Computer Vision

Project Report

Dataset:

[Dataset](https://drive.google.com/drive/folders/1GAlWh176Gwo5JBZyUTjgLfsSETC2S-e_?usp=sharing)

# Report:

The emotion detection model implemented in the provided code is designed to detect expressions of happiness, anger, neutrality, and surprise. It uses a Convolutional Neural Network (CNN) architecture to learn and classify images into these emotion categories. Here's a summary of the model and its implementation:

# Data Preparation:

* The data-set is loaded using an ImageDataGenerator from the specified directory.
* The images are re-scaled to a target size of 256x256 pixels and converted to grayscale.
* The class mode is set to categorical for one-hot encoding of the labels.
* The data was of total 37 images consisting of labels: Anger, Happy, Neutral and Surprise.

# Data Augmentation:

* The code performs data augmentation to increase the size of the data-set.
* For each image in the data-set, it creates 10 augmented versions by applying random affine transformations, such as shear.
* The augmented images and their corresponding labels are stored in separate arrays.
* After augmentation dataset was increased to 840 images

# Model Architecture:

* The model is implemented using the Keras API with a Sequential model.
* It consists of several layers, including Conv2D, MaxPooling2D, Flatten and Dense
* The model uses the Adam optimizer with adjusted learning rate.
* The loss function used is categorical cross-entropy, and accuracy is used as the evaluation metric.

# Model Training and Evaluation:

* The code implements k-fold cross-validation with 20 folds to train and evaluate the model.
* For each fold, the model is trained for 10 epochs on the training data and evaluated on the test data.
* The accuracy, confusion matrix, and other performance metrics are calculated for both the training and test sets.
* The model's performance is summarized by printing the accuracy mean and standard deviation and plotting box and whisker plots.

# Graphs and Visualization:

* The code includes functions to plot diagnostic learning curves and training/validation graphs for each fold.
* It also plots the average loss for training and validation data across all folds.

# Image Capture and Prediction:

* The code provides functionality to capture an image using the device's camera.
* The captured image is preprocessed to match the input requirements of the model.
* The model predicts the emotion label for the captured image and displays the result.

# Result:

* The model was run on multiple fold and got a high Train test Accuracy

Overall, the implemented emotion detection model demonstrates the process of training a CNN using augmented images and evaluating its performance. The model can be further optimized by adjusting hyperparameters, exploring different architectures, or increasing the dataset size.