Facial Expressions Recognitions with Supervised Contrastive Learning

Xiaotian Wang*

Department of Computer Science, University of Wisconsin – Madison, Madison, Wisconsin,

* Corresponding Author Email: xwang2294@wisc.edu

Abstract. Facial expressions are significant ways that humans express and perceive emotions. Facial Expression Recognition (FER) can be challenging in computer vision since facial expressions tend to be different among individuals. Researchers have introduced many outstanding methodologies and approaches to achieve the best performance in classifying and identifying emotions based on images of facial expressions. Supervised Contrastive Learning (SCL), outperforming supervised learning with cross-entropy loss, has gained much attention due to its recent performance in computer vision. In this work, we extended traditional approaches to common cross entropy loss and made approaches of contrastive learning to tackle the issue. Specifically, the frameworks in this work focus on the performance of cross entropy loss and supervised contrastive loss on the FER-2013 dataset. The result showed that supervised contrastive loss substantially improved accuracy by about 2% to 6% for ResNet architectures.

Keywords: Facial Expression Recognition; Convultional Neural Networks; Supervised Contrastive Learning.

1. Introduction

Facial expression is a facial gesture that expresses the human's emotions, goals, and intentions in the communication process, playing a significant role in human interaction. Inspiring by the emotional expression study in 1872, expressions analysis from facial expressions has become an emerging topic in computer vision and pattern recognition [1]. The facial expression of individuals detected from an image could potentially differ significantly due to different facial appearances and image quality, making detecting facial expressions complicated in computer vision. Facial Expressions Recognition (FER) works aimed at classifying expressions on faces into categories, such as anger, disgust, fear, happiness, or sad. Similar to most common tasks in computer vision, the basis of FER models is also Convolutional Neural Network and classification. Taking images showing individuals' facial expressions as inputs, the model captures the features on the images and then classifies them.

FER2013 was designed by Goodfellow et al. It was provided for a Kaggle competition to encourage researchers to develop FER systems with high performance. The database comprises more than 30,000 pictures in 7 categories. Figure 1 shows a sample of angry expressions in the dataset. All images in the dataset were in greyscale, i.e., only one channel for each image. So, the size of each image in the dataset is 48 x 48 x 1. Most current FER systems for this dataset achieve relatively good performance using different approaches. Based on the research by Khanzada et al. [2], most published works on this dataset achieved accuracy ranging from 65% to 70%. The highest performance from published works got 75.2% test accuracy.

Figure 1. Sample of the dataset representing Angry and Happy

In 2020 Supervised Contrastive learning, introduced by Khosla et al., achieved top-1 81.4% accuracy on the ImageNet dataset, showing a consistently better performance than cross-entropy and other architectures [2]. The Supervised Contrastive loss showed its stableness and advantages of avoiding natural corruptions based on the results. The idea of Contrastive learning is to pull together the "anchor," i.e., the target image and the "positives" (matching sample) in embedding space. Concurrently, the anchor would also push apart its "negative" pairs (unmatched sample). Noted that contrastive learning would contrast a single positive to the entire batch, and it does not gather positives of the same class. Thus, Supervised Contrastive learning, instead, takes the set of all images of one category as positives and contrast against all negatives in the batch or the remainder of the batch. The training process was separated into two stages: the pre-training stage and the training stage. For the pre-training stage, the model will try to learn each representation in the datasets, so that the images of the same class are drawn near to each other, and images of the different categories will be pushed apart. In the training stage, the network is then frozen, which means the weights from the pre-training stage stop updating. In the next stage, the classification layer will then get trained using the cross entropy.

In most recent works related to Supervised Contrastive learning, Resnet's appearance tends to be more frequent than other architectures. In this work, we applied Supervised Contrastive loss to ResNet50, ResNet101, ResNet151, VGG16, and DenseNet121. The training process for each architecture has one encoder and one classifier. This is because a projection head is placed after the encoder when applying Supervised Contrastive learning. The encoder contains basic structures of the pre-trained architecture and data augmentation layers. Noted that data augmentation is an essential process for SCL since augmented data takes significant parts in the positives of one class. The classifier took the encoders' output and was only trained to classify labels. When applying Supervised Contrastive loss, the encoder is separately trained with the projection head. Since the pre-training stage and the training stage are separate. The classifier will then get trained individually while making the encoder not trainable.

The work emphasizes the performance of Supervised Contrastive loss on common architectures. The models are designed to evaluate the effects of applying Supervised Contrastive loss instead of their performances. The test accuracy for each structure with cross-entropy ranges from 50% to 60%. The paper mainly focuses on the difference in performance on applying cross entropy and Supervised Contrastive loss.

The rest of the paper is organized as follows. The next part will discuss recent and previous works from researchers related to either FER or Supervised Contrastive Learning. Section 3 introduces our approach to constructing structures for supervised contrastive loss. The detailed framework and experimental data are introduced in section 4, and next section will then analyze and discuss the result. Finally, the last section describes the overall work and direction for future exploration.

2. Related work

2.1. Facial Expressions Recognition

Facial expression is a causal and effective way for human beings to express their inner needs and emotions. Industries tend to shift their efforts to obtain customer emotional feedback to automatic facial expression analysis. Previous studies on facial expressions often divide expressions into six basic classes: disgust, fear, joy, surprise, sadness, and anger, plus neutral, [3] whereas compound emotions such as surprise and anger are not included in the categories. Although contempt has been added to Basic Expressions, it is not currently classified as a separate category in most well-known facial image datasets. [4] Facial expression recognition proceed in three stages: face acquisition, facial data extraction, and facial expression recognition. Face acquisition, also called the face detection process can use the Haar Cascade Classifier method [1]. The extraction process of facial expression analysis of features has two methods: the method based on characteristics of facial shape and the method based on appearance. The image filter can use Gabor wavelets [5].

Over the past two decades, researchers in computer vision have contributed more than 20 datasets for facial expression recognition. Specifically, for the FER-2013 dataset, researchers have made their approaches to achieve the best performance on the dataset. Pramerdorfer & Kampel, in 2016, identified and overcame the issue of common-used architectures in computer vision. Their result shows a state-of-art performance for the dataset, achieving 72.7% test accuracy for VGG and 72.4% for ResNet [6]. Compared to standard datasets in computer vision, published architectures did not accomplish comparatively satisfying test accuracy. Theoretically, the FER-2013 dataset contains many label errors due to how it was collected [6]. Basic architectures like ResNet and VGG generally achieve high and consistent results, around 70% [6,7]. This paper aims to explore possible ways, i.e., supervised contrastive loss, to improve overall performance and extend the exploration of the application of Supervised Contrastive Learning.

2.2. Supervised Contrative Learning

Since first introduced in 2020 by Khosla et al. [8], SCL has been widely discussed by researchers and applied in many works in computer vision. One of the inspirations for this paper comes from a research paper from Jaiswal et al. on the survey of SCL [9]. Jaiswal et al. pointed out four possible issues for Supervised Contrastive learning. They indicated that a lack of theoretical foundation and database biases could profoundly affect the performance. Jaiswal et al. also showed that architecture design and sampling techniques might contribute to the instability of the performance of Supervised Contrastive Learning. Indeed, few recently published works have explored the performance of Supervised Contrastive Loss on other common architectures like VGG. Islam et al., in June 2022, researched on classification of diabetic retinopathy using CSL [10]. The result shows that applying supervised contrastive loss improves the test accuracy of common architectures by 2%. However, it also shows inconsistent accuracy scores on different architectures, meaning Supervised Contrastive loss might also negatively affect their performance—connecting with the issue of theoretical foundation and database biases. It is unclear the stability of overall improvement applying SCL. Our work will evaluate SCL's performance on different architectures by monitoring train & test accuracy and loss.

3. Methodology

Classifying facial expressions has no differences from common image recognition problems. The focus of the neural network for this problem is also recognizing patterns or features on images, specifical images of human faces. This means researchers could apply the most common CNN structures to this problem, for example, ResNet, AlexNet, or VGG. However, the performance of different structures for the dataset could differ significantly due to the dataset's features and environment. To construct the structures with the highest test accuracy in the dataset, we constructed five models with their best validation accuracy. The standard of measuring the performance of different structures was mainly based on their validation accuracy and accuracy for the test set.

3.1. ResNet

Residual Neural Networks (ResNet) [11] have been used in training deeper neural networks. ResNet architecture consists of multiple residual blocks. One primary advantage of the architecture is its simplicity of residual functions. The depth of the function could directly affect the accuracy of training. The problem is to reduce the negative effect of adding more depth to the residual function, which could possibly cause the degradation problem.

The depth is an important factor in building Convolutional Neural Network. Choices of layer number could have considerable effects on the model's performance. Gradient dispersion explosion could potentially disturb deep training networks, causing failure for the model to converge. The degradation problem illustrates that deep networks cannot be easily optimized well. For a given structure, if the input is x, H(x) is then any desired mapping, and F(x) would denote the residual

mapping. The expected learning process to get F(x) is by H(x) – x. This way, the desired mapping H(x) can be obtained from F(x) + x. This is done because residual learning is easier than direct learning from basic features. Taking the residual as 0, the network now merely learns the identity function, i.e., the identity mapping. The degradation problem will no longer affect the performance of the network; Indeed, the residual is unlikely to be 0, enabling the network to learn new features based on x, significantly improving the overall performance. It directly skips the output of a certain layer from the previous layers and make it the input of the subsequent layer. This means the last feature layer will be partially contributed linearly by one of the previous layers.

In this work, we use five ResNet models with different structures: resnet50, resnet101, and the ResNet model built by ourselves. The training process is carried out using pre-stored weights (ImageNet). Among them, ResNet50 and ResNet101 use structures in the Keras library. Finally, we also built a self-constructed ResNet model. The following describes the structure of the network. The input for all networks is re-scale to 48*48*3.

The ResNet50 model in this work is from Keras TensorFlow, using pre-trained weights on ImageNet. The architecture, as specified in its name, contains 50 layers. The image sample will first go through a convolution layer of 64 kernels of stride of 2 with a kernel size of 7 x 7. The following layer is a max-pooling layer of 2 kernels. Followed by the critical part of the ResNet model, the residual block, we used three layers of the residual block with different parameters. The fully connected layer is replaced with a global average pool layer, performing a global average pooling operation on the spatial data and uses SoftMax activation to generate predictions. The model contains about 23m parameters. For the ResNet101, the architecture contains roughly 42m parameters. In order to attempt to minimize the number of parameters without losing performance, we also construct an architecture with fewer layers and smaller kernels for convolution layers. The system consists of one convolutional layer and three residual blocks. Each contains four convolutional layers with batch normalizations. The model only has 1.1 million trainable parameters.

3.2. VGG

The Deep Convolutional SNN Architectures, VGG, aims to increase the depth of common architectures in order to achieve better performance. Once training is completed, the neuron threshold of a particular layer is set equal to the maximum activation of all ReLUs in the corresponding layer. [3] VGG-16 outperform common CNN architecture by its depth and its ability to evaluate the effects of its depth on the performance in some complicate dataset. In Simonyan & Zisserman's work, by using a convolutional filter of size 3, they thoroughly compare the effects of depth on architecture. The result shows that 16-19 weight layers (VGG16) demonstrate a consistent performance. VGG-16 finally got 92.7% test accuracy in ImageNet [12].

The concept of VGG was proposed by Yan & Zisserman in 2013; the architecture was then introduced in the 2014 ImageNet Challenge [18]. There are two basic ideas to distinguish VGG from other models. The first idea is for VGG to combine 3x3 filters with standing for a larger size. The second idea is to use three 3x3 layers instead of one 7x7 layer. The benefit of using three layers is that the additional three non-linear activation layers can improve the discriminative of the decision functions to get a faster convergence speed.

Furthermore, it significantly helps reduce the parameters in the architecture; this potentially reduces the possibility of overfitting during training. The basic architecture of the VGG model considers an RGB image as input, which is fixed size 224x224 images. Input image will transfer to stacks of convolution layers of 3x3, with ReLU activations. Every two layers have 64 filters. After that, three fully connected layers and a flattening layer will take the output and keep updating. The activation function of the output is SoftMax, which is used for categorical classification. Figure 3 shows the detailed step.

Figure 2. Detailed VGG model structure

In this report, VGG16 was trained and evaluated. VGG16 gets an input of a grayscale image whose size is 48x48 and has 5 MaxPooling layers and three Conv2D layers with Relu activation function before each MaxPooling layer two dense layers. There are 14m parameters. Model is compiled with SGD optimizer. The loss function is categorical cross-entropy, and the metric uses accuracy. Early stopping is a method that keeps another set of data as validation data when training the data. The validation accuracy is recorded. When the accuracy of the validation data stop growing, the training process will cease instead of going through all the epochs.

3.3. Encoder

The training process for each architecture contains two stages. The first stage, the encoder, contains the input layer, data augmentation, basic architecture of the model, and output layer. The data augmentation layer is a significant process in SCL. An effective data augmentation method could help reduce mutual information between views, which could help improve the accuracy of the SCL model. The data augmentation consists of a normalization layer and four layers for executing random horizontal flip, rotation, width, and height. The specific architecture would then process the data by taking the output of data augmentation. Noted that the encoder's work is to train the weights and extract features from sample images. Since SCL needs to freeze the weight after training the encoder, it does not contain flattened and dense layers for classification. When proceeding with the SCL, a projection head will take the output of the encoder and project it into 128 units for pre-training. Then the encoder can be individually trained, monitoring the supervised contrastive loss and its validation loss.

For the baseline classification mode, the encoder and classifier are considered as a single model. The cross-entropy loss function for the baseline classification model is the categorical cross entropy loss function provided by Keras, and it was specified in compiling phase. However, for the SCL approach, the encoder and projection head were trained and optimized using supervised contrastive loss function. The supervised contrastive loss function takes labels and featured vectors as input whenever get called. The logic of the code follows the idea introduced by Khosla et al. [2]. The supervised contrastive loss function will first normalize feature vectors using L2 norm. The function will then compute the logit using the (matrix) multiplication of normalized feature vectors and their transpose. The logits are also divided by the temperature parameters to adjust the contrastive power. In the final step, the loss function will call the built in N-pair loss from Keras taking the labels and the logits.

3.4. Classifier

The classifier’s job is to train the classification layer. In this process, the classifier first takes the output from encoders and puts them into a flatten layer, converting resultant 2-D arrays from pooled feature maps into linear vectors. The continuous linear vector goes through three dropout layers using the dropout rate 0.5 and two dense layers with 4096 and 1024 units. The final layer at the end will train the classifier to recognize categories using Softmax. The difference between the traditional method and SCL is the order of training. After the pre-training stage, the encoder is attached with projection layers. All the weights in the architectures for encoders are trained. The next step in the training stage is to make the layers in the encoder untrainable since the weights are expected to be frozen. In this case, the classifier will take the output of the encoder with projection and set all layers

in the encoder to be untrainable. This way, the classifier can be trained using Softmax and Cross Entropy.

4. Experiment

This section evaluates the proposed models on the FER2013 dataset and their corresponding performance with SCL. We first provide an overview of the specific dataset. Then, we display the comparison between the original network structure and SCL. At last, we discuss the possible reasons why original networks perform worse than our variant.

4.1. Dataset

At the International Conference on Machine Learning (ICML) in 2013, Pierre-Luc Carrier and Aaron Courville's famous work, the Facial Expression Recognition dataset (FER2013), has brought attention to many researchers in computer vision. The dataset consists of more than 35 thousand of images of people's facial expressions. There are in total 7 categories of micro expression in the dataset. The categories are marked with labels based on seven classifications from 0 to 6. Table (I) shows the summary of different categories in the dataset.

Table 1. Summary of FER-2013 Based On Categories

Name	Training Set	Testing Set
Angry	3995	958
Disgust	436	111
Fear	4097	1024
Happy	7215	1774
Sad	4830	831
Surprise	3171	1247
Neutral	4965	1233
Total	28709	7178

4.2. Experimental Settings

The number of images in the dataset is far from enough for a training set. Fortunately, emotion classification only contains images of the faces, and data augmentation can be done to these faces. Specifically, training data are randomly rotated to a maximum of 10 degrees, followed by a random horizontal flipping and a width and height shift of 0.2 for data augmentation. Figure (4) shows the example of the original data and the processed data. Our models are implemented with the popular TensorFlow framework. For a fair comparison, all the models use the optimizer SGD [14] with the default hyperparameters in TensorFlow (with a learning rate equal to 0.001). The whole training took place on Google Collaboratory in a free license, whose GPU is NVIDIA Tesla T4 at hand when we do the training. All the models got the training for at least 50 epochs and using 64 as batch size.

Figure 3. Example of data augmentation of a sample

4.3. Training Objective

Cross-entropy loss function, nowadays, has become a widely used loss function in classification work. In this work, all basic architecture without SCL used categorical cross entropy for the multi-class classification work. The basic logic of categorical cross entropy is based on the equation:

$$J(W) = -\frac{1}{N}\sum_{i=1}^{N} y_i log(\hat{y}_i) \quad (1)$$

In equation (1), $J(W)$ is the loss, and W denotes the model parameters, i.e., the weights. y_i is true label, and $\hat{y}_i$ is the predicted label using one-hot encoded scheme, which means their values consist of a binary vector of size 7. The aim of the architecture is to minimize the loss and improve accuracy. Thus, the cross-entropy loss is a significant factor for monitoring our model's performance.

The supervised contrastive loss, in this work, used the metrics from Khosla et al.'s work. The equation (2) is built for handle arbitrary case when the dataset has more than one sample belonging to the class of anchor. The loss function is shown below:

$$L^{sup} = \sum_{i\in I} L_i^{sup} = \sum_{i\in I} -Log\left(\frac{1}{|P(i)|}\sum_{p\in P(i)}\frac{exp\left(\frac{Z_i \cdot Z_p}{\tau}\right)}{\sum_{\alpha\in A(i)} exp\left(\frac{Z_i \cdot Z_\alpha}{\tau}\right)}\right) \quad (2)$$

In equation (2), Z, the featured vectors, is the projection of the output from the encoder. $\tau \in R^+$ denotes the temperature parameter that affects the smoothness of the model. In this work, the temperature is set to be 0.05 for all experiments. In a single minibatch, $i \in I \equiv \{1,2,\dots,2N\}$ is the index of the anchor, where N is the number of sampled pairs. $A(i) \equiv I\backslash\{i\}$ then is the set of indices without i. $P(i)$ denotes the positives in a batch with respect to anchor i. $|P(i)|$ then is the cardinality of the set of the positives.

Compared with other loss functions in supervised learning, this novel loss function has two advantages. First, the loss covers all positives in a minibatch. For all entries in a minibatch, the loss can give a closely aligned representation of the same class. This way, the representation space can give a more robust and clear clustering of samples based on their classes. Second, as the positives increase, the contrastive power of supervised learning increases in these metrics. This is an important property that is proven to be able to improve the overall accuracy of the model.

4.4. Results and Analysis

The first part of the work is to evaluate the general performance of the architectures using cross-entropy. For each model, encoders are attached with classifiers and get trained together. The result is shown in table (2).

Table 2. Training results of different architectures using cross-entropy

Name	Parameters	Accuracy
ResNet50	36.1m	48.41%
ResNet101	42.6m	52.49%
Self-Constructed ResNet	6.4m	49.54%
VGG16	21.0m	56.38%
DenseNet121	15.4m	47.38%

When training ResNet-based network architecture, the training loss and accuracy show consistent results on the dataset. Specifically, when ResNet50 is trained on the default learning rate with Adam optimizer, the cross-entropy loss starts around two and finishes around 0.5, showing a consistent learning pattern. The test accuracy of ResNet50 reaches around 51.18%. Noted that this accuracy is not satisfying enough to be a state-of-art architecture for the dataset. However, the accuracy is compared with the performance of supervised contrastive loss. ResNet101 performs better, and its train accuracy achieved around 63%.

Nevertheless, similar to ResNet50, ResNet101 suffers from overfitting, merely reaching an accuracy of 52.81%. In Figure (5), The validation loss keeps fluctuating and shows an inconsistent

pattern. The performance of self-constructed ResNet also reaches similar test accuracy, around 47.26%. Although its performance is not as good as the pre-trained model, it has a much smaller architecture with fewer total parameters, which outperforms other models in speed. The performance of VGG16 and DenseNet121 are similar to ResNet even though the validation accuracy (using a validation split of 0.2) fluctuates around 1.4. The test accuracy for our VGG16 achieved around 50%, which is high enough for our evaluations on supervised contrastive loss on the dataset. The learning process for DenseNet has the best result. The validation loss keeps decreasing for the first 50 epochs. However, the loss suddenly increased and fluctuated. One possible answer is that the training batch encountered extreme or abnormal data, and the weights in the whole model were changed dramatically regarding the large gradients. Due to hardware limitations, this might contribute to the considerable fluctuation of the model training. This hypothesis can be partly verified in the training process of VGG16 and DenseNet121, where some of its weights jumped up to infinity and messed up the whole prediction.

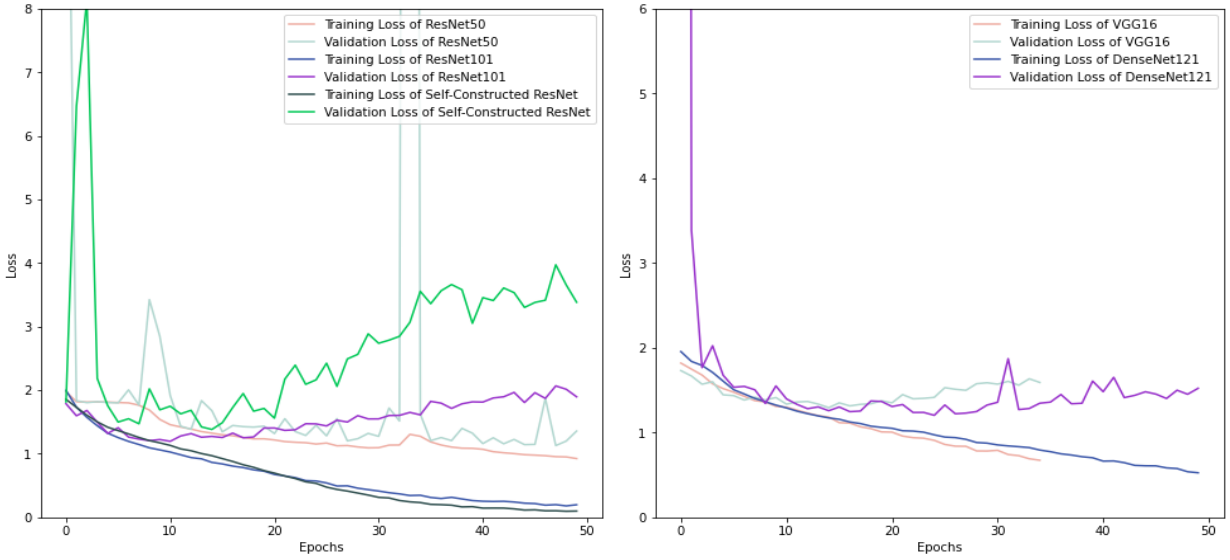

Figure 4. Training and Validation Loss of ResNet, VGG, and DenseNet with Cross-entropy loss function (VGG16 trained for 33 epochs because of early-stopping)

For the next part, we break up the encoder and the classifier, adding a projection layer after receiving the output of the encoder. The encoder for each corresponding architecture with the projection layer was then trained individually in the pre-train stage. During the pre-training stage, the supervised contrastive loss traces the learning progress. For the ResNet models, the training effects are very consistent. The supervised contrastive loss decreased consistently from around 5 to 3, showing that the encoder is learning as expected. Figure 6 record the supervised contrastive loss when training the encoder of three ResNet Model.

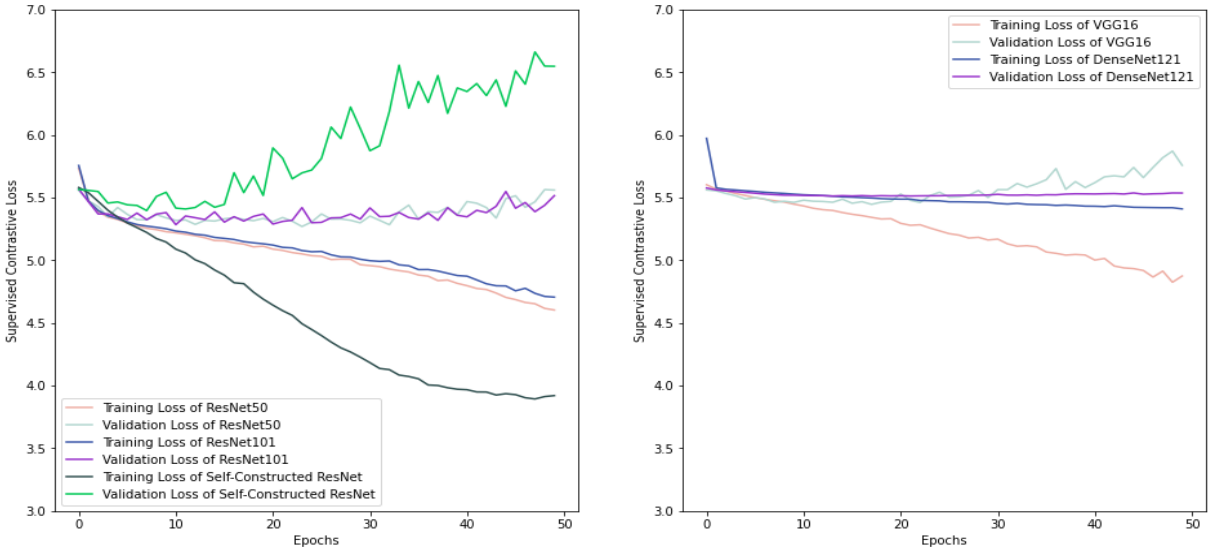

Figure 5. Training and Validation Loss of ResNet, VGG, and DenseNet with SCL

In the training stage, the weights in the encoder are frozen, i.e., all layers in the encoder become untrainable. The classifier then gets trained using SoftMax and cross-entropy. The accuracy is shown in Table (3). For each architecture, the classifier of each architecture runs for 50 epochs. The second column of Table (3) shows the training accuracy of training the classifier, i.e., the accuracy on training set. The values drawn from the training accuracy of last epoch. The third column shows the evaluation of each architecture's performance on the test set in the dataset. The values of the test accuracy are drawn from the evaluation function of each model. The final test accuracy for ResNet50, ResNet101, and self-constructed Resnet with SCL achieved 54.93%, 56.12%, and 52.05%, showing a consistent improvement in performance.

However, the supervised contrastive loss for VGG16's and DenseNet's encoder changed much slower. The loss reduced to 4.88 from 5.62, showing a slow and weak learning pattern in Figure (6). The change in loss, as shown in the Figure, seems insignificant. The resulting test accuracy is also not promising, reaching 47.38% ,40.48%, which are much smaller than the performance using cross-entropy.

Table 3. Performance of different architectures training in two-stages with supervised contrastive loss

Name	Training Accuracy	Test Accuracy
ResNet50	78.32%	54.81%
ResNet101	75.79%	56.38%
Self-Constructed ResNet	92.83%	52.49%
VGG16	71.95%	47.38%
DenseNet121	43.77%	40.38%

5. Discussion

In the normal process of CNN model training, all architectures are trained using the cross-entropy loss function for classification work in only one stage. In the SCL attempt, the features in the sample space are trained using the encoder attaching to the projection layer using supervised contrastive loss function and Adam optimizer, which is also defined as the pre-training stage in this work. After the pre-training stage, the classifier would get trained for classifying the labels in the second stage, or training stage.

The general performance of CNN models in this work is moderate. In table (2), the test accuracy for our models using the cross-entropy loss function range from 47% to 56%. Even though they are not performing as well as the state-of-the-art architectures in the FER-2013 dataset, the study focuses

on comparing the one-stage model with cross-entropy and the two-stage model with supervised contrastive loss.

The test accuracy of ResNet models using supervised contrastive loss function, in figure 5, illustrates a consistent improvement in its performance. The loss function is a significant factor in tracking the learning process [22]. According to the graph of supervised contrastive loss verse epochs number, the loss keeps moving downward slowly, showing that the learning is proceeding. However, the validation loss fluctuates around 5 to 6 and does not decrease as expected. The problem appears in all ResNet models. Based on that, we proposed two possible reasons:

Overfitting. Based on the loss plot, the model demonstrates a lower loss value in the training set and a relatively unstable error pattern. Figure (6) shows a typical example of overfitting, which resembles figure (7). This indicated that our model's complexity cause memorization of irrelevant noise instead of the features themselves [20]. Indeed, we made a few attempts to minimize the effects of overfitting, such as data augmentation, regularization, and batch normalization, which slightly reduced the validation loss of ResNet50 and self-constructed ResNet.

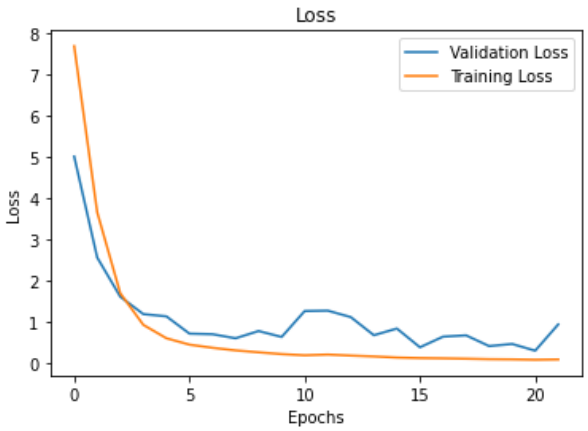

Figure 6. Example of overfitting when training CNN models.

The problem of the dataset. According to the research conducted by Goodfellow et al., FER-2013 contains many label errors, potentially making it hard for the training process [6]. In this case, the validation loss could fluctuate whenever a falsely labeled sample comes in. In the facial expression recognition competition, Bergstra & Cox's unique approach using a "null" model for the dataset yielded exciting results. In their work, all layers expect the final classifier layer to be untrainable, i.e., no learning, which surprisingly achieved an accuracy of 60% [2]. This proves that the dataset's characteristics may contribute to many strange cases. In this work, all these architectures are designed to classify the ImageNet [21] dataset (using weights from ImageNet), which is much more significant in the number of categories. In these cases, images in different categories differ dramatically. Whereas in the dataset of FER-2013, the facial image of individuals does not demonstrate contrast differences.

Table 4. Summary of the accuracies using different loss function

Name	Test Accuracy in One-Stage	Test Accuracy Using SCL	SCL's Effects on Accuracy
ResNet50	48.41%	54.81%	6.40%
ResNet101	52.49%	56.38%	3.89%
Self-Constructed ResNet	49.54%	52.49%	2.95%
VGG16	56.38%	47.38%	-9.00%
DenseNet121	47.38%	40.48%	-7.00%

The results were all collected in Table (4). The second column of the table shows the test accuracy using baseline classification models on the test set, which is drawn from Table (1). The third column of the table is the test accuracy on the test set using supervised contrastive loss, which are from Table

(3). The final column summarizes the effects of SCL on the general performance on the dataset. The values were evaluated comparing the test accuracy using baseline classification models and using supervised contrastive loss.

The overfitting problem and the dataset may cause some issues with the performance of the ResNet models in this work. Nonetheless, SCL provided a substantial improvement in the classification work. Table (4) showed that the ResNet model grows around 2-6% in test accuracy. The improvement shows consistency and stability, as the models were trained several times. The result suggests that supervised contrastive loss helps effectively extract features in the representation space in the FER-2013 dataset.

The performance of VGG16 and DenseNet121 using supervised contrastive loss appears to be worse than the model with cross-entropy, as shown in Table (4). The corresponding test accuracies with SCL achieved 47.38% and 40.48%. Based on the loss plot in the pre-training stage, the supervised contrastive loss did not show a significant change, suggesting a poor learning process. The problem for the VGG16 model with SCL is quite similar to ResNet. As shown in figure (6), the loss decreased at a plodding pace, and the instability of validation loss demonstrated a sign of overfitting. Besides the problem of overfitting and dataset, the size of the model and optimizer also contributes significantly to the performance. The VGG model used an Adam optimizer taking a learning rate of 0.00025 (achieved the best accuracy), and merely the last seven layers were set to be trainable. The model achieved a test accuracy of around 47.38% by adjusting these parameters. A better learning rate and layers number choice can potentially improve the performance using SCL.

The result showed that different architectural designs and sampling techniques could potentially affect the performance in contrastive learning. The statement is arguably true in this work. Failure to apply SCL affirms the supervised contrastive loss's dependency on specific design and the need for more theoretical work on supervised contrastive loss applying to different architectures.

6. Conclusion

This paper has extensively evaluated the performance of different architectures in the FER-2013 dataset and compared cross-entropy loss and supervised contrastive loss function. In this work, we clearly explain the application of SCL in facial expression recognition problems. We have constructed five models to examine the performance of supervised contrastive loss. We have shown that SCL can substantially increase the performance of the ResNet model by increasing 4% the test accuracy. The result based on VGG16 and DenseNet121 reveals the potential problem of SCL on overfitting and dataset. Finally, this work concludes by discussing some open problems of SCL that could be improved. New theoretical work is needed for better applications of supervised contrastive learning.

References

[1] L. Zahara, P. Musa, E. Prasetyo Wibowo, I. Karim, and S. Bahri Musa, "The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based raspberry pi," in Proc. 5th Int. Conf. Informat. Comput. (ICIC), Nov. 2020, pp. 1–9.

[2] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In NeurIPS, 2020.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[4] A. Khanzada, C. Bai, and F. T. Celepcikay, "Facial expression recognition with deep learning," arXiv preprint arXiv:2004.11823, 2020.

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. ICLR, 2014

[7] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.- H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In Neural information processing, pages 117–124. Springer, 2013.

[8] M. R. Islam, L. F. Abdulrazak, M. Nahiduzzaman, M. O. F. Goni, M. S. Anower, M. Ahsan, J. Haider, M. Kowalski. Applying supervised contrastive learning for the detection of diabetic retinopathy and its severity levels from fundus images. Computers in Biology and Medicine. 2022 May 10:105602.

[9] J. Bergstra, D.D. Cox, Hyperparameter optimization and boosting for classifying facial expressions: how good can a "null" model be?, arXiv:1306.3476 (2013).

[10] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in International Joint Conference on Neural Networks, 2015, in press.

[11] FER-2013 Dataset. Accessed: May 1, 2020. [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facialexpression-recognition-challenge/data

[12] C. Pramerdorfer and M. Kampel. Facial expression recognition using convolutional neural networks: State of the art. CoRR, abs/1612.02903, 2016.

[13] M. I. Georgescu, R. T. Ionescu, and M. Popescu, ''Local learning with deep and handcrafted features for facial expression recognition,'' IEEE Access, vol. 7, pp. 64827–64836, 2018.

[14] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," Technologies, vol. 9, no. 1, p. 2, 2021.

[15] T. Kanade, J. Cohn, and Y.-L. Tian. Comprehensive database for facial expression analysis. In Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[16] Y. Khaireddin and Z. Chen, Facial emotion recognition: State of the art performance on FER2013, 2021, arXiv:2105.03588.

[17] S. Li and W. Deng, "Deep facial expression recognition: A survey," CoRR, vol. abs/1804.08348, Jun. 2018.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252, 2015

[19] Y. Tian, T. Kanade, and J. F. Cohn, "Facial expression recognition," in Handbook of Face Recognition. London, U.K.: Springer, 2011, pp. 487–519.

[20] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, ''Convolutional neural networks: An overview and application in radiology,'' Insights into Imag., vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018- 0639-9.

[21] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi. A theoretical analysis of contrastive unsupervised representation learning. ICML, 2019

[22] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," IEEE Trans. Comput. Imag., vol. 3, no. 1, pp. 47–57, Mar. 2017.