REAL-TIME EMOTION DETECTION USING DEEP LEARNING

Understanding Real Time Facial Emotions with Convolutional Neural Networks

Roop Chandrika Mallela

Lalithya Kalluri

Sachin Dudam

Shiva kumar Goud Mucharla

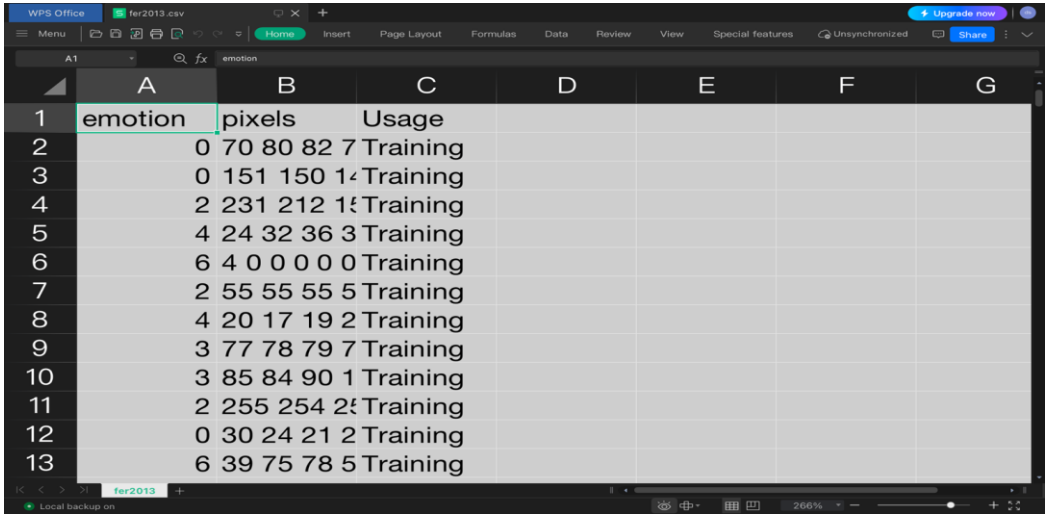
CONTENTS

INTRODUCTION (General Description of Project)

- Emotion detection involves using machine or deep learning techniques to detect and analyze the emotions of images.
- Real-time emotion detection can be used for capturing student's emotions in the class during the session.
- Our project is to detect the emotion on a student's face by capturing them real-time and classify their emotions into one of **seven categories**, using deep convolutional neural networks.
- The model we used is trained on the **FER-2013** dataset which was published on International Conference on Machine Learning (ICML). This dataset consists of 35887 grayscale, 48x48 sized face images with **seven emotions** - angry, disgusted, fearful, happy, neutral, sad and surprised.
- After training of the model, we are loading the same model to detect the real-time emotions of student's in the classroom sessions.

SAMPLE DATA PICTURE

The following picture is the sample data of FER 2013 we have used:-



	A	B	C	D	E	F	G
1	emotion	pixels	Usage				
2	0	70 80 82 7	Training				
3	0	151 150 1	Training				
4	2	231 212 1	Training				
5	4	24 32 36 3	Training				
6	6	4 0 0 0 0 0	Training				
7	2	55 55 55 5	Training				
8	4	20 17 19 2	Training				
9	3	77 78 79 7	Training				
10	3	85 84 90 1	Training				
11	2	255 254 2	Training				
12	0	30 24 21 2	Training				
13	6	39 75 78 5	Training				

BUSINESS PROBLEM

Our business problem is to develop a real-time emotion detection technology to analyze students' emotions during classroom sessions. Using machine and deep learning techniques, we aim to capture students' facial expressions and classify them into seven emotion categories: angry, disgusted, fearful, happy, neutral, sad, and surprised. Implementing deep convolutional neural networks trained on the FER-2013 dataset, containing 35,887 grayscale face images, our goal is to increase classroom engagement and also to improve teaching effectiveness by providing instant feedback on students' emotional states.

PROJECT FLOW

The flow of project is as follows, we have ran everything on Jetson Nano:-

- Firstly, we have downloaded the FER 2013 dataset from kaggle website (https://www.kaggle.com/datasets/deadskull7/fer2013).
- The above dataset contains code of emotions (0 to 6) and pixels of images which are of 35887 grayscale and 48x48 sized face images.
- In the next step, we have divided the dataset into train and test dataset. We have also divided the images into seven different emotions with respect to the emotion codes in the data.
- After which, we have created a Convolutional Neural Network (CNN) Model (In this code, we have added two modes- one is for training and other is for starting the webcam to capture the real-time emotions), plotted the model accuracy, model loss. (python emotions.py - -mode train)
- Now, we have used the above created model and launched the webcam to capture the real-time emotions. (python emotions.py - -mode test)

REQUIREMENTS

SOFTWARE REQUIREMENTS

- ✓ Mobaxterm
- ✓ Tensorflow
- ✓ OpenCV
- ✓ Python3

HARDWARE REQUIREMENTS

- ✓ Jetson Nano
- ✓ Webcam
- ✓ USB Connector to connect Jetson Nano and PC

MODEL USED

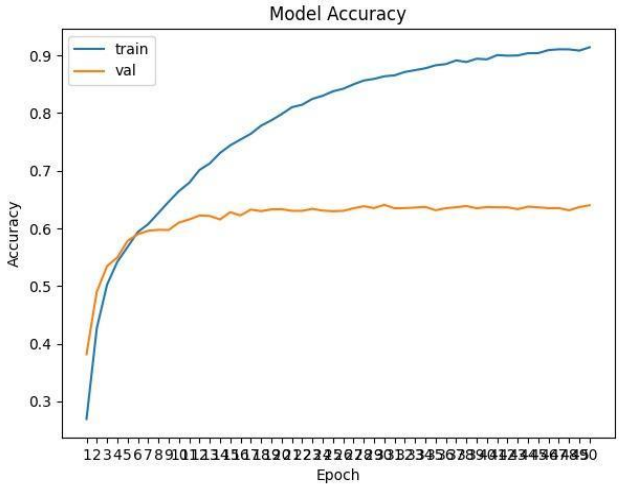
- The model we have used is Convolutional Neural Networks (CNN) which is a deep learning technique.
- A **Convolutional Neural Network (CNN)** is a type of Deep Learning neural network architecture commonly used in Computer Vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data. It consists of three layers – input layer, hidden layer and output layer.
- **Model Architecture**
  - ✓ Utilizes a Convolutional Neural Network (CNN)
  - ✓ Input: Grayscale face images resized to 48x48 pixels
  - ✓ Output: Seven softmax scores corresponding to different emotions
- **Algorithm Overview**
  - ✓ Face Detection: Haar Cascade method to detect faces in webcam feed
  - ✓ Preprocessing: Resize detected face images to match input size of the CNN
  - ✓ Prediction: CNN predicts the dominant emotion for each face
  - ✓ Display: Emotion with maximum score is displayed on the screen

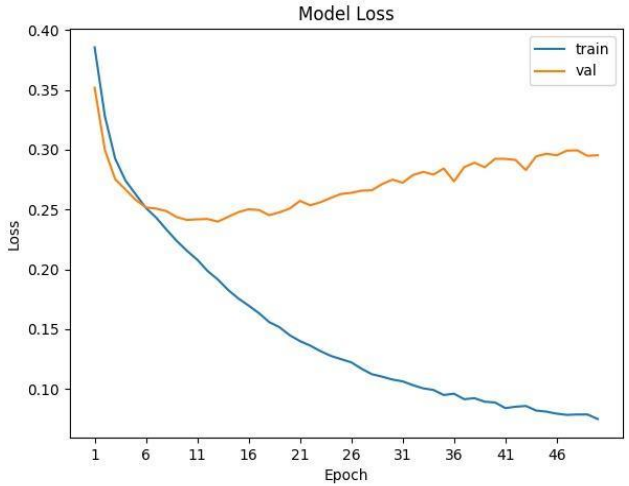
WHAT ARE WE PREDICTING ?

- We are trying to predict the emotions of the students real-time using webcam and Jetson Nano. Emotion with maximum score is displayed on the screen.
- This can be used to increase classroom engagement and also to improve teaching effectiveness by providing instant feedback on students' emotional states.

MODEL ACCURACY AND RESULTS

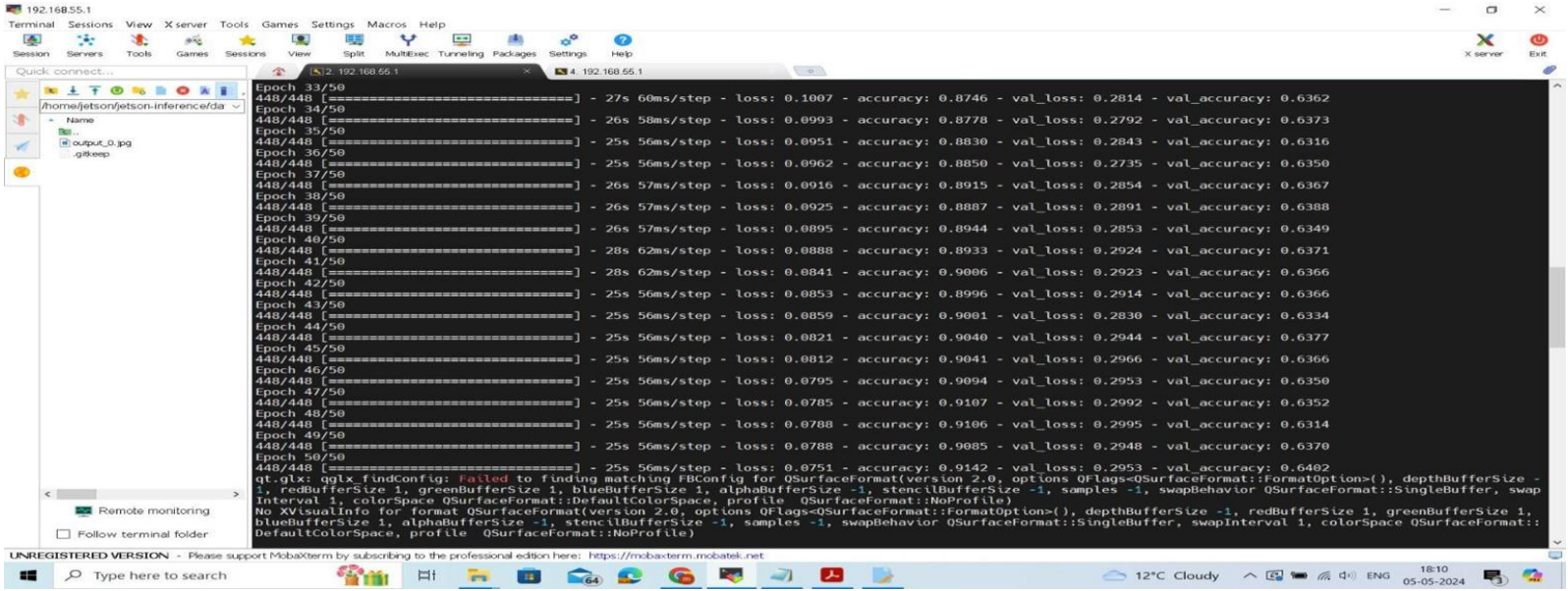
The following plot is the accuracy and loss plot created after training the model on the dataset





MODEL ACCURACY AND RESULTS (Contd.)

The following picture is the output we have received after we trained the model



WHY DID WE CHOSE THIS MODEL?

- They excel at learning hierarchical representations of visual data, making them ideal for analyzing facial expressions captured by a webcam.
- The power of CNNs and specifically the Conv2D model, we can achieve robust and real-time emotion detection from webcam feeds.
- While other models like FCNs, RNNs, and SVMs have their strengths, Conv2D emerged as the most suitable choice for our real-time emotion detection application, balancing performance, efficiency, and ease of implementation.
- CNNs reduce the number of parameters through shared weights and local connectivity, making them especially efficient at recognizing patterns directly from pixels

CHALLENGES

OpenCV Compatibility Issue:

We faced errors with OpenCV, impacting project progress.

Resorted to an older, more stable version of OpenCV to mitigate errors and ensure smooth execution.

Optimizer Compatibility Error:

Encountered errors with the Adam optimizer during model training. Switched to the legacy optimizer.

Training Time for Large Dataset:

Additional time and resources to accommodate the large dataset

Adjustment in Model Training Method:

Faced challenges with the model.fit_generator function. Adapted the training process by utilizing model.fit instead, streamlining model training and optimizing resource utilization.

COST OF PROJECT

- High-Level Costs
  - ✓ Time: Training the model and experimenting with different architectures may require significant time.
  - ✓ Computational Resources: Training deep learning models may require access to powerful GPUs or cloud computing services.
  - ✓ Data Collection: Acquiring and preparing datasets may involve costs depending on the source.
- Considerations: Highlight the need to allocate resources effectively to achieve desired results.

CONCLUSION

- Our project successfully demonstrated the capability of convolutional neural networks to detect and classify emotions in real-time, enhancing classroom engagement and teaching effectiveness.
- By using the FER-2013 dataset, we trained a robust model that can accurately interpret various facial expressions through a webcam setup.
- Despite facing technical challenges, the adaptability and effectiveness of our model suggest significant potential for future applications in educational settings and beyond.
- Future enhancements will focus on optimizing model performance and integrating this technology into a comprehensive classroom management system.

FUTURE WORK

- By placing the webcam and Jetson Nano in the classroom, we can monitor the emotions of the classroom students through a web dashboard.
- For this, we need to first capture the emotions of the students and then store into a database. From the dashboard, we can know how many students have attended the class and the vibe of the classroom.

PROJECT DEMO

REFERENCES

- **For data -** https://www.kaggle.com/datasets/deadskull7/fer2013
- **Code Reference -** https://github.com/atulapra/Emotion detection/tree/master
- **CNN defintion -** https://www.geeksforgeeks.org/introduction-convolution-neural-network/