**Computer vision module project  
report**

**Sheraz karim**

**Date :- 2024-10-21**

**Github repo link :-**  https://github.com/Sherazkarim1/face\_expression\_recognition.git

This repository contains a machine learning project for recognizing and classifying human facial expressions. By analyzing images or real-time video input, the model detects emotions such as happiness, sadness, anger, surprise, and more. The goal is to provide a robust solution for emotion detection, which can be applied in areas like security, entertainment, healthcare, and human-computer interaction.

GitHub Repository: https://github.com/Sherazkarim1/face\_expression\_recognition.git

**Abstract :-**

This project aims to classify facial expressions using a Convolutional Neural Network (CNN) model trained on a Kaggle facial expression recognition dataset. The dataset comprises images of human faces labeled with emotions such as happiness, sadness, and anger. The images were preprocessed through resizing, normalization, and data augmentation techniques to enhance the model's performance. The CNN model was trained over 50 epochs, achieving a validation accuracy of approximately 44.67%. While the accuracy indicates moderate success in recognizing facial expressions, the results suggest room for improvement through model fine-tuning and advanced deep learning techniques. This project demonstrates the potential of CNNs in the domain of emotion detection from facial images.

**Introduction :-**

Model Architecture:

The model used in this project is a Convolutional Neural Network (CNN) designed to classify facial expressions into various emotion categories. CNNs are widely regarded as effective for image-based tasks due to their ability to capture spatial hierarchies and patterns in images. Below is a detailed breakdown of the model architecture:

1. Input Layer: The input to the model consists of images resized to 64x64 pixels with 3 color channels (RGB). The images were preprocessed by normalizing pixel values between 0 and 1 to ensure consistent data input across the model.
2. Convolutional Layers: Several convolutional layers were employed to extract features from the images. Each convolutional layer applies a series of filters to detect different patterns such as edges, textures, and complex facial features. These layers use ReLU (Rectified Linear Unit) activation functions to introduce non-linearity, allowing the network to model complex relationships in the data. Key hyperparameters in these layers include:
   * Filter size: 3x3
   * Stride: 1
   * Padding: 'same' to preserve the spatial dimensions of the output.
3. Max Pooling Layers: After each convolutional layer, Max Pooling layers are added to downsample the feature maps, reducing the computational complexity and focusing on the most significant features. Max pooling was performed with a 2x2 window, effectively halving the dimensions of the feature maps.
4. Fully Connected Layers: After the convolutional and pooling layers, the output is flattened and passed through fully connected (dense) layers. These layers connect every neuron from the previous layer to the neurons in the current layer, combining the high-level features extracted by the convolutional layers.
5. Dropout: To prevent overfitting, dropout layers were used during training, which randomly set a fraction of the input units to zero at each update during the training phase. This helps the model generalize better on unseen data.
6. Output Layer: The final layer is a fully connected layer with a softmax activation function. The softmax function converts the logits into probabilities, and the model outputs a probability distribution over the emotion classes. The class with the highest probability is selected as the predicted emotion.
7. Loss Function: The model was trained using the categorical cross-entropy loss function, which is commonly used for multi-class classification problems. This loss function measures the difference between the predicted probability distribution and the actual distribution (ground truth).
8. Optimizer: The Adam optimizer was employed to minimize the loss during training. Adam is an adaptive learning rate optimization algorithm that combines the advantages of both RMSProp and Stochastic Gradient Descent (SGD), resulting in faster convergence and better performance.

Summary of Layers:

* Input: 64x64x3 images
* Conv2D + ReLU + Max Pooling (several layers)
* Flatten
* Fully Connected (Dense) Layers + Dropout
* Softmax Output Layer for classification

By stacking multiple convolutional and pooling layers, the network effectively learns spatial hierarchies in the image data, enabling it to classify facial expressions with increasing accuracy as the training progresses.

Training Data: Expression in the Wild (ExpW) Dataset

The Expression in the Wild (ExpW) dataset is a large-scale collection of facial images that are annotated with seven different expression labels. This dataset is specifically designed to recognize facial expressions in real-world, unconstrained settings, making it particularly challenging and valuable for training deep learning models.

Key features of the ExpW dataset:

1. Real-World Images: The images in this dataset are sourced from the wild, meaning they contain varying lighting conditions, poses, and backgrounds. This makes the dataset more realistic compared to controlled environments and enhances the model’s generalizability.
2. Expression Categories: The dataset includes the following seven expression labels:
   * Happy
   * Sad
   * Angry
   * Surprise
   * Disgust
   * Fear
   * Neutral
3. Diverse Demographics: The dataset consists of faces from people of various ages, genders, and ethnicities, further increasing the complexity and diversity of the images.
4. Image Variability: Images in the ExpW dataset vary in resolution, quality, and pose. Some faces are partially occluded by objects such as glasses or hands, while others may have varied expressions due to head tilts or different facial orientations.
5. Dataset Size: The ExpW dataset contains tens of thousands of images, providing ample data for training deep learning models like CNNs. The large size ensures the model has access to diverse training examples, improving its robustness to noise and variations in facial expressions.

Data Preprocessing: To prepare the ExpW dataset for training, the following preprocessing steps were applied:

The Expression in the Wild (ExpW) dataset contains approximately 91,793 images of faces. Each image is labeled with one of seven facial expression categories, making it a comprehensive dataset for facial expression recognition tasks.  
Contains labeled data.

Workflow for Facial Expression Recognition Using the ExpW Dataset

The overall workflow of the project involves several stages, from data preprocessing to model evaluation. Below is a step-by-step breakdown of the workflow:

### 1. **Data Collection**

* The dataset used in this project is the Expression in the Wild (ExpW) dataset, which contains approximately 91,793 facial images annotated with seven expression labels (happy, sad, angry, surprised, disgusted, fear, and neutral).

### 2. **Data Preprocessing**

* Image Resizing: All images were resized to a uniform size of 64x64 pixels to ensure consistent input to the CNN model.
* Normalization: The pixel values of the images were scaled between 0 and 1 by dividing the pixel intensities by 255. This normalization step is crucial to speed up convergence during training.
* Data Augmentation: To make the model more robust and prevent overfitting, data augmentation techniques were applied. These techniques include:
  + Random horizontal flips
  + Random rotations
  + Zooming in/out
* Splitting the Dataset: The dataset was split into training and validation sets to monitor the model's performance and prevent overfitting. The majority of the data was used for training, while a smaller subset was reserved for validation.

### 3. **Model Design**

* Convolutional Neural Network (CNN):
  + The CNN model was designed with multiple layers, including convolutional, max-pooling, and fully connected layers. The architecture was tuned to capture hierarchical patterns in facial expressions.
* Activation Function: ReLU (Rectified Linear Unit) was used as the activation function for all layers except the final output layer, which used a softmax function to produce probabilities for each expression class.

### 4. **Model Compilation**

* The model was compiled using:
  + Loss Function: Categorical Cross-Entropy, which is suitable for multi-class classification problems.
  + Optimizer: Adam optimizer, which adapts the learning rate during training and provides faster convergence.
  + Metrics: Accuracy was used as the primary evaluation metric to assess the performance of the model on both training and validation datasets.

### 5. **Training the Model**

* The model was trained over 50 epochs, with each epoch involving a forward pass through the training data and a backward pass to update the weights.
* Batch Size: A batch size of 32 was used to optimize memory usage while ensuring efficient training.
* Monitoring Performance: During training, the model’s performance was monitored using validation data to ensure it was learning effectively and to detect any signs of overfitting.

### 6. **Evaluation**

* After training, the model’s accuracy and loss on the validation set were evaluated to assess its generalization performance. The results showed that the model steadily improved its accuracy, reaching approximately 44.67% validation accuracy after 13 epochs.
* Confusion Matrix: A confusion matrix was generated to understand the model’s strengths and weaknesses in classifying different facial expressions.

Results and Evaluation

After training the Convolutional Neural Network (CNN) on the Expression in the Wild (ExpW) dataset, the model's performance was evaluated on both the training and validation datasets. The results show a steady improvement in accuracy over the course of 50 epochs, reflecting the model's ability to learn and generalize features related to facial expressions.

### 1. **Training and Validation Accuracy**

The CNN model was trained for 50 epochs, with the training and validation accuracies improving incrementally. Below are the accuracy and loss details at key epochs:

* Epoch 1:
  + Training Accuracy: 17.47%
  + Validation Accuracy: 21.93%
  + Training Loss: 1.9371
  + Validation Loss: 1.8602
* Epoch 5:
  + Training Accuracy: 28.38%
  + Validation Accuracy: 33.24%
  + Training Loss: 1.7744
  + Validation Loss: 1.6706
* Epoch 10:
  + Training Accuracy: 36.90%
  + Validation Accuracy: 42.04%
  + Training Loss: 1.6063
  + Validation Loss: 1.4845
* Epoch 13:
  + Training Accuracy: 40.21%
  + Validation Accuracy: 44.67%
  + Training Loss: 1.5315
  + Validation Loss: 1.4434

The model achieved a maximum validation accuracy of 44.67% at epoch 13, indicating a moderate performance in classifying facial expressions and at the 50th epoch the **validation accuracy was 61.38%.**

**Conclusion :-**

In this project, a Convolutional Neural Network (CNN) was developed to classify facial expressions using the Expression in the Wild (ExpW) dataset, achieving a test accuracy of 61.38%. The model demonstrated a stronger ability to recognize facial expressions under real-world conditions, improving its generalization to previously unseen data. Despite this improvement, challenges remain due to the dataset's variability in pose, lighting, and occlusion. Further enhancements, such as applying transfer learning, refining the network architecture, and addressing class imbalance, could push the performance even higher. Overall, the project provides a solid base for advancing facial expression recognition with promising results.

**Related screen shots :-**

