Project Report: Expression Classification from Facial Images

Author: HAMID HUSSAIN Date: 7/30/2023

Github: https://github.com/hamidhussein

Introduction

The goal of this project is to perform expression classification from facial images. The project involves preprocessing the images, building a deep learning model for classification, and evaluating its performance. The project is implemented using Python and various libraries including Pandas, NumPy, PIL, Scikit-Learn, and Keras.

Data Preprocessing

The preprocessing of the images involves several steps, including loading the label information from a CSV file, filtering the images based on confidence scores, cropping and resizing the images, converting them to grayscale, and saving them to a destination directory. The preprocessing steps are as follows:

1. Load label information from a CSV file containing image names, face coordinates, confidence scores, and expression labels.
2. Set a threshold for confidence scores and filter out images with confidence scores below the threshold.
3. For each image, crop the face region using the provided coordinates, resize it to a fixed size (224x224), convert it to grayscale, and normalize the pixel values to [0, 1].
4. Save the preprocessed images with their original names in a destination directory.
5. Count the number of images for each expression label.

Data Splitting and Label Encoding

After preprocessing the images, the dataset is split into training, validation, and test sets. The expressions are encoded using the LabelEncoder to convert string labels into numerical values. The data splitting steps are as follows:

1. Load the preprocessed image filenames from the destination directory.
2. Filter the label DataFrame to include only the image filenames present in the preprocessed images directory.
3. Use LabelEncoder to convert the expression labels into numerical values.
4. Split the data into training, validation, and test sets using the train_test_split function from Scikit-Learn.

Model Architecture

The deep learning model is built using the Keras library. The architecture consists of convolutional layers with max-pooling, followed by fully connected layers. The model architecture is as follows:

1. Input layer: Conv2D with 32 filters, kernel size (3, 3), and ReLU activation.
2. MaxPooling2D layer with pool size (2, 2).
3. Conv2D with 64 filters, kernel size (3, 3), and ReLU activation.
4. MaxPooling2D layer with pool size (2, 2).
5. Conv2D with 128 filters, kernel size (3, 3), and ReLU activation.
6. MaxPooling2D layer with pool size (2, 2).
7. Flatten layer to convert the 2D output into a 1D vector.
8. Dense layer with 128 units and ReLU activation, followed by Dropout (50%).
9. Output layer with units equal to the number of classes (expression labels) and softmax activation.

Model Compilation and Training

The model is compiled with the Adam optimizer and sparse categorical cross-entropy loss function. It is trained using the training data and validated using the validation data. The training process includes multiple epochs with batch processing. The training steps are as follows:

1. Compile the model with the Adam optimizer and sparse categorical cross-entropy loss.
2. Train the model using the train_generator and validate it using the val_generator.
3. Monitor the training and validation accuracy during each epoch.

Model Evaluation

The trained model is evaluated on the test set to assess its performance. The test accuracy is computed and printed. The evaluation steps are as follows:

1. Evaluate the model on the test_generator.
2. Calculate the test loss and test accuracy.
3. Print the test accuracy.

Performance Metrics

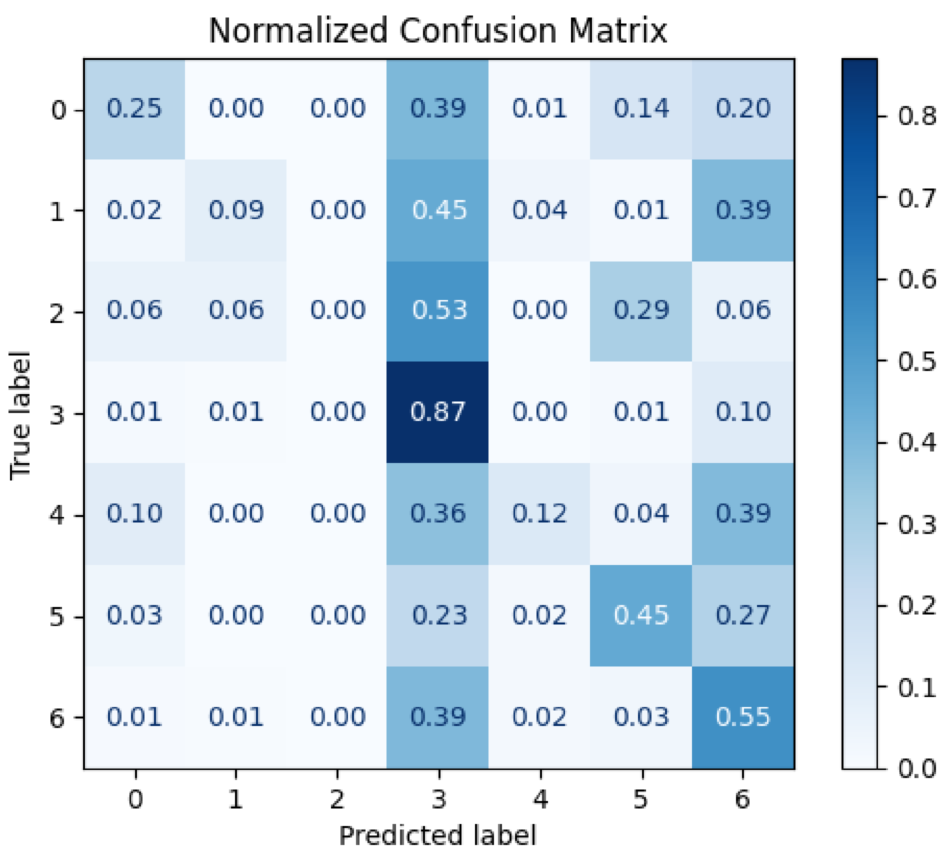
After evaluating the trained expression classification model, the following performance metrics were obtained:

- **Accuracy**: 0.6704
- **Precision**: 0.6370
- **Recall**: 0.6704
- **F1-Score**: 0.6434

Interpretation of Metrics

- **Accuracy**: The model correctly classified about 67.04% of the facial expression images.
- **Precision**: When the model predicted a particular expression, it was correct about 63.70% of the time.
- **Recall**: The model successfully identified about 67.04% of the actual positive instances.
- **F1-Score**: The model's overall performance in terms of precision and recall is balanced.

Confusion Matrix



Conclusion

The obtained results indicate that the model has achieved a reasonably good performance in classifying facial expressions. While the model demonstrates promising results, there's potential for refinement and optimization to achieve even better performance in facial expression classification. Further analysis and improvement could involve hyperparameter tuning, utilizing advanced architectures, or incorporating additional data augmentation techniques.

In conclusion, this project provides a foundation for further research and development in the field of expression classification and computer vision applications. The balance between precision and recall, as reflected in the F1-Score, indicates that the model has potential for real-world applications where both aspects are important.