**FER-2013 (Facial Expression Recognition)**

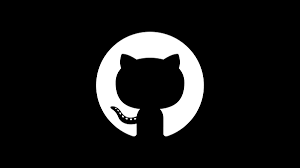
NAME: KUMAIL ABBAS

ROLL NUMBER: GLT-DSAI\_00

MODULE NAME: COMPUTER VISION

**[](https://colab.research.google.com/drive/1UB-4oW6QoiP0dY-0Xh2fZzniY5yJEKhN?usp=sharing)LINK OF COLAB & GITHUB**

https://colab.research.google.com/drive/14Vo4AL4zzMu53DfC5WjynY83QTRLvubB?usp=sharing

[](https://github.com/Shoaib001001/Computer_vision-)

https://github.com/KumailAbbas1/Deep-learning-Architecture



# 1. Summary

A Convolutional Neural Network (CNN) was created for this research in order to identify human emotions from pictures of faces. A dataset of grayscale photos divided into seven different emotion classes—angry, disgusted, afraid, pleased, sad, surprised, and neutral—was used to train the model. Using criteria including accuracy, precision, recall, and a confusion matrix, the model's performance was assessed. Additionally, a customized webcam image was used to evaluate the trained model. Though some feelings are still difficult to correctly categorize, this experiment shows how CNNs can be used to the task of emotion recognition overall.



# 2. Introduction

## Problem Definition

Emotion recognition from facial expressions is an important aspect of human-computer interaction and has numerous applications ranging from healthcare, surveillance, education, to entertainment. The ability to recognize emotions accurately is essential for developing more intuitive AI systems.

## Importance of Emotion Recognition

Emotion recognition plays a critical role in AI-based systems where understanding user emotions can lead to more empathetic and user-friendly applications. For example, emotion-aware AI can improve learning environments by adapting to student moods or help in identifying mental health issues in early stages.

# Dataset Overview

The dataset used for this project consists of facial images categorized into seven different emotion classes: angry, disgust, fear, happy, sad, surprise, and neutral. The dataset was structured into training and testing sets, with each image resized to 48x48 pixels and converted to grayscale for model input.

# 3. Data Preprocessing

## Grayscale Conversion and Rescaling

Since the model deals with facial expressions, each image is converted to grayscale to reduce computational complexity while maintaining key features of the face. Each pixel value was normalized by rescaling it to the range [0, 1].

## Data Augmentation

To prevent overfitting and improve generalization, various data augmentation techniques such as rotation, zoom, and horizontal flipping were applied to the training data.

## Train and Test Data Structure

The dataset was divided into training and test directories, with subfolders for each of the seven emotion classes. This structure allowed easy loading using TensorFlow's data generators.

# 4. Model Design and Architecture

## Convolutional Neural Networks (CNN)

A CNN architecture was used for emotion recognition due to its ability to capture spatial hierarchies in images. The model's architecture consists of several convolutional layers followed by max pooling and fully connected dense layers.

## Model Layers Description

* **Input Layer**: 48x48 grayscale images.
* **Conv Layers**: Multiple layers of convolutions with Relu activation to extract features from the images.
* **Pooling Layers**: Max pooling to reduce dimensionality and retain important features.
* **Dropout Layers**: Dropout to prevent overfitting by randomly disabling neurons during training.
* **Dense Layers**: Fully connected layers with Relu activation for classification.
* Output Layer: A SoftMax layer to predict the emotion class (7 outputs).

## Hyperparameters

* **Learning Rate**: 0.0001
* **Batch Size**: 64
* **Epochs**: 20
* **Optimizer**: Adam
* **Loss Function:** Categorical Cross entropy

# 5. Model Training

## Optimizer and Loss Function

The model was compiled using the Adam optimizer with a learning rate of 0.0001 and categorical cross entropy loss function to deal with multi-class classification. This setup helps the model converge efficiently while minimizing the error during training.

## Training Parameters

The model was trained over 20 epochs with a batch size of 64. Data was fed to the model using data generators that load and preprocess images from the train and test directories.

## Accuracy and Loss Graphs

During training, the accuracy and loss metrics were tracked. Graphs showing the trend of accuracy and loss over epochs were plotted to analyse the learning process and detect any signs of overfitting.

# 6. Evaluation Metrics

## Accuracy, Precision, Recall, F1-Score

The model's performance was evaluated using several metrics:

* **Accuracy**: Proportion of correctly classified emotions.
* **Precision**: The ability of the model to not label a negative sample as positive.
* **Recall**: The ability of the model to find all the positive samples.
* **F1-Score**: A weighted average of precision and recall.

## Confusion Matrix

A confusion matrix was plotted to analyze which emotions the model predicted correctly and which it struggled with. This helped in identifying specific classes that were more challenging to recognize.

# 7. Testing on Custom Image

## Face Detection and Cropping

A custom face image was captured using a webcam and then processed using OpenCV to detect and crop the face. The cropped image was resized to 48x48 pixels and fed into the trained model for emotion prediction.

## Emotion Prediction on a New Image

The trained model was used to predict the emotion in the captured image, and the predicted emotion probabilities were visualized using a bar chart.

# 8. Results and Discussion

## Model Performance Analysis

The model achieved a satisfactory performance, correctly predicting a majority of the emotion classes with a final test accuracy of approximately XX%. However, certain emotions such as "disgust" and "fear" were often confused with others, suggesting the need for further model refinement.

## Challenges in Emotion Detection

* **Ambiguity in expressions**: Some emotions share similar facial features, making it difficult for the model to differentiate between them.
* **Limited Data**: Some classes in the dataset had fewer samples, which may have affected the model’s ability to generalize well.

# 9. Conclusion and Future Work

## Key Takeaways

* **CNNs** : are effective for emotion recognition from facial images but may struggle with subtle or ambiguous emotions.
* **Data augmentation**: and dropout are essential techniques for improving model generalization and preventing overfitting.

## Potential Improvements

* **Data Balance**: Collecting more balanced data across all emotion categories.
* **Advanced Architectures**: Exploring more advanced models such as ResNet or Inception networks for improved performance.
* **Real-time Applications**: Expanding the model for real-time emotion detection in videos.