Emotion Recognition Using Convolutional Neural Networks

Supervisor: Antonio Prgomet

Student: Jenan Sarabi

Deep Learning -Data Science 22

Jenansarabi/Deep-Learning-CNN (github.com)

Table of Contents

1 Introduction

1.1 Background

Facial expressions play a vital role in human communication, understanding the intentions of others typically, people determine the emotional state of others, such as joy, sadness, and anger, by observing their facial expressions.

Emotion recognition from facial expressions plays a crucial role in various fields including healthcare, entertainment, and security. The goal of this project is to develop an accurate and efficient Facial Expression Recognition model.

Traditional methods relied heavily on manual feature extraction, but with the advent of deep learning, particularly convolutional neural networks (CNNs), automated emotion recognition has seen significant advancements.

1.2 Purpose and Research Question

The purpose of this project is to develop a robust CNN-based model for accurately recognizing emotions from facial images.

The primary research question guiding:

Develop convolutional neural networks effectively recognize and classify emotions from facial expressions.

In addition to developing the CNN- model for emotion recognition, this project also aims to create a Streamlit application connected to the model.

2.1 Exploratory Data Analysis (EDA)

The Face Expression Recognition dataset available on Kaggle comprises grayscale images of human faces, categorized into seven different facial expressions: angry, disgust, fear, happy, sad, surprise, and neutral.

The dataset is divided into two subsets:

Training Set: This subset consists of 28821 grayscale images used for training the emotion recognition model.

Validation Set: The test set contains of 7066 grayscale images, which are held out for evaluating the trained model's performance.

2.2Distribution of Emotions:

In the distribution plot of emotions, we observe that the category 'Happy' has the highest number of images, indicating it is the most represented emotion in the dataset. Conversely, the category 'Disgust' exhibits the lowest number of images, suggesting it is the least represented emotion. Following 'Disgust', the category 'Surprise' also demonstrates a relatively lower number of images compared to other emotions.

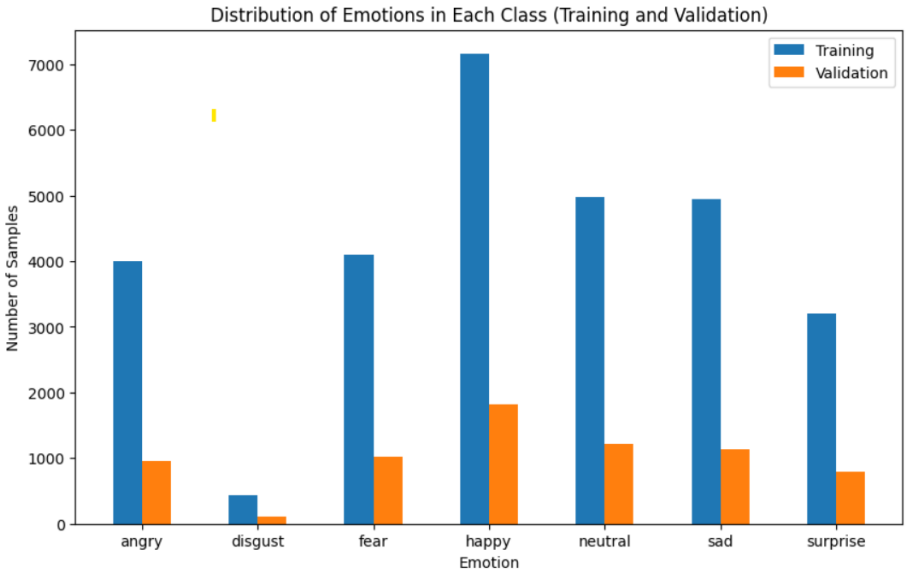

Figure 1 Explain distribution of emotions in each class.

3.Method and Models

3.1.1Convolutional Neural Networks (CNNs)

The model consists of multiple convolutional layers with batch normalization, ReLU activation, max pooling, and dropout regularization to extract hierarchical features from input images and prevent overfitting.

Fully connected layers are used for classification, followed by batch normalization, ReLU activation, and dropout regularization.

The model is compiled with appropriate optimization parameters and loss function for efficient training and evaluation.

This CNN model architecture is designed to effectively classify emotions from grayscale images and incorporates various techniques to improve learning and generalization performance.

3.1.2 Libraises used in Project

Pandas, NumPy, and Keras were utilized for data manipulation, preprocessing, and model development.

3.2 Preprocessing for model input encompasses several essential steps.

Image Loading and Resizing:

Load images from the dataset and resize them to a standard dimension, typically (56, 56) pixels, ensuring uniformity in input size for the CNN.

Pixel Value Normalization:

Normalize pixel values to a range between 0 and 1, facilitating stable model training and convergence.

Data Augmentation:

Data augmentation is done to avoid bias towards one expression . Data augmentation increases the amount of data using techniques like rotation, shifting, and flipping to increase dataset diversity and prevent overfitting.

3.3 Streamlit application

The Streamlit app presented in this project serves as a powerful tool for real-time facial emotion detection. Leveraging a pre-trained convolutional neural network (CNN) model and OpenCV for face detection, the app offers seamless integration and intuitive functionality.

4.Result

The original model, designed for 48x48 pixel images, faced issues with overfitting and bias. It tended to detect only one dominant emotion in each run, failing to capture the complexity of diverse emotional expressions. However, by incorporating augmentation and regularization techniques, significant improvements were achieved.

Augmentation diversified the training dataset by introducing variations, enabling the model to learn more generalized features. Meanwhile, regularization methods like dropout and weight decay helped prevent overfitting, ensuring the model's learned representations were more robust.

As a result, the refined model exhibited enhanced performance, demonstrating improved resilience to overfitting and bias. These findings underscore the critical role of preprocessing strategies and regularization techniques in deep learning tasks, highlighting their importance in refining model performance and addressing common challenges encountered during model training. Overall, these adjustments led to the development of a more accurate and reliable deep learning model for emotion detection.

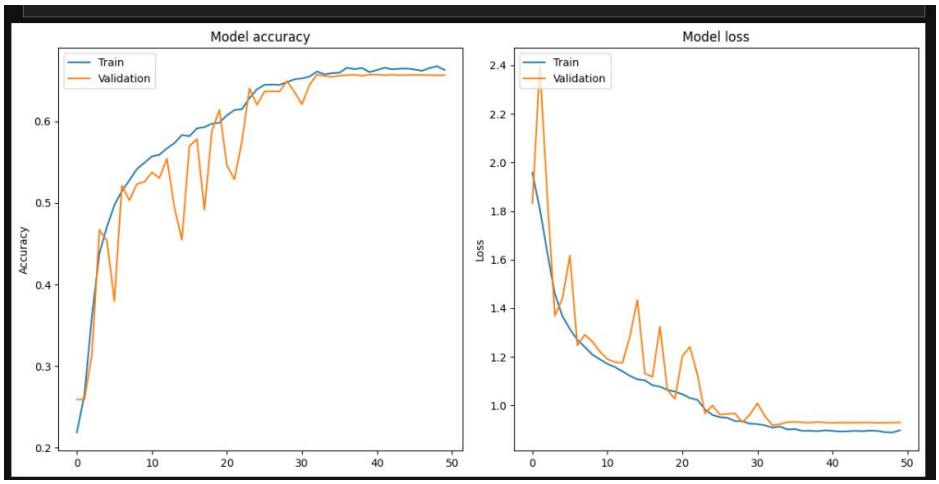

Figure 2- Plot for Best model accuracy.

5.Conclusion

In summary, this project aimed to enhance facial expression recognition by employing Convolutional Neural Networks (CNNs). Initially, the model faced challenges such as overfitting and bias, which were addressed by implementing data augmentation and regularization techniques. Augmentation diversified the dataset by introducing variations, while regularization methods like dropout prevented overfitting. Consequently, the refined model exhibited improved resilience and accuracy in detecting various facial emotions. These adjustments underscore the significance of preprocessing strategies and regularization techniques in enhancing deep learning model performance. Furthermore, the project developed a user-friendly Streamlit application to facilitate real-time emotion detection, detect faces in each frame of a video and give real-time predictions for each detected face.

Further analysis and adjustments are required to enhance the model's performance. This may involve exploring alternative model architectures and optimizing memory usage to address limitations. Additionally, experimenting with different preprocessing techniques, augmenting the dataset, and fine-tuning hyperparameters could lead to improved results. By iteratively refining the model and training strategies, we aim to develop a more accurate and efficient facial emotion recognition system.

Challenges Faced and Overcoming Them: The limited memory on my PC posed a significant obstacle throughout the project. During both training and loading model ,This challenge persisted even when attempting to implement and train the model on platforms like Kaggle or Google Colab with GPU support. Despite these constraints, I successfully managed this challenge.

Leveraging Streamlit alongside OpenCV notably simplified the project, highlighting the importance of adapting to overcome loading model technical challenges for successful project completion.

Overall, the project provides valuable insights into the application of CNNs for facial emotion recognition and emphasizes the importance of overcoming technical challenges to achieve project objectives.

Questions and answers

1. **How are AI, Machine Learning and Deep Learning related?**

   Artificial intelligence is a technique of turning a computer-based robot to work and act like humans.

AI basically helps us to enable the machine to think and to make own decision.

While machine learning is a branch of AI, it's statical tools to explore and understand the data which focuses on using data and algorithms to imitate how humans learn.

Deep learning is a type of machine learning and artificial intelligence that resembles the way humans gain certain types of knowledge, DL includes statistics and predictive modeling which make models quicker and simpler for major amounts of data.

In summary, machine learning is a broader concept that encompasses various techniques and approaches to enable machines to learn from data. Deep learning is a specialized subset of machine learning that focuses on neural networks with multiple layers, allowing for more complex and abstract representations of data.

2. How are Tensorflow and Keras related?

TensorFlow and Keras are closely related within the realm of data science and machine learning. Keras operates as a user-friendly abstraction layer built upon TensorFlow, simplifying the machine learning process, particularly for tasks like image recognition and natural language processing.

TensorFlow, on the other hand, provides more granular control and detail. Both are utilized for constructing and training models, with Keras offering a high-level interface and TensorFlow offering more extensive capabilities.

3-What is a parameter? What is a hyperparameter?

Parameters are the internal variables learned by the neural network during training, parameters such as weights and bias.

While Hyper parameters are external settings that influence how the learning algorithm operates , manually specified when training neural networks need to specify neurons in the hidden layers such as learning rate, epochs, number of layers, neurons in each layer are called hyperparameters.

4-When you are going to make model selection and model evaluation, you can use a training, validation and test data?

Explain how the different parts can be used.

In model selection and evaluation, training data teaches the model patterns. Validation data fine-tunes hyperparameters, avoiding overfitting. Test data assesses real-world performance, ensuring the model's reliability in unseen scenarios.

5. Explain what the below code does:

```
n_cols = X_train.shape[1]

nn_model = Sequential()
nn_model.add(Dense(100, activation = 'relu', input_shape = (n_cols, )))
nn_model.add(Dropout(rate=0.2))
nn_model.add(Dense(50, activation = 'relu'))
nn_model.add(Dense(1, activation = 'sigmoid'))

nn_model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

early_stopping_monitor = EarlyStopping(patience = 5)
nn_model.fit(X_train, y_train, validation_split = 0.2, epochs = 100, callbacks = [early_stopping_monitor])
```

The provided code defines a Sequential model binary classification by using sigmoid activation.

The model consists of an input layer, a dropout layer, a hidden layer, and an output layer.

Dense layers contain 100 ,50,1 neuron , the dropout layer value is 0.2 to prevent overfitting.

It uses ReLU activation for the hidden layers and sigmoid activation for the output layer, typical choices for binary classification.

The model is trained using the provided training data (x_train, y_train) for 100 epochs.

Early stopping with a patience of 5 epochs is employed to prevent overfitting and monitor the validation loss.

The model is overfitting and that’s means that have high variance that’s from flexibility in model that occurs to leading to overfitting.

6. What is the purpose of regularizing a model?

Regularizing a model helps prevent overfitting by constraining its flexibility, often by reducing weights in neural networks. Techniques like L1/L2 regularization, dropout, and early stopping simplify the model or augment training data. Dropout randomly deactivates neurons during training, promoting robustness and generalization. It effectively reduces the risk of memorizing noise in the training data, leading to improved performance on unseen data.

We have two ways to regularizing

1- Constraining a model to simplify it (Fewer degrees of freedom) like L1/L2, dropout early stopping
2- Data augmentation.

7. "Dropout" is a regularization technique, what is it?

Dropout is a regularization technique that randomly deactivates a fraction of neurons during training, with a specified dropout rate p. For example, if $p = 0.25$, 25% of neurons are inactive in each training step. During testing, all neurons are active. To compensate for dropout during testing, a keep probability of $(1-0.25)=0.75$ is used to scale the inputs.

8. "Early stopping" is a regularization technique, what is it?

Early stopping is a regularization technique used during the training of machine learning models,

Stop the learning once the validation error is minimum to stop overfitting process when the validation loss is going to start getting higher during training. This prevents the model from overfitting to the training data and helps find an optimal balance between bias and variance.

We want to train our model and make sure that converges to a solution and then have a separate process on top of it to deal with overfitting rather than stopping training early because combining these two processes namely training and mitigating overfitting will cause further confusion down the line.

9. Your colleague asks you what type of neural network is popular for image analysis, what answers you?

Convolutional Neural Networks (CNNs) are popular for image analysis tasks.

Batch normalization put it in before first layer and then effectively your impulse will be normalized

10. Explain at a general level how a "Convolutional Neural Network" works.

A Convolutional Neural Network (CNN) is a type of deep learning tool used for recognizing objects in images. It's handy for tasks like identifying things in pictures, which is useful in areas like self-driving cars and security cameras.

11. Your friend has an album with 100 different pictures that include e.g. tennis balls and zebras. How

he/she would have been able to classify those images even though he/she has no more data to

train a model on?

To train a model to classify images based on just 100 images can be challenging, the images number is very low. With CNN model could apply data processing to standardize, cropping images, and resizing images also data augmentation to increase the size of the dataset.

Additionally careful tuning of hyperparameter is important for improving model performance.

12. Your colleague asks you what type of neural network is suitable for conducting a

Sentiment analysis, what is your answer?

Natural Language Processing (NLP) is the suitable model for sentiment analysis.

13. Explain at a general level how a "Recurrent Neural Network" (RNN) works.

In RNN model the output depends on the input and previous output ,RNN models have a form of memory because an output depends on a previous output which turn previous output.

14. What does the below code do?

```
model.save('model_file.h5')
```

```
my_model = load_model('model_file.h5')
```

Save and Load model.h5

Model.save is to save model during and after training and to save the entire model to a HDF5 file. The '.h5' extension indicates that the model should be saved to HDF5.

My_model = Load_model('modlel_file.h5') is recreate the exact same model from that file