learned' Resnet50 layers. After training for 50 epochs, with a batch size of 128, we got 64% accuracy on the validation dataset. And we have got 62.6% accuracy from the VGG19 model. However, from our loss curves we have realized that our models were overfitting on the training dataset. The reason for our model to quickly overfit to the training dataset is due to the fact that the FER2013 dataset is quite small. So we need to have more data so as to enable our model to generalize facial features.

5.2 *Trial - 2: FER2013 + CKplus + AffectNet*

Preprocessing
CKPlus and AffectNet datasets were composed of image files (unlike the FER2013 dataset which is composed of numerical pixel values). However, we have seen that some of the images from the AffectNet dataset have been placed in the incorrect folder of their labels. Therefore, we have used a pseudo-labeled csv file which contains their correct labels, and we re-classified the images in that manner. In order to alleviate the overfitting problem we have faced with our earlier trial, one solution we have proposed was getting more data. Hence, in addition to including auxiliary datasets, we have utilized torchvision's built-in function, "Transforms" to apply some transformation techniques on our datasets, like Random horizontal flips, random rotations, and random translations.

Model Training
After adding auxiliary datasets and applying data augmentation techniques, we divided the whole data into train, validation and test sets with the same proportion as our previous setting (80 : 10 : 10). Then, we trained 4 CNN models and recorded the results as follows.

Model	Optimizer	Accuracy %
Resnet-50	SGD nesterov	62.14
Resnet-50	Adam	64.90
Resnet-50 (TL)	Adam	62.21
EfficientNet-B7	Adam	**67.00**
VGG-19	SGD nesterov	62.60

Table 1: Results of different models. The best results among all models are highlighted in **bold**. *TL* stands for Transfer Learning.

Error Analysis
We also tried to analyze the prediction of our model towards each emotion. In every iteration, we save the best values of our parameters when we detect a drop in the loss value. After saving the best values of the hyper parameters, we load them to get their prediction values so as to construct the confusion matrix.
As we can see from the figure on the left, when using only the FER2013 dataset, only few amount (about 26%) of actual 'disgust' images were correctly classified as 'disgust', and for our surprise, there is no any emotion (except very few 'angry' and 'fear' emotions) which has been misclassified as 'disgust'. This indicates that the model hasn't yet grasped the

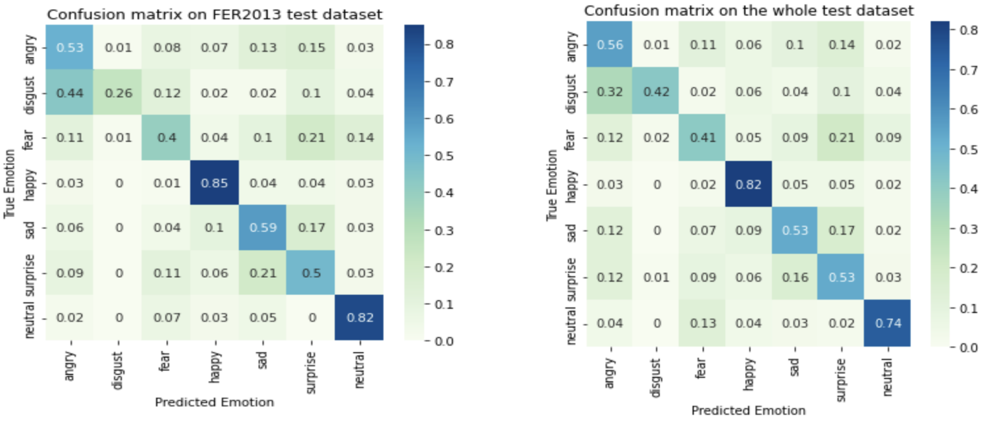

Figure 1: Confusion Matrices before and after adding auxiliary datasets

behavior of ‘disgust’ images. As we have discussed earlier, this is due to the presence of a small amount of images labeled as ’disgust’.

Now, after adding the auxiliary dataset and training our models on the whole dataset, we can see (from the confusion matrix given on the right) that the fraction of pictures which has been correctly classified to ‘disgust’ label has increased from 26% to 42%. This is due to the fact that the added auxiliary dataset provided the model with a higher exposure to the label, and hence, we got an increase in the accuracy of our model.

6 Conclusion

In this project, we examined various CNN architectures which we thought to perform well on such datasets. In order to combat the issue of class imbalance in the FER2013 dataset, we have added auxiliary datasets, and applied data augmentation techniques. In the data preprocessing stage, we have detected some incorrectly labeled images from affectnet, which triggered us to perform a thorough data cleaning. We have indeed proved that ResNet 50 and EfficientNet models are well suited for facial recognition related tasks. And from the error analysis we have made, we have concluded that in order for our model to make accurate predictions for all the labels, it’s best if our dataset contains a large (to prevent overfitting), and a balanced number of labels as much as we can.

However, the biggest challenge we have faced was a limit in our computational resource. We couldn’t train our models for a larger number of epochs (above 50). The loss curve of some of our models had a bit of bumps in between them, but in the overall they did not stop declining. This indicates that if we had trained our model for a higher number of epochs, the loss would have declined further. Also to train a single model, we need to wait for an extensive amount of time. Therefore, if we had given more time and computational resources, we would increase the number of epochs to investigate the convergence of the loss curve of our model. We would also tweak the hyperparameters even more than we did at this moment, trying out several possibilities to obtain even higher accuracy.

7 Contributions

Name	Contributions
Abbas Mammadov	Model building and training- Resnet50 Transfer Learning models Data pre-processing (AffectNet) LaTeX formatting for the reports
Kaleb Mesfin Asfaw	Model building and training- EfficientNet Error-Analysis Data pre-processing (CK+)
Zahra Bayramli	Model building and training- VGG19 Data Augmentation Hyperparameter Tuning
Tivan Varghese George	Data pre-processing (FER2013) Data Augmentation Hyperparameter Tuning

Table 2: Results of different models. The best results among all models are highlighted in **bold**. *TL* stands for Transfer Learning.

References

[1]https://github.com/abbasmammadov/Facial-Expression-Recognition

[2]https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[3]https://paperswithcode.com/dataset/affectnet

[4]https://www.kaggle.com/datasets/shawon10/ckplus

[5]https://arxiv.org/pdf/1804.08348.pdf

[6]https://arxiv.org/pdf/1512.03385.pdf

[7]https://arxiv.org/pdf/1409.1556.pdf

[8]https://www.koreascience.or.kr/article/JAKO201809253681042.pdf?fbclid=IwAR3gnwoUoE$_{a}rj0gKlfp8uM4Ih92eIwMcpChEw9CK$

[9]Pramerdorfer, C., Kampel, M.: Facial expression recognition using convolutional neural networks: state of the art. Preprint arXiv:1612.02903v1, 2016.

[10]Z. Zhang, P. Luo, C.-C. Loy, and X. Tang, “Learning Social Relation Traits from Face Images,” in Proc. IEEE Int. Conference on Computer Vision (ICCV), 2015, pp. 3631–3639.