FACIAL EMOTION RECOGNITION

Rohan B and Gunashree R

Department of Computer Science, RV University, Bengaluru, India

PROBLEM STATEMENT

Facial emotions play a crucial role in human communication and interaction. Recognizing these emotions from visual cues has numerous applications, including:

1. Human-Computer Interaction: Enhancing user experiences by adapting to emotions.
2. Psychology and Healthcare: Analyzing emotional well-being and mental health.
3. Security and Surveillance: Identifying individuals' emotional states in sensitive scenarios.

The objective of this project is to build a deep learning model that categorizes grayscale facial images into one of seven emotions: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. Using a dataset of labeled images, the model aims to achieve high accuracy and robustness in emotion recognition.

KEY STRENGTH

1. Clear Problem Statement: The project clearly outlines the problem and its potential applications.
2. CData Preprocessing: The preprocessing steps are well-defined and essential for model performance.
3. Model Architecture: The CNN architecture is suitable for image classification tasks.
4. Evaluation Metricst: The use of accuracy, precision, recall, and F1-score provides a comprehensive evaluation of the model's performance.emotional states in sensitive scenarios.

DATASET AND PREPROCESSING

Dataset The dataset consists of grayscale images divided into:

- Training set: Used for model learning.
- Test set: Used for performance evaluation.

Dataset Description

- The data consists of 48x48 pixel grayscale images of faces. The faces have been categorized into facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).
- The training set consists of 10,067 examples. The public test set used for the leaderboard consists of 7,060 examples. The final test set, which was used to determine the winner of the competition, consists of another 7,060 examples.

Data Preprocessing

Key preprocessing steps included:

- Grayscale Normalization: Pixel values scaled to the range [0, 1].
- Image Resizing: All images resized to 48×48 to standardize input dimensions.
- Label Encoding: Converting emotion labels into one-hot encoded vectors for classification.
- The fer2013.csv consists of three columns namely emotion, pixels, and purpose.
- The column in pixel first of all is stored in a list format.
- Since computational complexity is high for computing pixel values in the range of (0-255), the data in the pixel field is normalized to values between [0-1].
- The face objects stored are reshaped and resized to the mentioned size of 48 × 48.
- The respective emotion labels and their respective pixel values are stored in objects.
- We use scikit-learn's train-test-split() function to split the dataset into training and testing data. The test-size being 0.2 meaning, 20 percentage of data is for validation while 80 percentage of the data will be trained.

Data Augmentation To address data imbalance and increase dataset diversity, augmentation techniques were employed:

- Rotations: Random rotations to simulate varied head angles.
- Flips: Horizontal flips to add variability.
- Shifts and Zooms: Minor spatial adjustments to simulate natural variations.

MODEL ARCHITECTURE

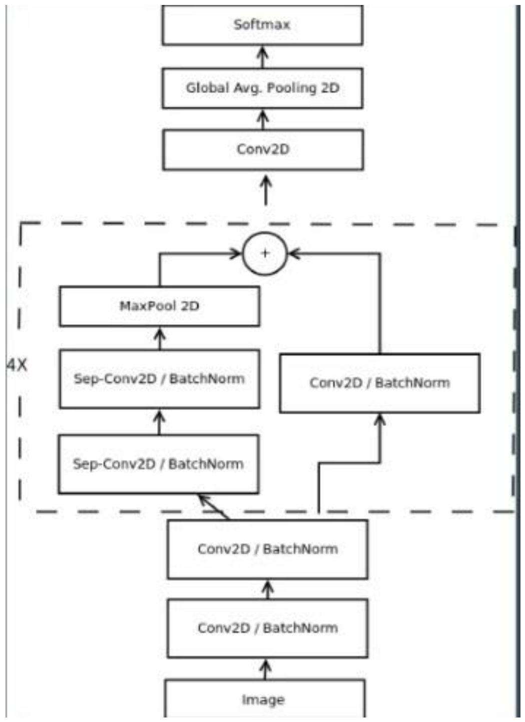


Fig. 1. : Model Architecture

The chosen approach utilized a custom Convolutional Neural Network (CNN) due to its ability to learn spatial features effectively.

Model Layers

- Convolutional Layers: Extract spatial features from images using filters of size 3×3.
- Batch Normalization: Accelerates training and improves convergence.
- MaxPooling Layers: Downsamples feature maps to reduce computational complexity.
- Dropout: Mitigates overfitting by randomly disabling neurons during training.
- Fully Connected Layers: Aggregate features for classification.
- Softmax Output Layer: Assigns probabilities to the seven emotion classes.

Layer Configuration

- Layer 1: 128 filters, Batch Normalization, MaxPooling, Dropout.
- Layer 2: 256 filters, Batch Normalization, MaxPooling, Dropout.
- Layer 3: 512 filters, Batch Normalization, MaxPooling, Dropout.
- Fully Connected Layers: Dense layers with 512 and 256 units, followed by Dropout.

TRAINING PROCESS

Hyperparameter Tuning

- Use automated hyperparameter tuning techniques like Grid Search, Random Search, or Bayesian Optimization.
- Experiment with different learning rate schedules and optimizers.
- Batch Size: 64
- Learning Rate: 0.0001 (with adaptive learning rate reduction on plateau).
- Optimizer: Adam optimizer for efficient training.
- Loss Function: Categorical Crossentropy.

Callbacks

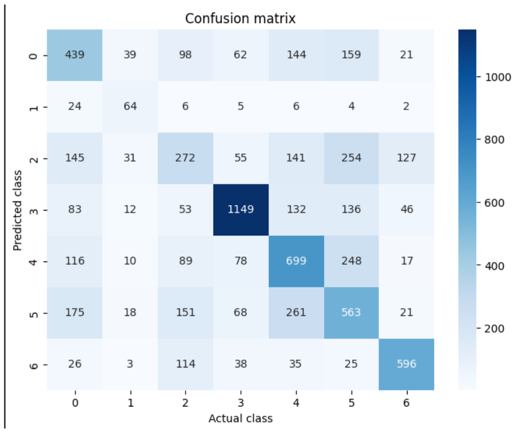
- Early Stopping: Stops training when validation loss does not improve for 20 epochs.
- Learning Rate Scheduler: Reduces the learning rate by a factor of 0.2 when validation loss plateaus.

Training Results

- Epochs: 100 (with early stopping).
- Validation Split: Model evaluated using a separate validation set during training.

PERFORMANCE EVALUATION

Confusion Matrix
Angry - [439 39 98 62 144 159 21]
Disgust - [24 64 6 5 6 4 2]
Fear - [145 31 272 55 141 254 127]
Happy - [83 12 53 1149 132 136 46]
Sad - [116 10 89 78 699 248 17]
Surprise - [175 18 151 68 261 563 21]
Neutral - [26 3 114 38 35 25 596]



Predicted class	0	1	2	3	4	5	6
0	439	39	98	62	144	159	21
1	24	64	6	5	6	4	2
2	145	31	272	55	141	254	127
3	83	12	53	1149	132	136	46
4	116	10	89	78	699	248	17
5	175	18	151	68	261	563	21
6	26	3	114	38	35	25	596

Fig. 2. : Confusion Matrix

EXPLANATION:
Rows: Represent the actual classes (true labels). Columns: Represent the predicted classes. Values: Indicate the number of instances that belong to the actual class (row) and were predicted as the class in the column.

EVALUATION METRICS

Accuracy: 53
Precision: 54
Recall: 53
F1-score: 53

ACCURACY AND LOSS GRAPH

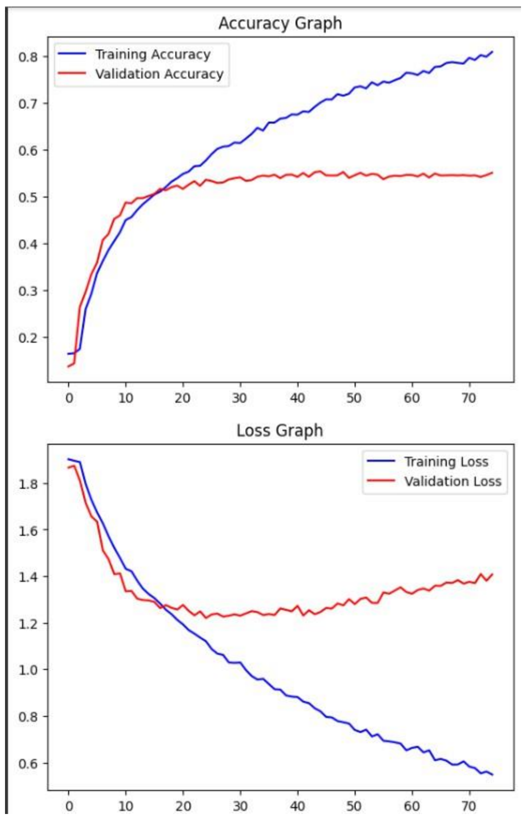


Fig. 3. : Accuracy and loss graph

Plotting Accuracy Graph:

- Extracts accuracy and validation accuracy from the training history.
- Plots them on a graph with the x-axis representing the number of epochs and the y-axis representing accuracy.
- Blue line ('b') shows training accuracy, and red line ('r') shows validation accuracy.
- The plt.legend() call adds a label to the graph for easy identification.

Plotting Loss Graph:

- Extracts loss and validation loss from the training history.
- Plots them similarly with epochs on the x-axis and loss on the y-axis.
- Training loss is shown in blue and validation loss in red.

Purpose:

- Accuracy Graph: Helps you see how well the model is learning during training and if it's generalizing well to validation data.
- Loss Graph: Shows how the error is reducing over time during training. This helps identify if the model is overfitting (training accuracy is high but validation accuracy is low) or underfitting (both are low).

CHALLANGES AND OBSERVATION

1. Class Imbalance: Some emotions, like 'Disgust', have fewer samples, leading to underperformance.
2. Feature Overlap: Emotions like 'Fear' and 'Sad' have overlapping visual features, causing misclassifications.
3. Dataset Limitations: Grayscale images provide less information compared to color images, potentially reducing model accuracy.

RECOMMENDATION FOR IMPROVEMENT

To enhance model performance, future iterations could explore:

- Advanced Architectures: Testing deeper networks or pre-trained models (e.g., VGG16, ResNet).
- Additional Augmentation: Introducing brightness, contrast, and noise augmentation for increased robustness.
- Transfer Learning: Leveraging pre-trained models to incorporate features from large-scale datasets.
- Class Rebalancing: Oversampling underrepresented classes or using class-weighted loss functions.
- Hyperparameter Optimization: Automated optimization using tools like Optuna or Hyperband.

CONCLUSION

This project successfully demonstrated the use of CNNs for facial emotion recognition. While the model shows promise in identifying certain emotions, further refinements are necessary to address class-specific challenges and improve overall accuracy. With advanced techniques and additional data, the system can become more robust and applicable to real-world scenarios.