High Impact Skills Development Program
Gilgit Baltistan

Expression Classification from Facial Images using CNN

Computer Vision Module Project

Faiza Ali

Email: faiza.gil.n@gmail.com

GitHub Profile: github.com/FaizaAli-Dev

Roll No: GIL23047

DSAI-Gilgit Section 1

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1. Summary

The project aims to develop a computer vision system using deep ConvNet to classify facial expressions from images. The task is challenging due to various factors such as intra-class variations, inter-class similarities, background clutter, illumination changes, pose variations, and occlusions. The system has potential applications in areas such as Human-Computer Interaction, recommendation systems, and psychological state analysis.

1.2. Aims and Objectives

1. Develop a facial expression classification model with high accuracy in recognizing diverse emotions (e.g., happiness, sadness, anger) from facial images.

2. Create a robust algorithm capable of handling real-world challenges, such as variations in lighting, poses, and occlusions, to ensure reliable performance in uncontrolled environments.

3. Enable emotion-aware applications by integrating the facial expression classification model into human-computer interaction systems, facilitating personalized responses based on users' emotional states.

1.3. Overview of the problem and potential application areas:

Expression Classification from Facial Images is a fascinating field of research in computer vision that focuses on automatically recognizing and classifying human facial expressions from images. Understanding and interpreting facial expressions is crucial for various applications, ranging from human-computer interaction to affective computing and beyond. Facial expressions convey valuable emotional and mental states of individuals, playing a significant role in human communication and understanding. By teaching machines to recognize facial expressions, we can enable more empathetic and interactive AI systems.

Facial expression classification from images is a challenging task due to various factors that can impact accuracy. However, solving this problem has real-world applications in several areas. Facial expression recognition enables Emotion-Aware Human-Computer Interaction, supporting personalized responses. In Affective Computing, it aids mental health monitoring. Additionally, it aids Human Behavior Analysis in social environments, offering valuable insights into emotional responses.

Chapter 2

REVIEW OF LITERATURE

Two recent articles from 2022-23 I have reviewed to understand the state of the art in facial expression classification. Article "Facial expression recognition using bidirectional LSTM – CNN" result show us the accuracy of 99.43% with data augmentation. Further in another research article "Facial emotion recognition using Handcrafted features and CNN" mentions that emotions are of utmost concern due to an immediate increase in a number of healthcare concerns such as depression, cancers, paralysis, trauma etc.

Features are extracted using HOG and SIFT technique in this following research which are further used to train neural network and the architecture i.e. explicit feature extraction coupled with convolution net and additional deep neural layers have been the reason for this outperforming accuracy. This model gives us accuracy of 98.48% for HOG-CNN and 97.96% for SIFT-CNN for CKplus dataset. Also for the Jaffe dataset the accuracies encountered were 91.43% for HOG-CNN while 82.85% for SIFT-CNN. The proposed model can be validated on other challenging facial emotion dataset

Chapter 3

REQUIREMENT SPECIFICATION

3.1. Model

The project utilized a deep ConvNet architecture for facial expression classification. The architecture consisted of 2 convolutional layers, followed by pooling and 1 fully connected layer. I implemented the model using TensorFlow and Keras libraries. The model's hyper parameters, such as learning rate and batch size, were tuned for optimal performance.

3.2. Required tools and technologies

I have used Google Colab a cloud-based Jupyter Notebook service that provides free access to GPUs and TPUs. Colab can accelerate the training of deep learning models.

TensorFlow: TensorFlow is an essential deep learning library that provides tools for building and training neural networks. I have imported it as tf.

Matplotlib: Matplotlib is a popular data visualization library in Python. It is used for plotting and displaying images or graphs. I have imported it as plt.

Keras: Keras is an easy-to-use deep learning library that works on top of TensorFlow. It allows us to build and train neural networks with a high-level API.

Chapter 4

DESIGN

4.1 Dataset:

The initial dataset “The Expression in-the-Wild (ExpW)” was consist of training and test dataset, that I further divided it into three sets: training, validation, and testing. The training set was used to train the model, the validation set was used for hyper-parameter tuning, and the testing set was used to evaluate the final model's performance. The report provided statistics about the dataset, including the number of images per class and the distribution of expressions.

The Expression in-the-Wild (ExpW) dataset is is used here and contains 91,793 faces manually labeled with expressions. Each of the face images is annotated as one of the seven basic expression categories: “angry (0)”, “disgust (1)”, “fear (2)”, “happy (3)”, “sad (4)”, “surprise (5)”, or “neutral (6)”.

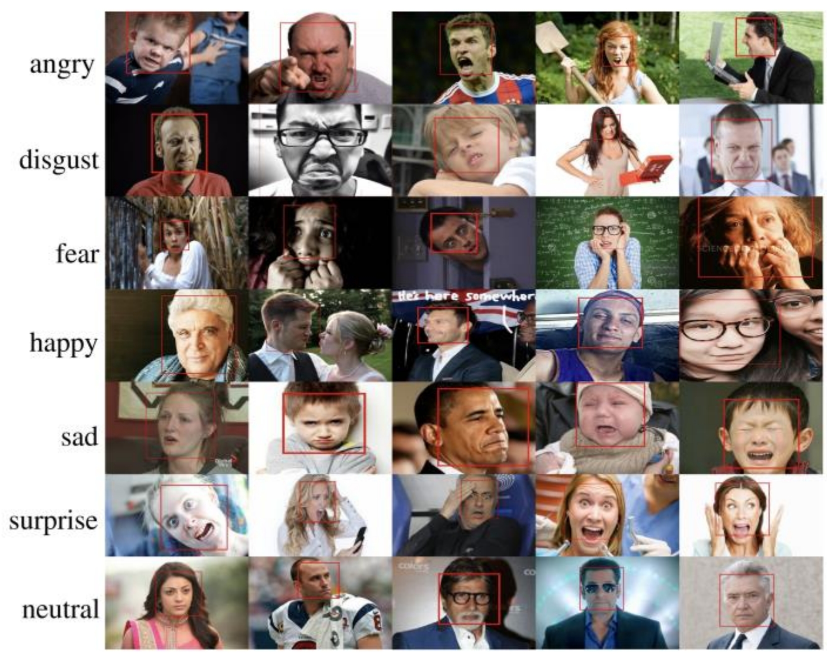

Figure 1: Sample Images from ExpW Dataset

System Architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 32)        320

 max_pooling2d (MaxPooling2D  (None, 24, 24, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 24, 24, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 64)       0
 2D)

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 256)               2359552

 dense_1 (Dense)             (None, 512)               131584

 dense_2 (Dense)             (None, 7)                 3591

=================================================================
Total params: 2,513,543
Trainable params: 2,513,543
Non-trainable params: 0
```

Figure 2Model details

Figure 3 General Information

Chapter 5

SYSTEM IMPLEMENTATION

The data set provided was consist of more than 90,000 images (train, test and labels), that I split it into train, validation and test for further processing. I have used 2 convolutional layer with 64 batches and Adam optimizer.

5.2 Process Flow

1. Dataset selection
2. Load dataset, and link colab with google drive to get the data in unzipped format
3. Define dataset path
4. Data Preprocessing
5. Split the data into Training/Validation/Test
6. Define the CNN model architecture
7. Train the model using fit function
8. Evaluate the performance of the model by measuring its accuracy, precision, recall, and F1 Score.
9. Tested the model on manual images from each class

5.2 Hyperparamerter Tuning

To optimize the model's performance, hyper-parameter tuning was conducted using the validation set. The report described the hyperparameters that were tuned, the range of values tested, and the evaluation metrics used to select the best set of hyperparameters.

Chapter 6

SYSTEM TESTING AND EVALUATION

6.1 Results and Evaluations:

The report presented the results of the facial expression classification task. The model's performance was evaluated using accuracy metrics. We get 60% accuracy of the model. A confusion matrix was provided to gain deeper insight into the model's performance, identifying which expressions were correctly classified and which ones had higher confusion.

6.2 Analysis of Results:

The result provided insights into the challenges faced by the model, such as inter-class similarities and occlusions. The confusion matrix helped identify specific expressions that were more difficult to classify accurately. Possible reasons for misclassifications were discussed, such as ambiguous facial expressions or limitations of the dataset.

6.3 Further Improvements:

To enhance the results, the report suggested several potential avenues for improvement. These included exploring larger and more diverse datasets, leveraging state-of-the-art pre-trained models for feature extraction, experimenting with different network architectures, and incorporating advanced techniques such as attention mechanisms or ensemble learning.

Chapter 7

CONCLUSION

In conclusion, the project focused on developing a computer vision system for facial expression classification. The report provided a comprehensive overview of the problem, reviewed recent literature, described the model architecture, discussed the dataset and hyperparameter tuning, presented the results and evaluations, analyzed the findings, and proposed ways to improve the system's performance.

REFERENCES

1. C. Gautam and S. K. R., "Facial Emotion Recognition Using Handcrafted Features and CNN," in IEEE Xplore, Department of Computer Science & Engineering, Indira Gandhi Delhi Technical University for Women, Delhi, 110006, India.2023

2. R. Febriana, B. M. Halima, M. Christinaa, D. Ramdhana, and A. Chowandaa, "Facial Expression Recognition Using Bidirectional LSTM-CNN," presented at the 7th International Conference on Computer Science and Computational Intelligence, 2022.