Facial Expression Recognition using Deep Learning Models

Gordon Lu, Revanth Reddy, Karthik Vadlamudi

Code Base: <https://github.com/glu99331/DL-Emotion-Detection>

Dataset: <https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data>

Introduction

Human emotion is defined by a large number of factors including mood, environment, and personal factors. These emotions may manifest in several physiological observable features of the individual, including factors of the person’s brain activity, heart rate, and facial expressions [1].  Predicting or estimating emotions based on these observable parameters can have great value in fields such as mental health monitoring, multimedia recommendation, and human-computer interaction. Here, we present methods for predicting emotion based on images taken as people experience various emotions. Facial recordings can capture emotional changes that manifest in facial expressions of individuals [2]. Specific emotions can be linked to changes in facial expressions corresponding to the muscles located in the face. These changes can be recognized using neural network models and linked to unique emotions based on facial muscle movements. In this project, we analyze several neural network architectures capable of recognizing input images of facial expressions and predicting the corresponding emotions. The neural network predicts 7 emotions (anger, disgust, fear, happiness, sadness, surprise, and neutral) based on 48 x 48 greyscale images of facial expressions. In this paper, we look at the efficacy of several different deep neural networks models in predicting emotions based off of these facial expressions.

Literature Survey

Human emotions are complex in that they are driven by many variables and expressed in a person-specific manner. This quality makes predicting emotions difficult due to their ill-defined and volatile nature that is unique to each person. Emotions are typically defined as categories (e.g. happy, sad, excited) but they can also be defined through quantitative measures of the emotion’s intensity and pleasantness [3]. These emotions can manifest themselves in a number of different ways either physiologically or visually. Physiologically, emotions can cause specific, observable reactions in several signals including the person’s heart rate, brain activity, temperature, or even in skin conductance [4]. Though changes in physiology can be observed, linking these changes to specific emotions has proven to be a difficult task. Similar to the plethora of physiological signals that can be indicative of emotion, facial expressions and movements of facial muscles are also highly correlated with specific emotional changes [2]. More specifically, certain movements and positions of key facial features can be indicative of an underlying emotion experienced by a person. For example, if a person is happy, they may move their lips and eyes in a specific direction to indicate their happiness. However, the specific changes are highly subjective, making it a difficult task to complete in the field of computer vision.

Convolutional neural networks have greatly improved the capabilities of emotion prediction based on images of people experiencing emotions. These neural networks are optimal for image recognition tasks and capable of learning features in images that correspond to specific labels given to them. A number of models have previously been used for facial expression recognition tasks successfully, including ResNet and VGG16 [5]. The accuracies for these tasks are typically much lower than similar tasks such as image recognition (i.e. MNIST) because of the similarity between facial images and high subject-subject variability. Two people may express the same emotion in unique ways that has to be captured by these models. Facial emotion recognition can benefit a number of fields ranging from healthcare to education. For example, facial images can be used to assess patients' emotions or mood and alert medical professionals in the case of anomalies. This can be important for the monitoring of those with mental illness such as post-traumatic stress disorder or major depressive disorder. In a workplace or education setting, emotion recognition can be used to optimize productivity by warning people about various factors such as stress. Optimal facial emotion recognition can have a profound effect on the fields of medicine and societal wellbeing [6].

In this paper, we analyze several different models (ResNet, VGG, and MobileNet) in a facial expression recognition task. Our dataset comes from a public repository of a facial expression recognition challenge [7]. The dataset consists of 35,887 grayscale images consisting of 48 x 48 pixels. Each image contains a facial expression from one of seven categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. For each image the face of the person is centered, and the true label is provided. The best performing models in the original challenge ranged between 60-70% accuracy, which is significantly lower than other image recognition tasks. As a basis of our model, we adopt code developed for a simpler facial expression task consisting of 5 emotions and apply it to the dataset. The goal of this project is to gain a deeper understanding of facial expressions and compare methods in predicting emotions associated A close-up of a person's face

Description automatically generatedwith them.

Technical Details and Model

In selecting our models, we built on the existing Resnet-18 model provided in the repository. Since the dataset consisted of grayscale images, while the pretrained models provided by keras were trained on RGB images, the author created a custom Resnet-18 model to comply with the dataset. Before diving into each model we selected, for every model, we should note that the optimizer, loss function and number of epochs were identical to ensure consistency among runs. In particular, we ran Adam for the optimizer, categorical cross entropy loss for the loss function and ran for 100 epochs. We also added early stopping to avoid unnecessary training in the model if the validation loss does not improve over a certain number of iterations.

Our first model we created followed the author’s approach. We created our own custom Resnet-50 to avoid the need to transform the images to RGB. With this custom Resnet-50, we specified L2 regularization in the convolutional layers and as well as a dropout layer and a learning rate scheduler that would decrease the learning rate if it plateaued after a certain number of epochs. The idea behind opting to use plateaued loss was to allow for the model to be dynamic in learning better parameters, in case it was stuck to avoid early stopping from triggering. In addition, the dropout layer and regularization were implemented to prevent overfitting.

The next set of models we introduce are from the existing pretrained models provided by keras. For each of these models, they were trained on RGB images, thus we needed to extend the dimensions of the images to comply with the model requirements.  The first of these models was the Resnet-50. Although we had already created a custom Resnet-50, the pretrained Resnet-50 was trained on RGB images and required the aforementioned transformation on the dataset. The pretrained Resnet-50 model was trained with the same common set of parameters, as well as with a plateaued learning rate scheduler.

We also tried two other pretrained models that have previously been used for image recognition tasks: VGG16 and MobileNetV2. VGG16 contains 16 layers consisting of small convolutional layers, and, unlike ResNet, does not use pooling and skipped connections for training. Alternatively, the VGG19 model has more layers and trainable parameters and can also be tested. The MobileNetV2 is a neural network architecture designed for mobile and embedded devices. As a result, it contains less trainable parameters and is significantly smaller than the other models being tested. Though the model may not achieve as high accuracies as the deeper and denser neural networks, it is useful in implementing these networks in a real-world situation with size and speed limitations. We use similar parameters as previous models and hope to gauge the effectiveness of each of these models in the facial expression recognition task.

Results

Baseline ResNet Training

A graph of a training and validation loss

Description automatically generatedA graph of a training and validation accuracy

Description automatically generated

We first established a baseline by running the custom Resnet18 model provided in the repository, which yielded around 63% accuracy. From the above plots, there is a staggering difference in the loss curves and accuracy plots. The explanation could be that there are an excessive amount of layers and parameters or perhaps it is encountering the vanishing gradients problem. The plan was to use prebuilt keras models to train on the dataset and compare the results to the baseline. We attempted to use the pretrained VGG16 and Resnet-50 models.

With the pretrained ResNet-50 model, we attained around 60% accuracy on the data using a plateau learning rate scheduler. As seen by the below loss curves, the curves are relatively close, indicating a lack of overfitting during training. However, the relatively low loss compared to the baseline custom Resnet-18 can be attributed to the lack of flexibility to add regularization parameters to the convolutional layers.

Custom ResNet-50 Training

A graph of training and validation

Description automatically generatedA graph of a training and validation accuracy

Description automatically generated

Following this approach, we opted to replicate the approach taken in the repository and define a custom ResNet-50. With this custom Resnet-50, we specified regularization in the convolutional layers and as well as a dropout layer and a similar learning rate scheduler, which yielded slightly better results of 65%. It is important to note that the validation and training loss curves differ quite a bit, indicating a degree of overfitting towards the training set, which is likely due to the setup of the custom ResNet-50 being more complex than the pretrained ResNet-50.

A graph of a training and validation accuracy

Description automatically generatedPre-Trained ResNet-50 Training

A graph of a training and validation loss

Description automatically generated

VGG16 Training

A graph of a training and validation accuracy

Description automatically generatedA graph of training and validation

Description automatically generated

The VGG16 model outperformed all other models in the emotion recognition task with an accuracy of 67.5%. This model contains 14,847,815 trainable parameters and allocates ~56 mB of memory, which is significantly higher than other models. Compared to other models, the training and validation appear to steadily increase as the model learns the dataset. Based on these results, it seems that deeper and denser networks are ideal for facial expression recognition tasks. The image for 2 different emotions may only have a nominal difference, and the model we use has to be capable of capturing these dynamics. As a result, deeper neural networks like VGG16 appear to work well for the dataset.

MobileNetV2 Training

A graph of blue lines

Description automatically generatedA graph of training and validation

Description automatically generated

The results for MobileNetV2 prediction of facial expressions are significantly worse (~53%) than the other models. This model also has fewer trainable parameters and allocates less space than the other models. However, the behavior of accuracy and loss seem to indicate that the models are overfitting to training data and failing to capture the behavior of the facial expressions. The accuracy of the model plateaus at around 55% and fails to exceed this limit despite training for more epochs. This could mean that a larger number of trainable parameters and neural network depth is needed to properly analyze facial expressions. The denser and deeper networks perform significantly better and the MobileNet should be reserved for tasks it excels at.

Conclusion

In this project, we attempted to use various neural network architectures to classify facial expressions into categories of emotions. Originally, we based our architecture on previous work based on emotion recognition with fewer emotions (5) and expanded the model to include 7 emotions and train with a larger dataset than originally intended. We hoped to try several models on the task of facial expression recognition and understand more about the nature of facial expressions and the benefits of each type of model. We were able to successfully train neural networks to recognize facial expressions. The strongest performing model ended up being the VGG16 model adopted for our uses. The MobileNetV2 model has the poorest performance of all models and appeared to overfit to the training data, based on training results. In general, the performance of our models was similar to the models that achieved the highest accuracies in the corresponding Kaggle database. However, we were unable to uncover more information on how facial expressions relate to emotions. Transformation methods highlighting certain features proved to be detrimental to our model performance, leading us to believe the entire image is needed for optimal classification. This is because emotions can impact the position of several key facial features. In future studies, we hope to further understand the behavior of these facial features to help augment data and achieve higher accuracies.

References

[1] Bradley, M. M., & Lang, P. J. (2000). Measuring emotion: Behavior, feeling, and physiology.

[2] Ko, B. C. (2018). A brief review of facial emotion recognition based on visual information. sensors, 18(2), 401.

[3] Russell, J. A. (1980). A circumplex model of affect. Journal of personality and social psychology, 39(6), 1161.

[4] Shu, L., Xie, J., Yang, M., Li, Z., Li, Z., Liao, D., ... & Yang, X. (2018). A review of emotion recognition using physiological signals. Sensors, 18(7), 2074.

[5] Mellouk, W., & Handouzi, W. (2020). Facial emotion recognition using deep learning: review and insights. Procedia Computer Science, 175, 689-694.

[6] Boughanem, H., Ghazouani, H., & Barhoumi, W. (2023). Facial Emotion Recognition in-the-Wild Using Deep Neural Networks: A Comprehensive Review. SN Computer Science, 5(1), 96.

[7] Dumitru, Ian Goodfellow, Will Cukierski, Yoshua Bengio. (2013). Challenges in Representation Learning: Facial Expression Recognition Challenge. Kaggle. https://kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge