**FACIAL EXPRESSION USING IMAGE CLASSIFICATION**

**TEAM 20:**

D. VISWANADHA VARMA (21B21A44C6)

M. EBRAHIM BAIG (21B21A44D0)

B. V. S. PRABHATH (21B21A44C4)

M. SARATH CHANDRA (21B21A44C2)

A. VEERA GANDADHAR (21B21A44D7)

As the technology is growing immensely in recent years, certain improvisations needs to be made. From the tools we use every day to the groundbreaking innovations that transform industries, technology is a fundamental force that continues to advance human capabilities and improve our lives. This title could be a valuable for Understanding Human Emotions, Practical Applications, Technological Advancement

We’ll be using OpenCV library for implementing our project. OpenCV is a huge open-source library for Computer Vision, Machine Learning and Image processing. OpenCV supports a wide variety of programming language like Python, C++ etc. It can process image and videos to identify facial expressions, finger prints.

**Modules required:**

* Tensorflow
* Keras
* Pandas
* Numpy
* Jupyter
* Opencv - contrib - python
* Scikit-learn

To top off the project, certain steps have to be noted:

**Input:**

Images are input to the CNN after being pre-processed (grayscale, resized, normalized). Labels are encoded to categorical values.

**Pre-processing:**

Grayscale conversion

Resizing to 48x48

Normalizing pixel values

**Model Prediction:** The CNN processes the image, and the emotion with the highest probability is predicted.

**Processing:**

1. Dataset Preparation

Datasets: You’ll need a labeled dataset of facial images with various expressions. Common datasets for this task include:

FER-2013: Facial Expression Recognition dataset with 7 classes (anger, disgust, fear, happiness, sadness, surprise, and neutral).

AffectNet: A large dataset with diverse facial expressions.

CK+ (The Extended Cohn-Kanade Dataset): A dataset of facial expressions with both positive and negative emotions.

You can use the Keras ImageDataGenerator class to load images and perform data augmentation for improving model generalization.

Face Detection: Before feeding the image into the model, you need to detect the face in the image. Libraries like OpenCV (Haar Cascade or Dlib) or MTCNN can help detect faces.

Grayscale Conversion: To reduce complexity and focus on features that convey emotion, converting images to grayscale is often done.

Resizing: Standardize the size of the image to the input shape expected by the model (e.g., 48x48 or 64x64).

Normalization: Scale pixel values to a range of [0, 1] by dividing by 255.

1. Feature Extraction

HOG (Histogram of Oriented Gradients): Useful for capturing features related to facial structures and expressions.

CNN (Convolutional Neural Networks): Automatically extracts features through convolution layers.

1. Model Architecture

A Convolutional Neural Network (CNN) is the most commonly used approach for image classification tasks like facial expression recognition

1. Model Training

Loss Function: categorical\_crossentropy is used for multi-class classification.

Optimizer: Adam is commonly used for training CNNs.

Metrics: Accuracy is commonly used to measure the performance of the model.

1. Data Augmentation

Using Keras’s ImageDataGenerator for augmenting the data (rotation, zoom, etc.) can improve generalization and performance

1. Evaluation and Fine-Tuning

After training, evaluate the model using the test dataset to see its performance and use techniques like k-fold cross-validation for better results. You can also fine-tune the model by adjusting hyperparameters or using transfer learning.

It mainly focuses on accurately identifying and categorizing the emotional state of a person based on their facial expressions in an image.

* Identifying basic emotions:

1. Happiness
2. Sadness
3. Anger
4. Fear
5. Neutral

* Recognizing Complex Emotions:

1. Surprise
2. Disgust

**Output:**

**Static Image Prediction:** The trained model predicts the correct emotion for static images. For example, an image of a "sad" face is correctly labeled as "sad."

**Real-Time Emotion Detection**: The webcam feed is processed to detect faces, extract features, and predict emotions for each face in real-time. The predictions are displayed on the screen with labels like "happy," "sad," "angry," etc.

**Performance:** The model seems to perform well on both static images and real-time webcam feeds, correctly identifying various emotions.

In summary, facial expression recognition using image classification focuses on building systems that can accurately, robustly, and efficiently identify human emotions from images, while also addressing ethical considerations related to privacy and bias. The model performs effectively on both the training and testing datasets, with the predictions being visually confirmed by displaying the images along with the predicted labels. The real-time webcam emotion detection feature is a practical demonstration of how this model can be used in real-world applications, like facial expression-based human-computer interaction.