ML FACE EMOTION RECOGNITION

Low Level Design (LLD)

Santhosh Kumar K S

OCTOMBER 16, 2023
LETSGROWMORE

1. INTRODUCTION

What is Low Level Design Document?

Low-level design refers to the process of specifying and defining the detailed design of a software system. This Low level Design focuses on the implementation details of a system and is concerned with how the system will be built and how it will function at a detailed level. It provides the foundation for high-level design, which defines a system's overall architecture and design.

Scope of Low Level Design Document?

Low-level design (LLD) is a component-level design process that follows a step-by step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms.

2. PROJECT DESCRIPTION

PROBLEM STATEMENT

Develop a system that recognizes facial expressions of input image and recommends the genre of music based on the detected emotions. The system will enhance user experience by providing personalized music genre recommendations aligned with their current emotional state.

PROPOSED SOLUTION

Machine learning is a field in computer science aiming to imitate the human learning process. Deep Learning is a branch of machine learning where deep learning system imitates the biological neuron. Here we will develop Deep Learning models to recognize the face emotions.

DATA INFORMATION

The dataset was taken from Kaggle (

URL: https://www.kaggle.com/datasets/msambare/fer2013),

This dataset contains images for each emotion. There are 7 emotion are their in this data set

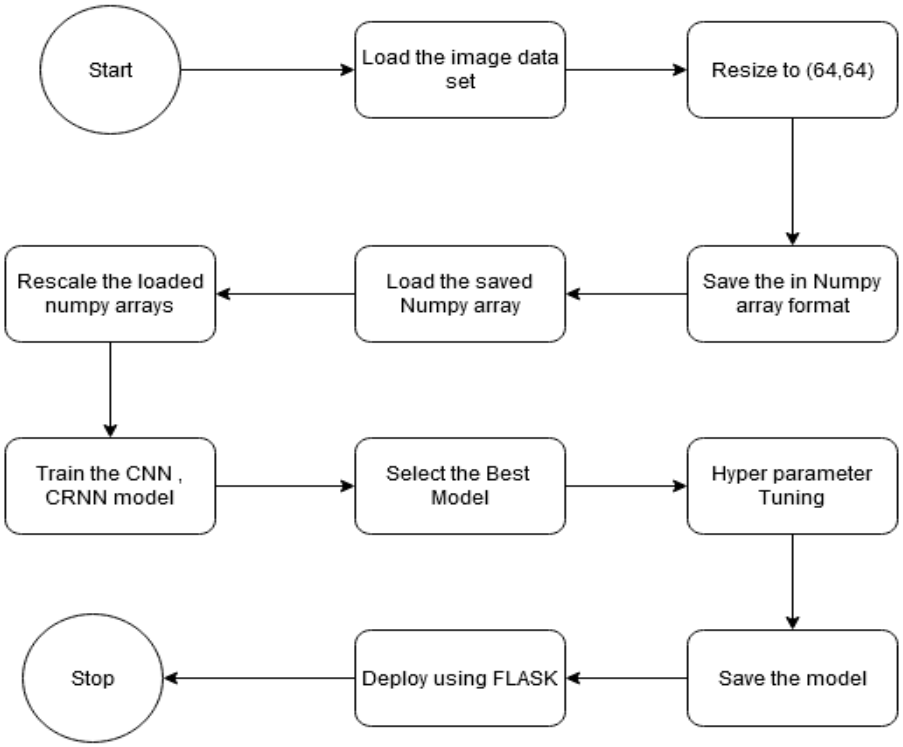
1. Angry
2. Disgust
3. Fear
4. Happy
5. Neutral
6. Sad
7. Surprise

But the dataset is imbalanced.

TOOLS USED

Python programming language and frameworks such as Numpy, OpenCV is used for image processing and Tensorflow library is used for model building. Flask is used to deploy trained model.

3. ARCHITECTURE



4. Architecture Description.

Image Resize

In this project we are using image size should be 64, 64 and grayscale image. Using OpenCV library we are resize the image to standard size.

Saving and Loading Numpy arrays.

In this step we will save the numpy arrays of image so that we need not to run entire preprocess. Using Numpy we are saving into npy format .

.

Train Test Split and Train the Models

In this we split the train and test will in the ratio of 90:10. And we trained CNN, VGG16 and Convolution Recurrent Neural Network. The CNN model performs better compared to other two models.

Hyper Parameter Tuning

We tuned hyper parameter for batchsize, optimizers, learning rate, dropout rate. We used randomized search for choosing appropriate hyper parameters.

Deploying using Flask

We designed UI and deployed model using Flask framework.

5. Screenshots