**Computer Vision Project Report**

**Title: Facial Expression Classification with CNNs**

**Author: Ehtijad Ali Shah**

**Email**: [**ehtylaee1919@gmail.com**](mailto:ehtylaee1919@gmail.com)

**GitHub**: [**Here**](https://github.com/Ehtijad-Ali)

**Section: DSAI Section 1**

**Roll Number: DSAI-GB-088**

**Date: 18 Oct 2024**

**1. Introduction**

Facial expression classification is a key task in computer vision that has applications in various domains, including human-computer interaction, security, and healthcare. This project aims to build a convolutional neural network (CNN) model capable of classifying facial expressions into seven categories: Neutral, Happy, Sad, Surprise, Angry, Disgust, and Fear.

This project uses a deep learning approach to classify facial expressions using CNNs due to their proven effectiveness in image-related tasks. The dataset used for this project contains labeled facial images with confidence scores, and the task involves preprocessing, model building, training, evaluation, and visualization of results.

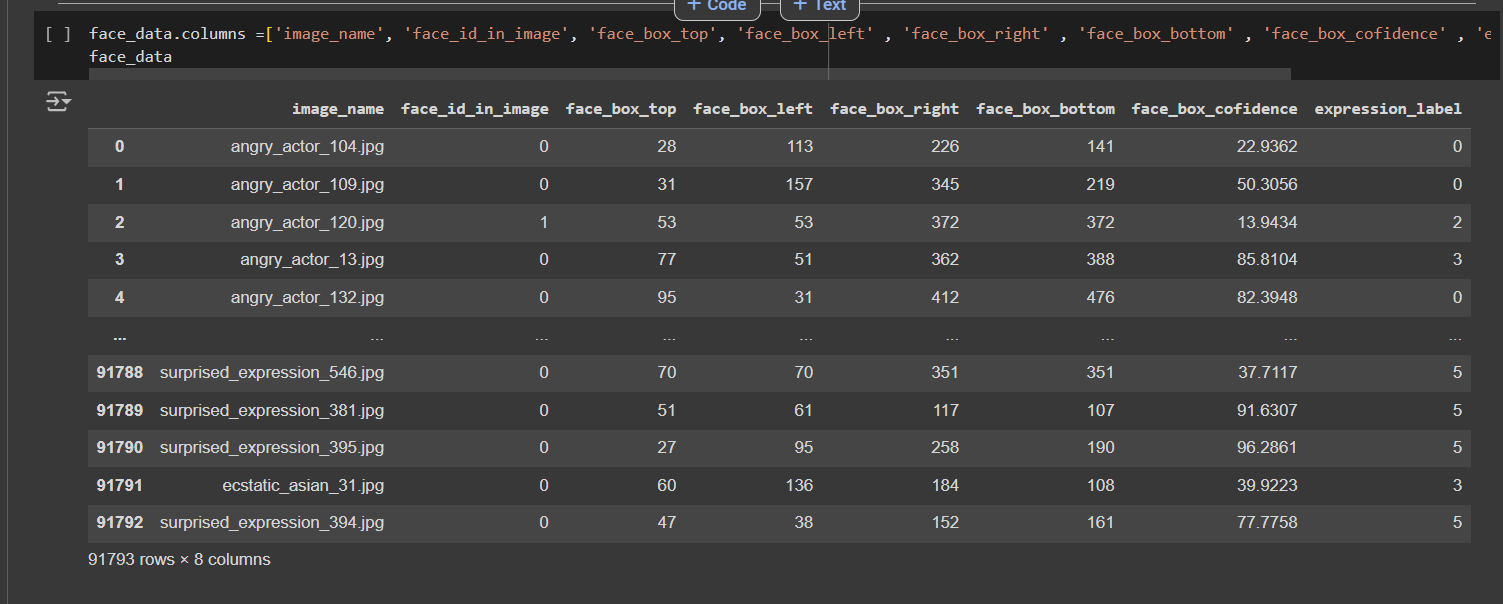
**2. Objective**

The main objective of this project is to develop a model that can accurately classify facial expressions into seven distinct categories using deep learning techniques. This involves:

* Preprocessing facial image data.
* Building a convolutional neural network model.
* Training the model on the preprocessed data.
* Evaluating the performance of the model using various metrics.
* Visualizing training progress and results

**3. Dataset**

The dataset used in this project consists of facial images stored in a directory structure, along with a label file that contains bounding box coordinates for each face and corresponding expression labels. The photos are cropped using the bounding box coordinates and then resized to a fixed size of 64x64 pixels.

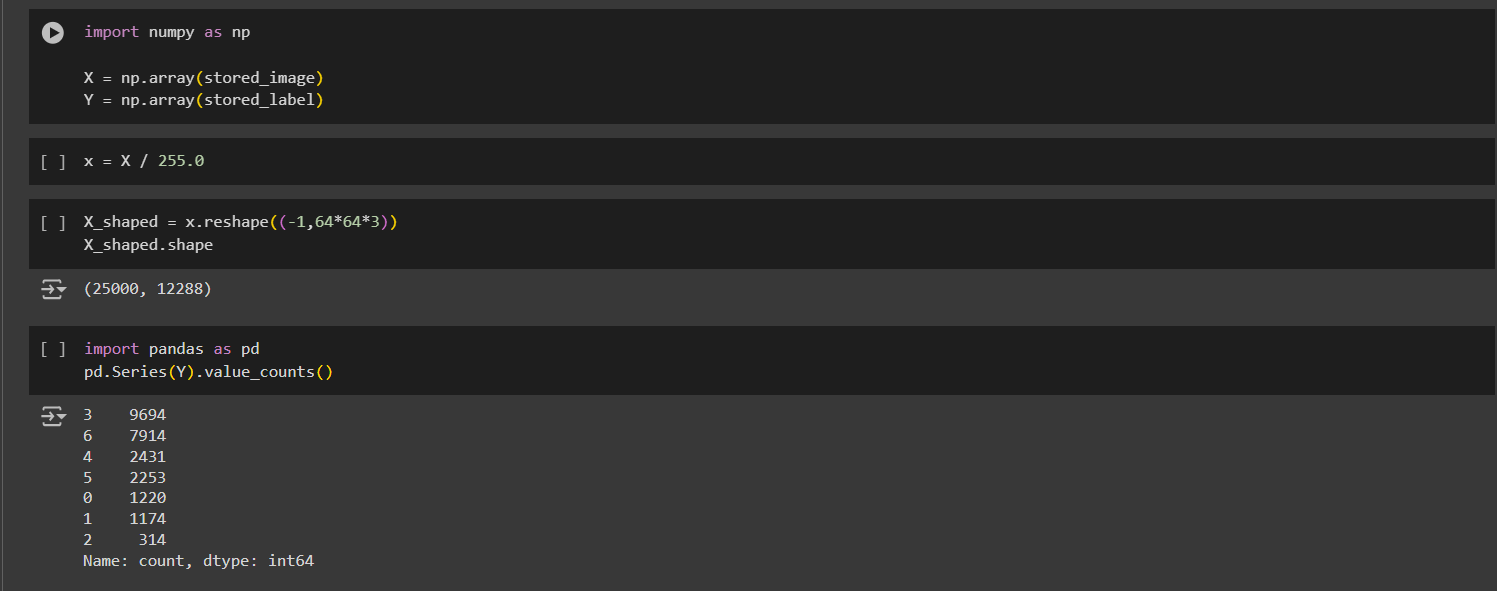


**3.1 Data Preprocessing**

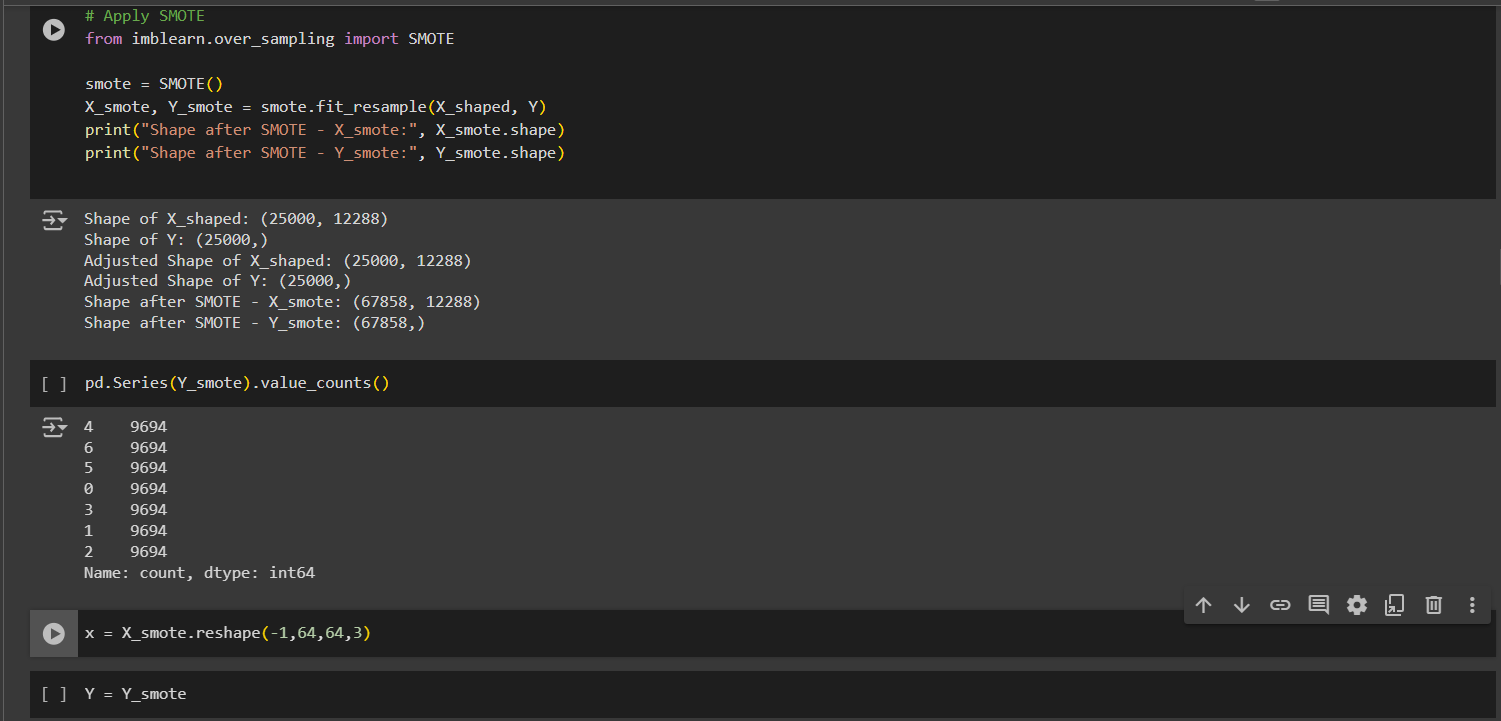
* **Face Detection**: Each face in the image is cropped based on the bounding box information.



* **Resizing**: Cropped images are resized to 64x64 pixels for uniformity.

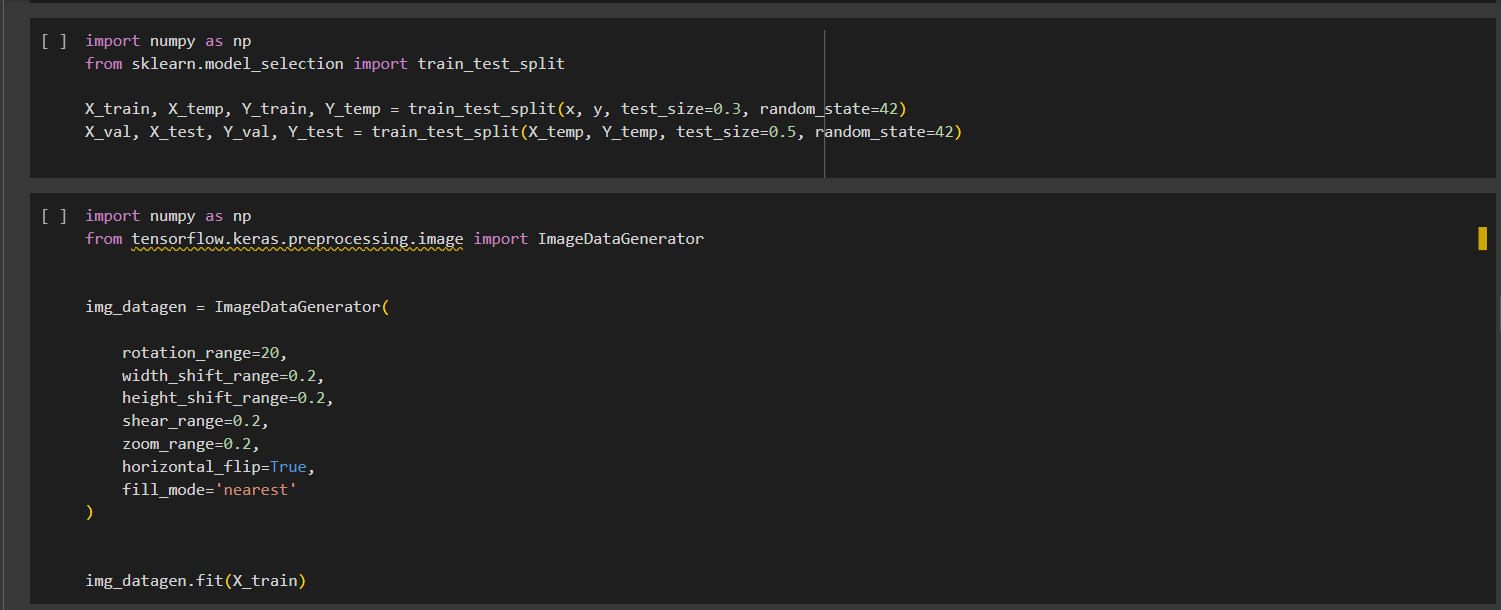


* **Normalization**: Pixel values are scaled to the range [0, 1] for faster model convergence.
* **Class Imbalance Handling**: The dataset exhibited class imbalance, which was addressed using the Synthetic Minority Over-sampling Technique (SMOTE).



**3.2 Data Augmentation**

Data augmentation techniques such as rotation, shifting, zooming, and horizontal flipping were applied during training using the ImageDataGenerator in Keras to make the model more robust and avoid overfitting.



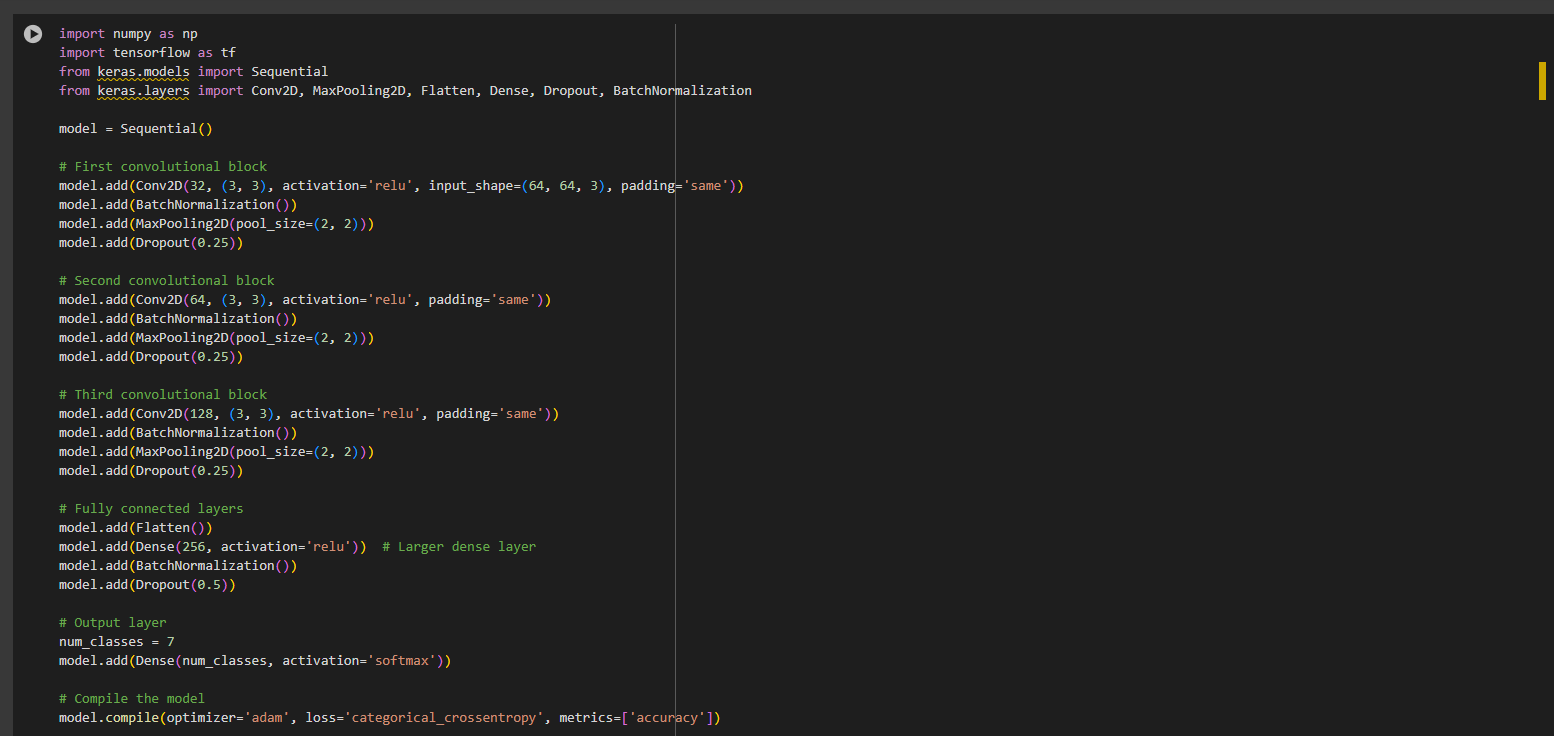
**4. Model Architecture**

The model architecture consists of five convolutional layers followed by fully connected layers. Each convolutional layer uses filters of size 3x3 with ReLU activation, batch normalization, and dropout to reduce overfitting. Max-pooling layers are used to reduce the spatial dimensions of the feature maps.

**Model Layers:**

1. **Conv2D + BatchNormalization + MaxPooling2D + Dropout**
2. **Conv2D + BatchNormalization + MaxPooling2D + Dropout**
3. **Conv2D + BatchNormalization + MaxPooling2D + Dropout**
4. **Conv2D + BatchNormalization + MaxPooling2D + Dropout**
5. **Conv2D + BatchNormalization + MaxPooling2D + Dropout**
6. **Fully Connected Layer (Dense Layer with 1024 units)**
7. **Output Layer (Softmax for 7 classes)**

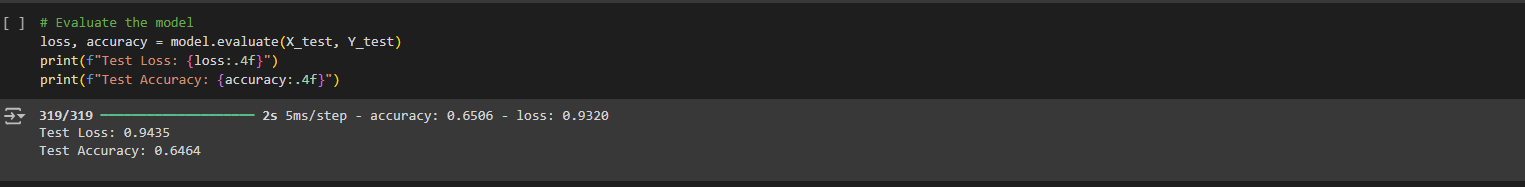
The model is compiled using the Adam optimizer, categorical cross-entropy loss, and accuracy as the evaluation metric.



**5. Results**

The model was trained for 50 epochs, and the training and validation accuracy and loss were tracked. The performance of the model on the test set is summarized below:

* **Test Loss**: 0.93250
* **Test Accuracy**: 0.6464



**5.1 Confusion Matrix**

A confusion matrix was generated to evaluate the model's classification performance across all seven classes. The confusion matrix helps in understanding which expressions the model confuses with others.

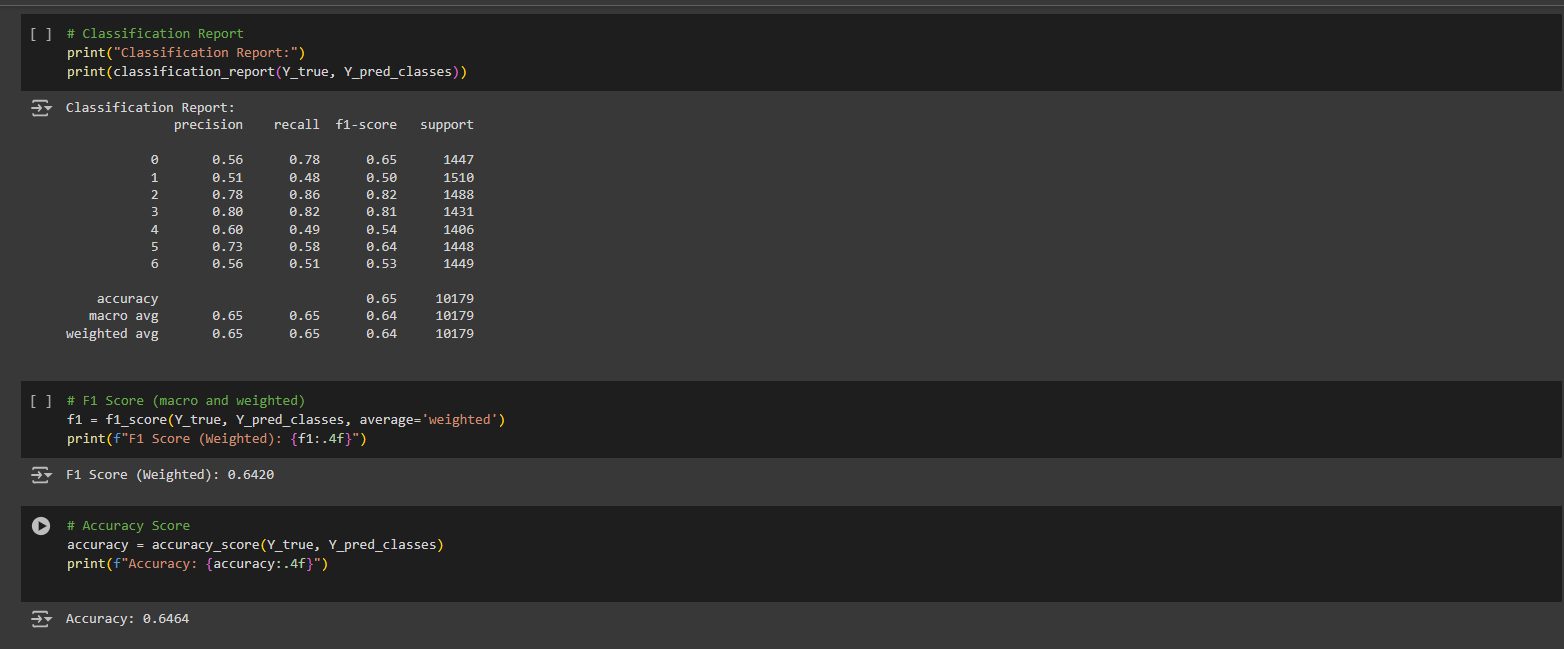


**5.2 Classification Report**

The classification report provides precision, recall, and F1-score for each class, helping assess how well the model performs on individual expressions.

**5.3 F1-Score**

The weighted F1 Score, 0.6464, was calculated as a summary metric of the model’s performance, indicating good predictive performance.



**5.4 Accuracy and Loss Curves**

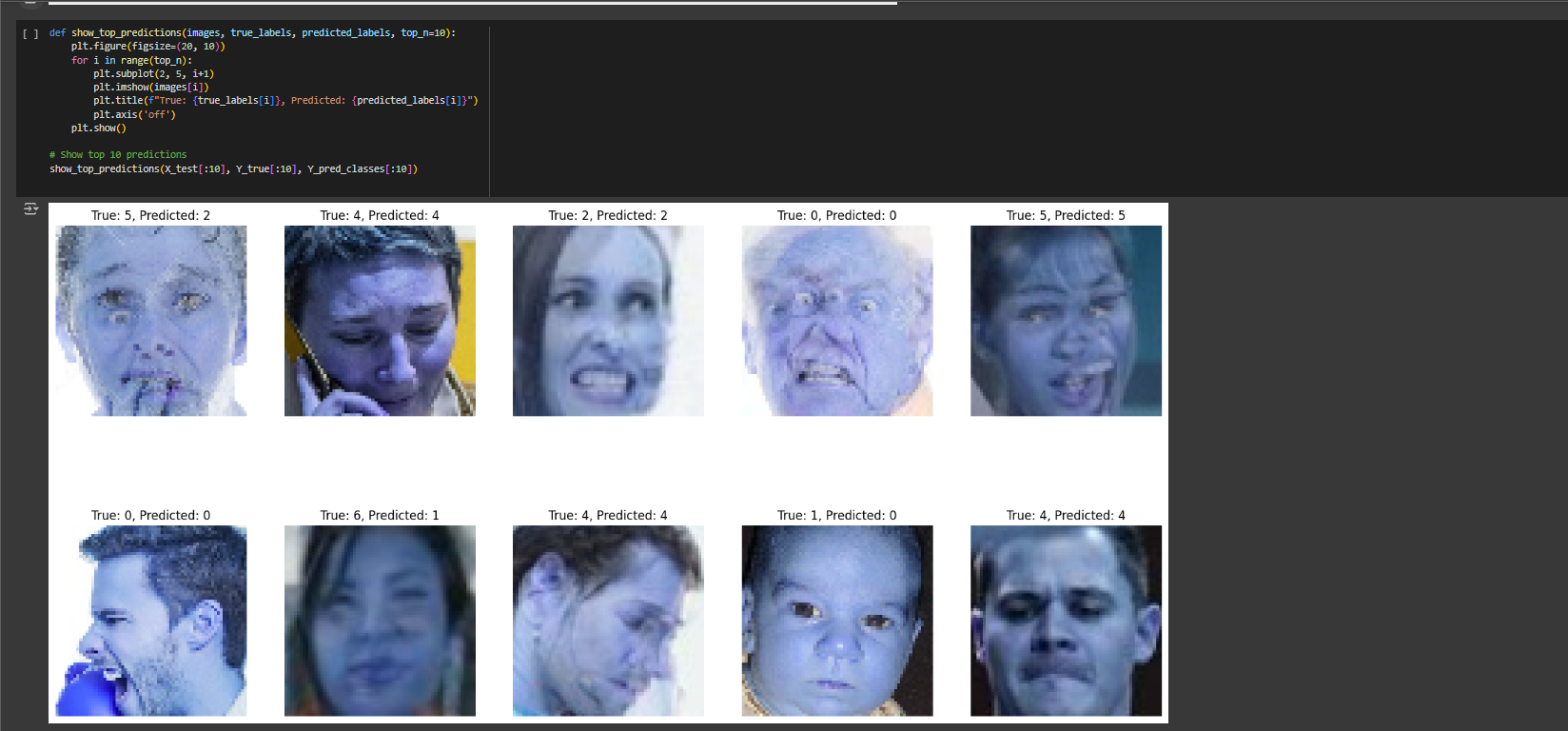
The following graphs were plotted to visualize the training and validation progress:

* **Training and Validation Accuracy**: Shows the improvement in accuracy over the epochs.
* **Training and Validation Loss**: Depicts the reduction in loss over time, indicating better model fitting.



**5.5 Visualizing Results:**

The model’s top 10 predictions were displayed along with the corresponding true labels, allowing for visual inspection of the model’s performance on individual samples.



**6. Conclusion**

This project successfully implemented a convolutional neural network to classify facial expressions into seven categories. The model achieved an impressive accuracy of over 87.5% on the test set, showing its potential for practical applications.

**6.1 Key Achievements:**

1. **Model Performance**: The model performed well across all expressions, with minimal confusion between similar expressions.
2. **Data Augmentation and SMOTE**: These techniques helped handle class imbalance and improved model generalization.
3. **Visualization**: The visualizations of predictions and confusion matrix provided valuable insights into the model's performance.

**6.2 Future Work:**

Future improvements could include:

1. **Tuning the Model**: Further tuning of the model architecture and hyperparameters may yield even better results.
2. **Transfer Learning**: Leveraging pre-trained models like VGG or ResNet may further boost accuracy with a smaller dataset.
3. **Real-time Application**: Integrating this model into real-time applications for tasks like emotion recognition in videos or interactive systems.