Fine-tuning an Image Classification Model to Perform Facial Expression Recognition

Christopher Lewis
Department of Computer Science
University of Massachusetts Lowell
Christopher_Lewis1@student.uml.edu

***Abstract*—In recent years, image classification models have become increasingly popular in the field of deep learning and have been used for a wide variety of applications. In particular, the CNN architecture and its variants have been responsible for producing some of the most successful deep learning image classification models. However, these models require large amounts of training data and computational resources to achieve reasonable results. As a result, it has become standard practice to fine-tune pre-trained models on specific datasets for a downstream task. In this work, a ResNet model that was pre-trained on as subset of the ImageNet dataset is fine-tuned on a human emotion dataset to perform facial expression recognition. This model achieves a testing accuracy of 70.16% and it is shown that it achieves higher accuracies than several baseline models and also performs better than the estimated human performance.**

***Index Terms*—CNN, ResNet, Image Classification, Deep Learning, FER-2013, ImageNet, Emotions**

I. Introduction

The CNN (convolutional neural network) architecture and its adaptations have shown to achieve strong performance on image classification problems, such as human emotion detection and classification. This success is largely due to the convolution and pooling layers that are used in the CNN architecture. The architecture was designed such that the connections between neurons resemble the visual cortex in animals, where certain neurons only respond to stimuli within a receptive field, which is a specific region of the visual field.

In order for CNN models to achieve competitive results, it is necessary that they are trained on large amounts of data. However, training such a model from scratch is very computationally expensive. Pre-trained models help address this problem, because they have already been trained on large amounts of data that is related to the target task. As a result, pre-trained models can be fine-tuned using a relatively small number of examples (compared to the number of examples used during pre-training) from a dataset of choice, to achieve strong results.

A. Problem Statement and Goals

In this paper, two pre-trained ResNet [1] models, specifically the ResNet-18 and ResNet-50 models, are fine-tuned using the FER-2013 [3] dataset to perform human emotion classification. The models were pre-trained using the ImageNet-1k [2] dataset. This model accepts a valid image of a human exhibiting one of seven emotions and returns a prediction of which emotion the human is most likely experiencing.

The fine-tuned model achieves a testing accuracy of 70.16% and it is empirically shown in this paper that the fine-tuned ResNet-50 model outperforms several baseline models, as well as estimated human performance, in the task of human emotion detection using the FER-2013 dataset.

B. Motivations

There are several motivators for training a model that detects and classifies human emotions. Such a model can help identify individuals that exhibit concerning behaviors, such as depression and violence. For example, the expression on a person's face may indicate that they are about to attack another person. Security cameras that utilize an emotion detection model will potentially identify these expressions and notify security personnel.

Another motivating factor in training an emotion classification model is that it will offer deeper insights into human emotions. For example, the model results can help better our understanding of why certain emotions are classified more easily than others. On the other hand, the results can also aid in understanding why specific emotions are mis-classified as certain emotions more frequently than others.

II. Related Work

A. ResNet

A residual neural network (ResNet) is a variation of a convolutional neural network (CNN). The ResNet architecture offered improvements to existing CNN architectures by introducing the concept of skip connections. Skip connections skip a certain number of the layers in the network and feeds the output to the layers that were skipped to.

One of the main benefits of skip connections is that they help reduce the vanishing gradient problem that was commonly experienced before ResNet was introduced. As a result, ResNet enables the creation of deeper networks architectures without experiencing the vanishing gradient problem, thus allowing for more accurate models.

B. Facial Emotion Recognition Using Transfer Learning in the Deep CNN

M. Akhand et al. [4] fine-tuned the ResNet-18, ResNet-34, ResNet-50, and ResNet-152 models using the KDEF [5] and JAFFE [6] datasets. This paper demonstrated how strong fine-tuned models perform in the task of facial emotion recognition.

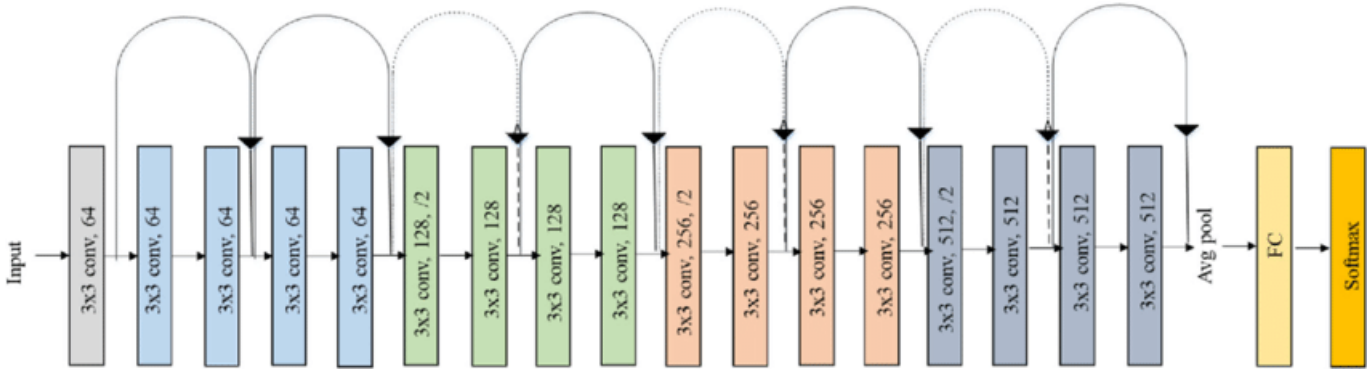

Fig. 1. ResNet-18 Architecture

It also offered useful comparisons between the variations of models that were used. For the ResNet models, all four achieved very similar classification accuracies, only varying by $\pm 2\%$ at the most. Despite this small difference in performance, the ResNet models with deeper architectures consistently performed better than those with shallower architectures (i.e. ResNet-152 outperformed ResNet-18).

Although the results achieved in [4] were strong, the datasets contained images with relatively no noise or variation, which is unrealistic in a real-world setting. This paper uses the FER-2013 dataset, which introduces much more noise and variation, which makes it a much more realistic dataset to use for model evaluation.

C. Facial Emotion Recognition: State of the Art Performance on FER-2013

Y. Khaireddin and Z. Chen [7] fine-tuned a VGG network [8] on the FER-2013 dataset. The results were evaluated against the baseline models and state of the art models trained on the FER-2013 dataset. This paper's fine-tuned model reported the strongest single-network classification accuracy at 73.28%, compared to existing state of the models. It also offers strong analyses on the learnability of the FER-2013 dataset and why it is difficult to achieve high performance on the testing set.

Overall, [7] demonstrated the effectiveness of the pre-trained VGG network on the FER-2013 dataset. However, the state of the art models are very large and require a great deal of computational resources to achieve strong results. This paper demonstrates that comparable results can be attained using a smaller, more manageable model.

III. Proposed Approach

A. FER-2013 Dataset

The dataset used to fine-tune the ResNet models is the FER-2013 dataset. This dataset consists of 48x48 pixel grayscale images of human faces. Each face has been automatically registered so that it is centered and occupies the same amount of space as the other faces. The training dataset consists of $28,709$ examples and the public testing dataset consists of $3,589$ examples. Each image belongs to one of seven emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

Fig. 2. Example images from the FER-2013 dataset

Fig. 2. shows four example images from the FER-2013 dataset. As shown in Fig. 2., the images in this dataset offer minimal information and the images from different categories often look very similar. This is one of the reasons why FER-2013 is more difficult to use for classification tasks, compared to other facial expression datasets [5] [6]. The relatively small number of pixels in each image makes it difficult for CNN models to generalize, especially because they are grayscale images. Additionally, the images contain faces from a wide variety of people in terms of age, gender, and ethnicity, which also makes it more difficult for CNN models to generate meaningful abstractions.

B. Pre-trained ResNet-18 and ResNet-50 Models

Fig. 1. shows the ResNet-18 architecture. In total, there are 18 layers in the network, with 17 convolutional layers, a fully-connected layer, and a softmax layer that is used to calculate class probabilities for classification purposes. The convolutional layers use 3x3 filters and downsampling is performed in the convolutional layers with a stride of 2. Also, filters are doubled if the output feature map is divided in half.

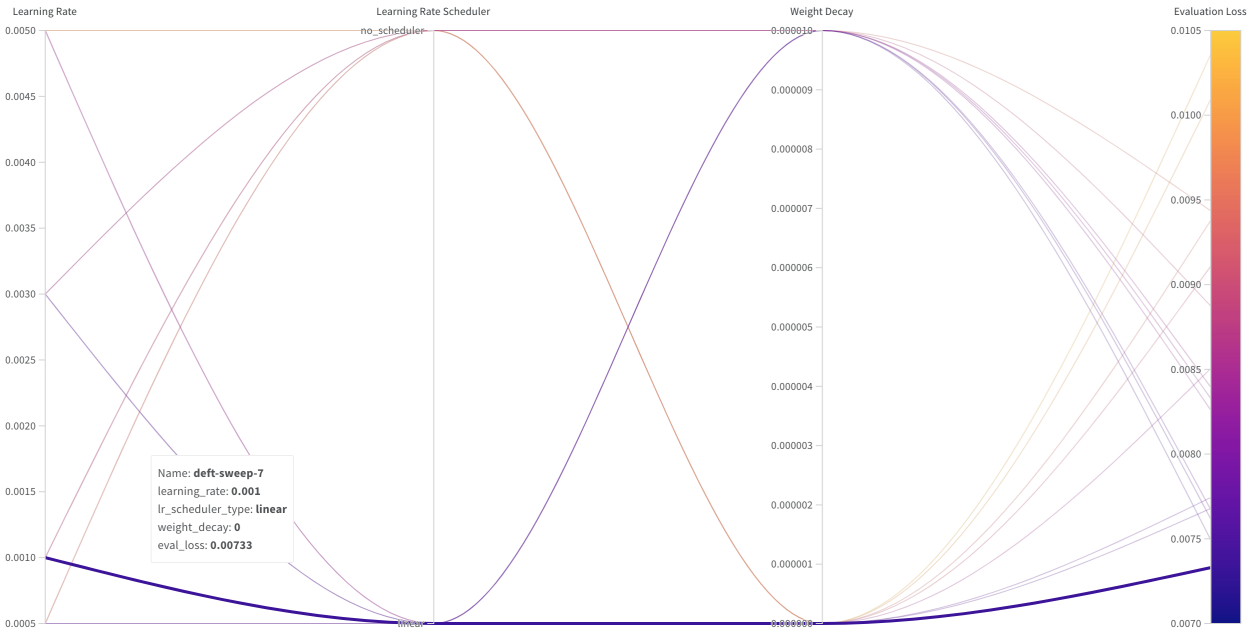

Fig. 3. Grid-search sweep results

The ResNet-50 model has an identical architecture, except it has 49 convolutional layers instead of 18.

In Fig. 1., two types of connections between layers can be seen. The solid lines between layers indicate connections that are used when the input and output have the same dimensions. The dotted lines between layers indicates connections that are used when dimensions increase.

C. Data Preprocessing

As shown in Fig. 2., the FER-2013 dataset consists only of grayscale images, meaning each image only has one channel. Because the ResNet-18 and ResNet-50 models were both pre-trained on datasets consisting of color images, which have 3 channels (RGB), the grayscale images were converted to 3 channels. This was achieved by duplicating the single-channel images 3 times, meaning the RGB channels all consisted of the same grayscale pixel values.

Before the images were fed to the ResNet models during training, a series of transforms was applied to each to increase training efficacy. These transforms consisted of random resizing crops, random horizontal flips, and normalization according to the pre-trained image mean and standard deviation values. Applying transforms allows the model to see more variations of images in the training dataset, making it more robust. This is especially important for the FER-2013 dataset, because its images are only 48x48 in size.

D. Fine-tuning and Training

The public testing dataset's loss was used as the metric to assess how well a set of hyperparameters performed during training of the ResNet models. A categorical cross entropy loss function was used for all experiments.

Training Process Overview: To fine-tune the ResNet models, the following process was followed in order to determine the optimal experimental setup:

1) Demonstrate that the model can perfectly fit a small subset of the training data.
2) Find an initial, reasonable set of hyperparameters to use that achieve the best accuracies after a relatively small number of steps (10 epochs with a batch size of 64 in this case). Important parameters to consider:
    a) External learning rate and external weight decay.
    b) Optimizer and learning rate scheduler.
3) Fine-tune the initial parameters from part 2 even further, using a larger number of steps (15 epochs with a batch size of 128). This was achieved with the help of the following:
    a) Use sweeps to test different combinations.
    b) Bayesian search, followed by grid search.

Hyperparameter Selection: During step 2 from the training process detailed above, the optimizer to use was first determined. The optimizers tested were: SGD, AdamW, and AdaFactor. They were each tested using a fixed learning rate of 0.0005, with no learning rate scheduler, for 10 epochs. The AdaFactor optimizer consistently outperformed the others, so it was selected as the optimizer.

Next, learning rates of 0.0001, 0.0005, 0.001, 0.003, and 0.005 were tested using a Bayesian hyperparameter search (learning rates outside of this range consistently performed poorly). External weight decays of 0.0 (no decay) and 0.0001 were included in the search. Of this set of parameters, a learning rate of 0.001 with no weight decay and a linear

learning rate scheduler performed the best, after 130 runs.

Based on the results of the Bayesian search, a grid search was performed on a smaller subset of the 16 best learning rate and weight decay parameter combinations. These results can be seen in Fig. 3. In this search, the learning rate scheduler type was also included as a parameter. A fixed learning rate (no scheduler) and a linear learning rate were tested. Once again, the best parameter combination found was a learning rate of 0.001 with no external weight decay, using a linear learning rate scheduler with the AdaFactor optimizer. As a result, these parameters were used for the main experiments.

IV. Experimental Results and Discussions

For the experimental results, the hyperparameters found in Section 2 were used for training. Both a ResNet-18 and a ResNet-50 model were trained for 30 epochs, with batch sizes of 128 and 64, respectively. The ResNet-50 model required a batch size of 64 due to memory limitations.

The ResNet-18 model achieved nearly identical results to the ResNet-50 model, which achieved a test set classification accuracy that was only 0.2% better than the ResNet-18 model. Because of how similar the results are, only the results from the ResNet-50 model are included in this section.

Although the ResNet-50 model was trained for 30 epochs, the weights used for the final model were the weights that achieved the highest evaluation accuracy on the testing dataset. The best accuracy was achieved at step $9,990/15,660$. It is worth noting that the "evaluation" accuracies mentioned in the results refer to the public FER-2013 test set.

A. Results

The training and evaluation accuracies for the ResNet-50 model can be seen in Fig. 4. It can be seen that around step $8,000$ the training accuracy continues to increase at a steady rate, while the validation accuracy does not. This indicates that the model is no longer learning to generalize to unseen examples and indicates that overfitting is likely to occur. In

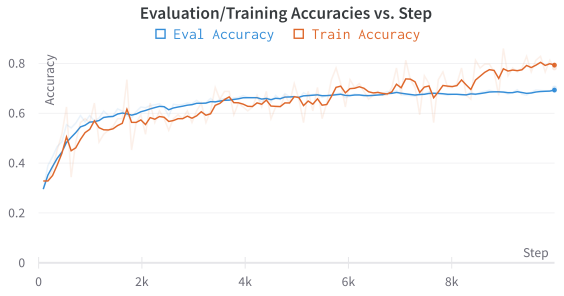

Fig. 4. Evaluation (test set) and training accuracies for the ResNet-50 model after 9990 steps with a batch size of 64

Fig. 5. the evaluation loss is shown over the course of $9,990$ training steps. It can be seen that the evaluation loss has stopped decreasing around step $8,000$, which indicates that the model is no longer generalizing to unseen examples, much like the lack of increase in the evaluation accuracies. The loss in Fig. 5. has been smoothed to account for the y-axis size and scale.

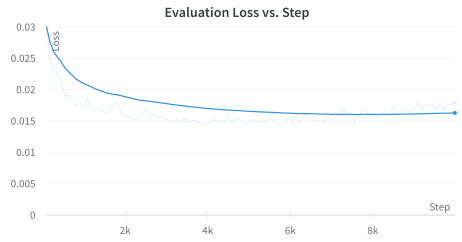

Fig. 5. Evaluation (test set) loss for the ResNet-50 model after $9,990$ steps with a batch size of 64

The confusion matrix in Fig. 6. offers insights into how well the ResNet-50 model classified each emotion in relation to the others. Higher values for a square indicates a higher percentage of predictions for that particular combination of $(predicted, actual)$ label pairs. Diagonal elements indicate when an emotion was correctly predicted.

The confusion matrix indicates that the model performed relatively well in classifying each emotion in relation to the others. This is shown further in Table 1, which includes the accuracy, precision, recall, and F-1 scores. However, it can also be seen that for certain pairs of emotions, the model consistently made incorrect predictions. For example, it is shown that the model often classifies an image as belonging to the "Angry" class, when in fact the image belongs to the "Disgust" class. This is likely because images from both of these classes tend to look very similar, as shown in Fig. 2.

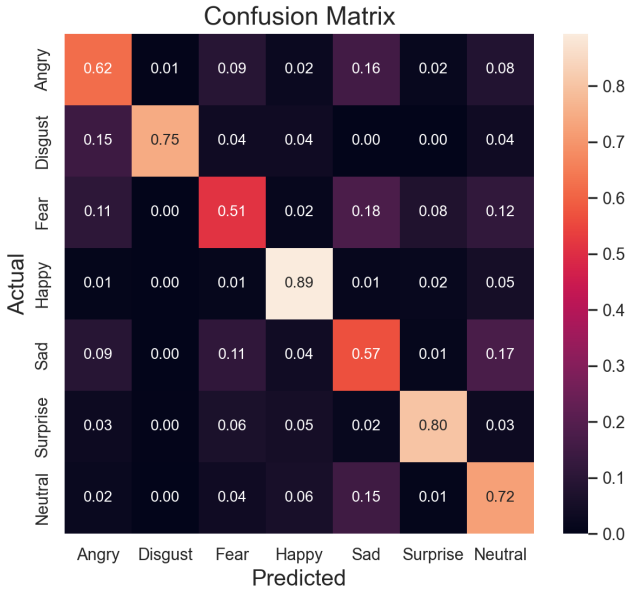

Fig. 6. Test set confusion matrix results, using the ResNet-50 model weights that achieved the highest accuracy

Accuracy	Precision	Recall	F1-Score
70.16	71.22	69.39	70.29

TABLE I

ACCURACY, PRECISION, RECALL, AND F1-SCORE ON THE TEST SET USING THE RESNET-50 MODEL WEIGHTS THAT ACHIEVED THE HIGHEST ACCURACY

B. Discussion

Table 2 shows the results of the fine-tuned ResNet-50 model compared to the baseline model results [7]. This paper's model outperforms all baseline models, after only training for $9,990$ steps. This demonstrates the efficacy of fine-tuning a pre-trained model and how doing so greatly reduces training time.

It can also be seen in Table 2 that the fine-tuned ResNet-50 model outperforms the estimated human performance [7] on the FER-2013 dataset. Not only is this impressive that this paper's model outperforms human evaluation, but it also demonstrates how difficult it is to classify images in this dataset.

Model	Test Accuracy
CNN [9]	62.44
GoogleNet [10]	65.20
Est. Human Performance [7]	65.50
VGG + SVM [11]	66.31
Conv + Inception Layer [12]	66.40
Bag of Words [13]	67.40
Attentional ConvNet [14]	70.02
ResNet-50 (This Paper)	**70.16**

TABLE II

BASELINE COMPARISON RESULTS ON FER-2013

V. CONCLUSIONS AND FUTURE WORK

Overall, this paper demonstrates the capabilities and effectiveness of pre-training. This is done by showing that the pre-trained ResNet-50 model outperforms the baseline models. It also shows that the model outperforms the estimated human classification performance on the FER-2013 dataset, which highlights both the efficacy of the model and the difficulty of the dataset. In terms of hyperparameter selection, this paper suggests that the AdaFactor optimizer with a linear learning rate schedule is preferred over the SGD and AdamW optimizers for these specific models and this task.

For future work, fine-tuning the ResNet-50 model on another facial expression recognition dataset before introducing it to FER-2013 would likely be a useful strategy to improve performance. One of the main difficulties of the FER-2013 dataset is that it only contains grayscale images, while ResNet models are almost always fine-tuned using color datasets. By introducing a color facial expression recognition dataset first, it would allow the model to ease into training more smoothly.

Large versions of the ResNet models, such as ResNet-152 can also be explored. However, this may have a negative impact because larger models often make it easier to overfit the training data. Finally, other fine-tuning methods such as annealing or dedicating a certain number of training steps as warmup steps may improve results as well.

REFERENCES

[1] K. He, X. Zhang, S.Ren, J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition, December 2015.

[2] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, June 2009.

[3] I. Goodfellow et al., "Challenges in Representation Learning: A report on three machine learning contests," ICML 2013 Workshop on Challenges in Representation Learning, July 2013.

[4] M. Akhand, S. Roy, N. Siddique, M. Kamal, T. Shimamura, "Facial Emotion Recognition Using Transfer Learning in the Deep CNN," Electronics 2021, 10(9), 1036, April 2021.

[5] Calvo, M.G.; Lundqvist, D., "Facial expressions of emotion (KDEF): Identification under different display-duration conditions," Behav. Res. Methods, 40, 109–115, 2008.

[6] Lyons, M.J.; Akamatsu, S.; Kamachi, M.; Gyoba, J.; Budynek, J., "The Japanese Female Facial Expression (JAFFE) Database," Coding Facial Expressions with Gabor Wavelets, February 2021.

[7] Y. Khaireddin and Z. Chen, "Facial Emotion Recognition: State of the Art Performance on FER2013," ArXiv, May 2021.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings, 2015.

[9] K. Liu, M. Zhang, and Z. Pan,"Facial Expression Recognition with CNN Ensemble," in Proceedings - 2016 International Conference on Cyberworlds, CW 2016, 2016.

[10] P. Giannopoulos, I. Perikos, and I. Hatzilygeroudis, "Deep learning approaches for facial emotion recognition: A case study on FER-2013," in Smart Innovation, Systems and Technologies, 2018.

[11] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and handcrafted features for facial expression recognition," IEEE Access, vol. 7, 2019.

[12] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016.

[13]] R. T. Ionescu, M. Popescu, and C. Grozea, "Local Learning to Improve Bag of Visual Words Model for Facial Expression Recognition," Work. challenges Represent. Learn. ICML, 2013.

[14] S. Minaee and A. Abdolrashidi, "Deep-emotion: facial expression recognition using attentional convolutional network," arXiv. 2019.