**Face Expression Recognition**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**(E23CSEU1750) Soham Sharma**

**(E23CSEU1752) Mikhil Yadav**

**(E23CSEU1755) Fuzail Ahmad Khan**

Submitted to

**DR. ASHIMA YADAV (Add your Lab Faculty Name)**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Content | Page No |
| 1 | Abstract |  |
| 2 | Existing/Related Work |  |
| 3 | Methodology |  |
| 4 | Hardware/Software Requirements |  |
| 5 | Experimental Results |  |
| 6 | Conclusions and Future Scope |  |

1. **Abstract**
2. **Related Work (If Any)**
3. **Methodology**
4. **Hardware/Software Required**
5. **Experimental Results**
6. **Conclusions**
7. **Future Scope**
8. **GitHub Link of Your Complete Project**

<Create a Github account and add your code, dataset and readme file,PPT>

<Past the link here>

Abstract

With this project we aim to train a model to categorize various expressions in order to identify human emotions from facial photos. The project requires a substantial knowledge of python libraries like matplotlib, opencv, numpy, tenserflow, pandas and various other libraries for model training.

This project has some far-reaching applications in human and computer interactions:

Adaptive Interfaces: By recognizing facial expressions, user interfaces can become more responsive. For example, depending on the user's present emotional state, gadgets can change the screen's brightness, offer feedback, or even recommend actions.

Virtual Assistants: AI-driven assistants such as Alexa or Siri may be able to identify user emotions to enhance communication by providing sympathetic reactions to user annoyance or enthusiasm.

Emotional Support Systems: Facial expression recognition can help users who struggle with communication detect emotions and fill in knowledge gaps.

Existing work related to the topic

Dataset:

<https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer?resource=download>

Existing Work:

<https://www.kaggle.com/code/shivambhardwaj0101/emotion-detection-fer-2013?select=test>

<https://www.kaggle.com/code/shaikhabdulrafay03/fer-2013>

<https://www.kaggle.com/code/vkoriukina/fer2018-emotion-recognition-pytorch-resnet-18>

<https://www.kaggle.com/code/matzewolf/fer2013-preprocessing>

Novelty:

The way our project differs from the existing ones is the data pre-processing. We have employed opencv for the data preprocessing step. This allows us for more detailed and intricate usage of the available data. We also use pre trained Haar cascades, highly proficient at identifying facial features, eyes, and smiles in pictures. This also impacts the model performance thus making it unique and distinct from the existing work.

Methodology

The methodology employed here remains the same with many working ml models. We started with the first and foremost data preprocessing

Data preprocessing

1. Extraction of Data:  
   Objective: Extract images from compressed files to the local directory for convenient access.  
   Importance: Arranges the data into the proper folders in order to prepare it for processing and model training.
2. Face Recognition:  
   Haar Cascade Classifier: To identify faces in the photos, we employed OpenCV's pre-trained Haar cascade classifier.  
   Relevance: By ensuring that the model concentrates on facial expressions free from background noise, cropping to face areas improves recognition accuracy.
3. Resizing:  
   Procedure: Resize every face that is found to a standard scale (48x48 pixels) or the entire image if no faces are detected.  
   Significance: By standardizing image size, input data becomes consistent, which lowers computational burden and improves model efficiency.
4. Converting to Grayscale:  
   Reasoning: No colour information is required because the dataset photos are grayscale.  
   Importance: it simplifies the data and concentrates the model on the textural and structural aspects of the face that are important for identifying emotions.
5. Normalization  
   Procedure: Divide the pixel values by 255 to scale them to [0,1].  
   Importance: By guaranteeing a steady gradient flow and enhancing convergence, normalization speeds up model training.
6. Enhancement of Data:  
   Techniques for Augmentation: Used flips, zooms, brightness tweaks, and random rotations.  
   Importance: By increasing the dataset's diversity, augmentation improves the model's ability to generalize and lessens overfitting.

It is crucial to visualize the data as it provides more insight about our data and the preprocessing carried out.

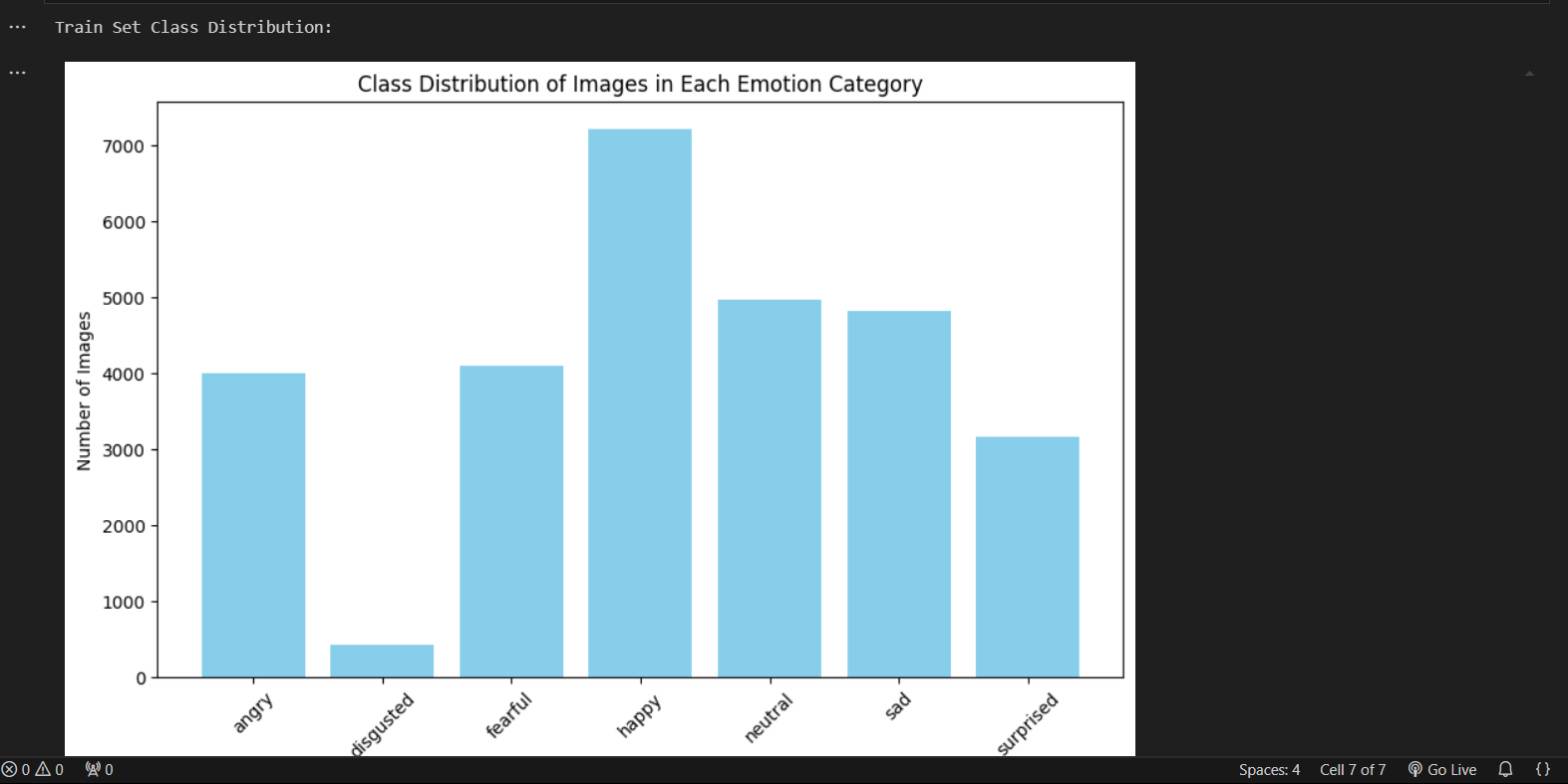
Data Visualization

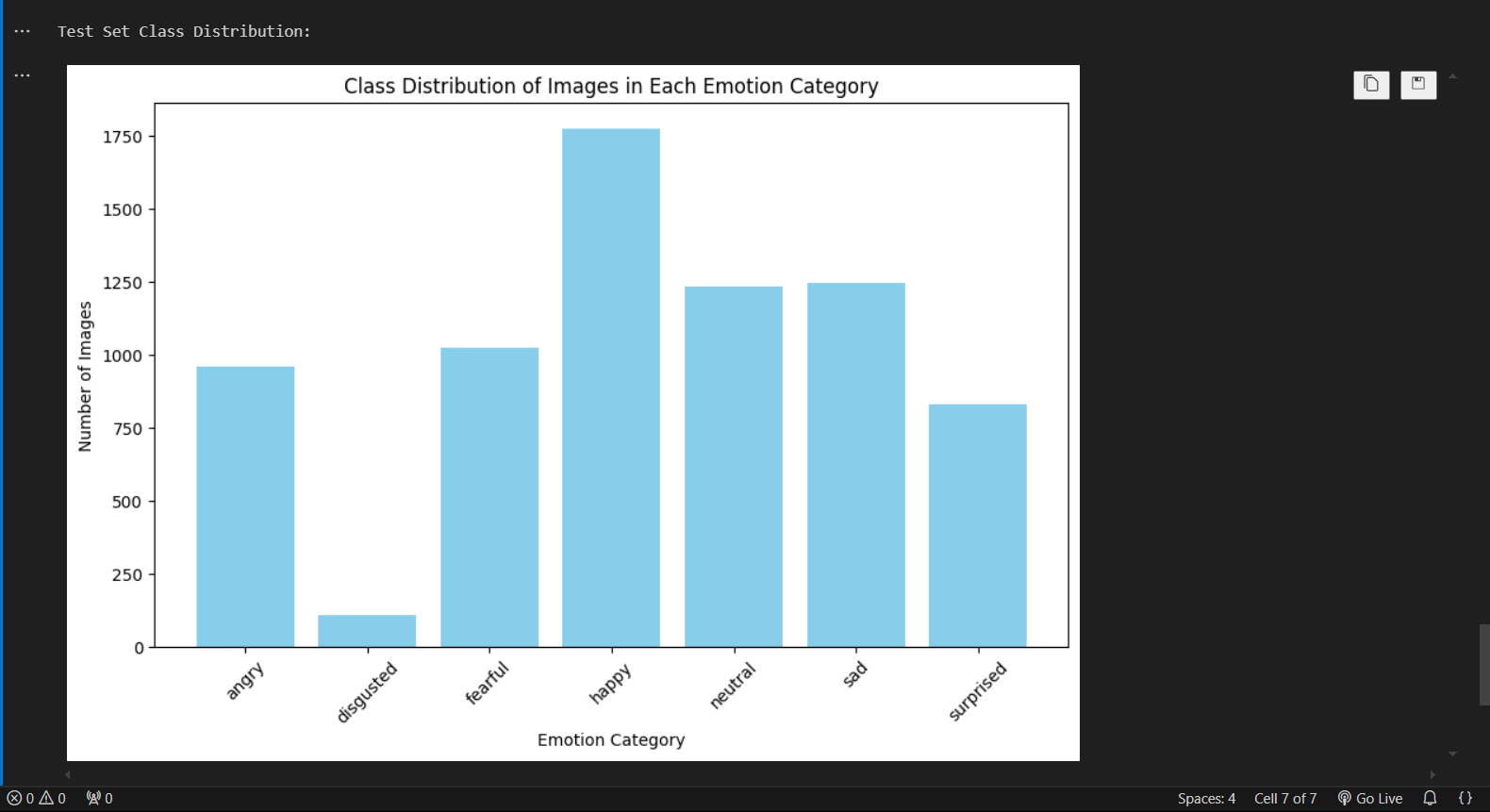
1. A representation of image display  
   Process: Display a few examples of processed photos for each area of emotion.  
   Value: Assists in visually confirming preprocessing operations (facial cropping, resizing, and grayscale conversion), guaranteeing the pipeline functions as intended. This helps in visually verifying the preprocessing steps.



This is what the train test looks with the augmentations carried out.

1. Visualization of Class Distribution:  
   Bar Chart: To display the number of photos in each emotion category for both the train and test sets, a bar chart was made.  
   The dataset balance among emotion classes is revealed in this step, which is important for determining whether class balancing strategies are necessary.





After the data preprocessing, augmentation and visualization all carried out we start with the development of our model. The models we have considered are the random forest classifier, SVM and a deep learning model CNN to provide an in depth analysis of our project.

As the standard ml models cannot work with raw image data we must perform suitable feature extraction so that the data can be interpreted by the ml model

Feature extraction

Local Binary Patterns (LBP): LBP is a texture descriptor that captures local patterns in an image. It is particularly useful for capturing fine-grained texture details and is often used in facial recognition tasks.

The advantages of LBP are that it captures local texture information. Also, it is computationally efficient and robust to illumination changes.

Histogram of Oriented Gradients (HOG): HOG captures the shape and structure of objects in an image by focusing on gradient orientation. It is particularly effective for object detection and facial feature recognition.

The advantages of HOG are that it helps to capture the geometric structure of the image and is also robust to small deformations or noise.

Principal Component Analysis (PCA): After extracting LBP or HOG features, the code optionally applies PCA for dimensionality reduction. It reduces the size of the feature vectors while retaining most of the important information (variance).

This is useful for speeding up training and avoiding overfitting in machine learning models.

Random Forest classifier:

Because Random Forest is a potent ensemble learning technique that mixes many decision trees to increase prediction accuracy and decrease overfitting, it can be utilized for Facial Expression Recognition (FER).

By using their capacity to identify intricate patterns and relationships in high-dimensional data, such as features extracted using Local Binary Patterns (LBP) or Histogram of Oriented Gradients (HOG), Random Forests can handle FER, which frequently entails differentiating minute variations in facial features.

Furthermore, Random Forests require little parameter modification, perform well with both continuous and categorical data, and are robust to noise. They are therefore a dependable option for FER, particularly in cases where the dataset is unbalanced or tiny.

But there are some drawbacks to this model:

Without proper adjustments (which isn’t feasible here), such as class weighting or oversampling, Random Forest struggles to handle imbalanced datasets. Like in the dataset given there seems to be some imbalance in the different classes of data.

Random Forest bias feature importance scores towards variables with more levels or higher variance.

Working with high-dimensional image data, Random Forest struggles to separate overlapping classes effectively, resulting in reduced accuracy.

Random Forest does not consider spatial relationships in image data, which is crucial for tasks like FER. Unlike CNNs, it treats features independently.

Therefore the accuracy of the model suffers here.

CNN:

Because Convolutional Neural Networks (CNNs) can automatically extract hierarchical characteristics from visual data, they are perfect for Facial Expression Recognition (FER).

CNNs learn spatial and visual patterns straight from raw images, catching tiny changes in face emotions, in contrast to classic models that rely on created features like LBP or HOG. They are particularly good at identifying local characteristics (such as movements of the mouth or eyes) while preserving spatial relationships throughout an image.

Additionally, CNNs are very resilient to changes in stance, illumination, and noise—all of which are frequent problems in FER datasets. CNNs are an effective option for precisely differentiating between a range of emotional expressions due to their capacity to generalize across intricate, high-dimensional image data.

This being a DL model gives better accuracy, but still suffers some drawbacks like:

Due to the dataset imbalance the cnn performance suffers a bit.

Complicated techniques like dropout, batch normalization, and learning rate scheduling are required to improve model generalization. optimizers like **AdamW** or **RMSprop**, and fine-tune learning rates and batch sizes which are very complicated and require years of experience to perform.

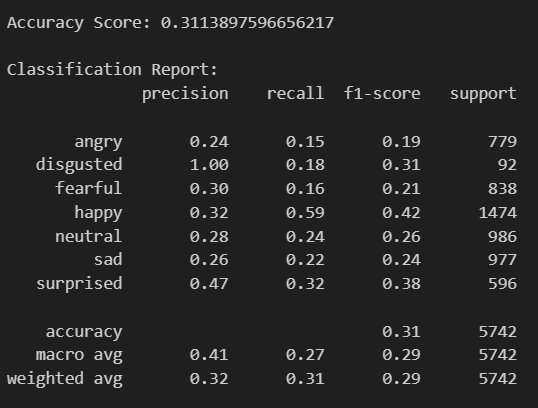
With the current preprocessing and a well-designed CNN, **~55-65% validation accuracy** is achievable.

Hardware/Software requirements

* A multi-core processor like Intel i5/i7 or AMD equivalent is sufficient for small-scale training and inference.
* A dedicated GPU (e.g., NVIDIA GTX 1650, RTX 3060, or better) is highly recommended for faster training of CNNs. A GPU with at least 4-8 GB VRAM is ideal.
* At least 8 GB for small-scale models; 16 GB or more for handling large datasets efficiently.
* Effective cooling to manage heat generated during extended model training.
* TensorFlow or PyTorch for building and training CNNs.
* NumPy, OpenCV, Matplotlib, scikit-learn, and pandas for data preprocessing and analysis.
* CUDA (11.x or later) and cuDNN (corresponding version for TensorFlow/PyTorch) if using an NVIDIA GPU.
* Jupyter Notebook, PyCharm, or Visual Studio Code for coding and experimentation.

Experimental Results:

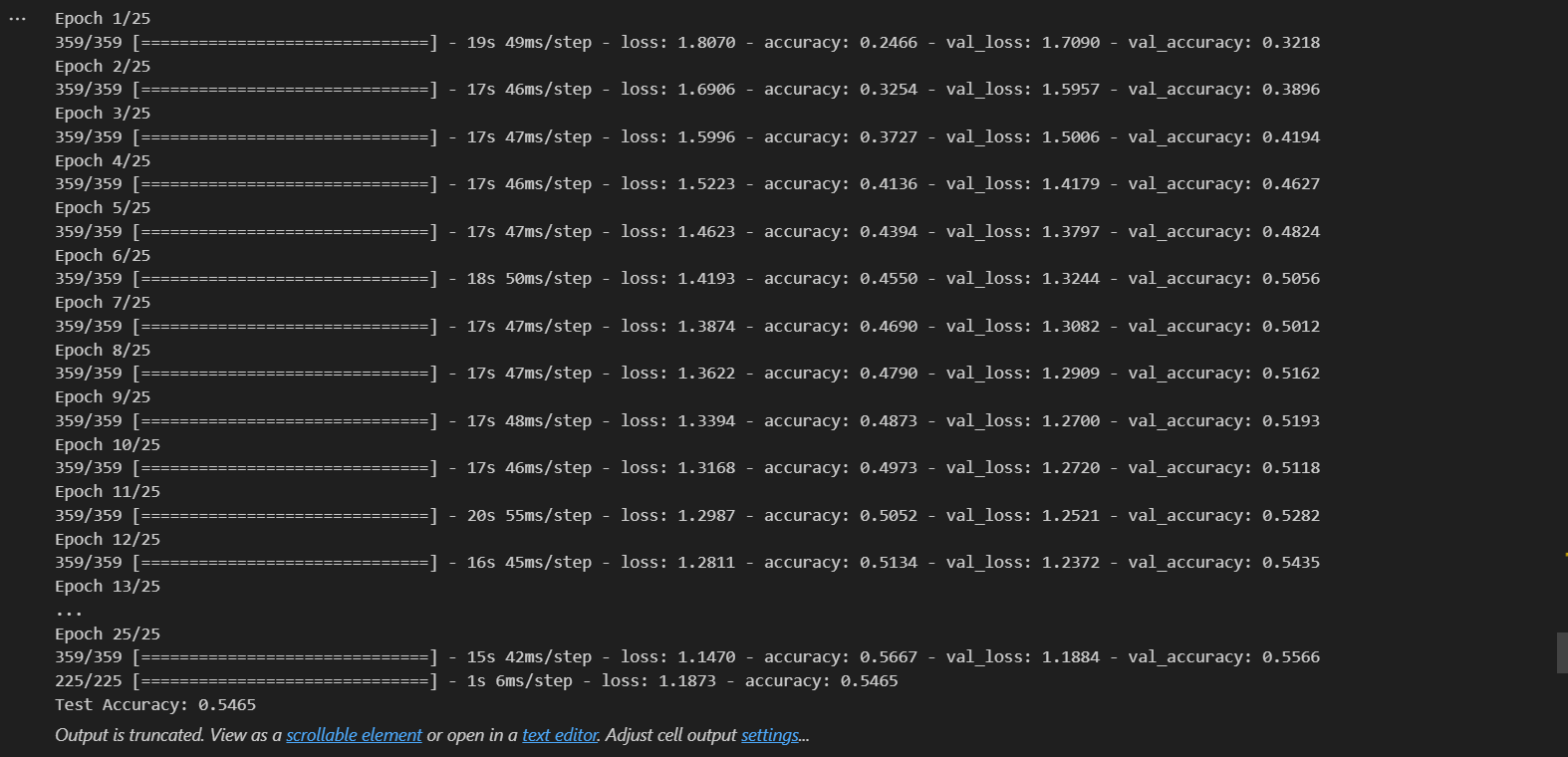
For Random Forest:



A screenshot of a graph

Description automatically generated

For CNN:



A graph of a person and person

Description automatically generated with medium confidence

Conclusion and Future Scope

Finally, this Facial Expression Recognition (FER) study shows how to effectively identify human emotions from facial photos using machine learning and deep learning approaches. The model outperforms conventional techniques like Random Forest in handling picture data by utilizing Convolutional Neural Networks (CNNs) and sophisticated preprocessing techniques to capture complex facial traits.

Model robustness is further increased by data augmentation and normalization, which boosts performance on actual data. Although it is difficult to achieve high accuracy because of differences in illumination, occlusions, and face diversity, the system has potential uses in surveillance, healthcare, and human-computer interaction.

For wider applicability, future improvements might concentrate on improving the model, incorporating real-time processing, and investigating multimodal emotion recognition.

In future the dataset can be improved by:

**Dataset Expansion:** Including a larger and more diverse set of images with varying lighting, backgrounds, and facial angles.

**Class Balancing:** Ensuring that all emotion classes are equally represented to avoid bias in the model.

**High-Resolution Images:** Incorporating higher-quality images to capture finer facial details.

**Metadata Addition:** Adding contextual information like age, gender, or ethnicity for more nuanced analysis.

Experimenting with state-of-the-art architectures like ResNet, EfficientNet, or Vision Transformers to capture deeper features can be extremely beneficial for The model and the accuracy with which we can predict different emotions.

GitHub Link

https://github.com/Soham-Sharma24/SML\_Project.git