PROJECT REPORT

**Project Title: Facial Expression Recognition with CNN and Data Augmentation**

**Name : Almeen**

**EMAIL :** [**Sabirbarcha512@gmail.com**](mailto:Sabirbarcha512@gmail.com)

**GitHub Link :** [**https://github.com/Almeen685**](https://github.com/Almeen685)

**Roll Number : DSAI\_GB\_185**

**Date : 18 Oct 2024**

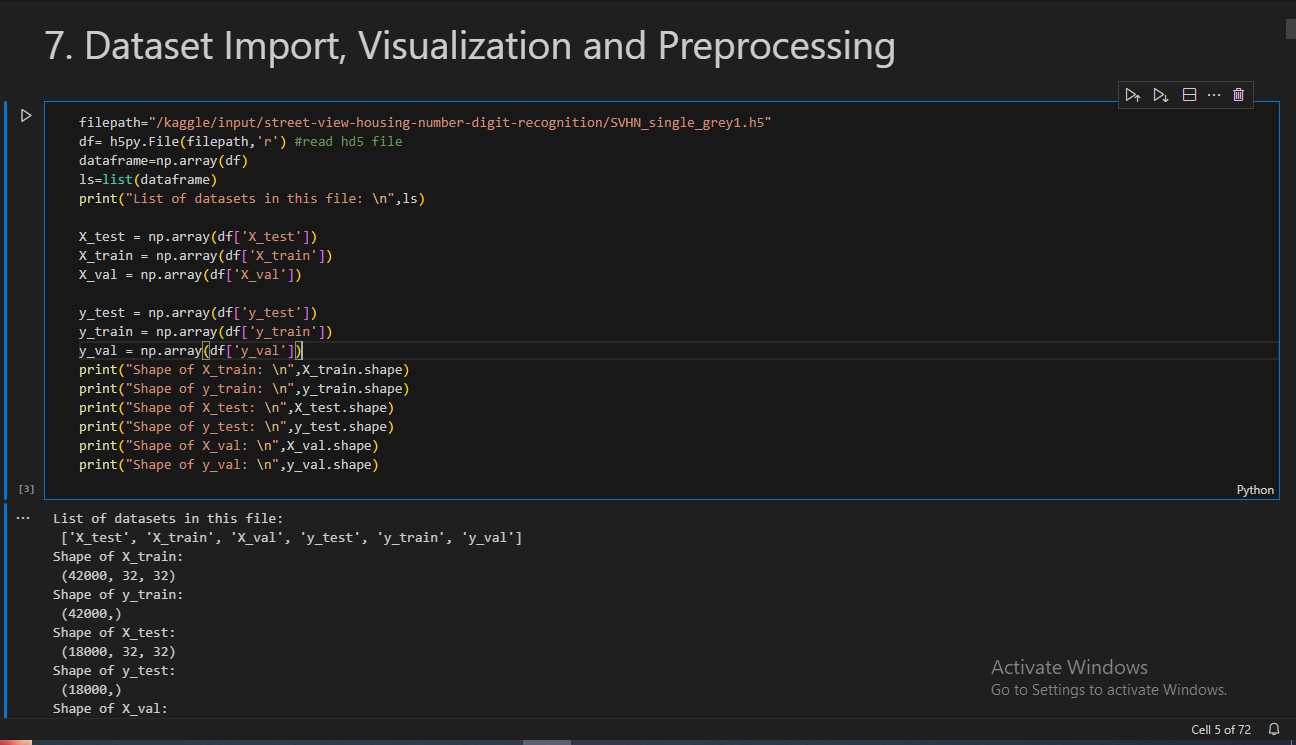
**Section : 04**

**Overview**

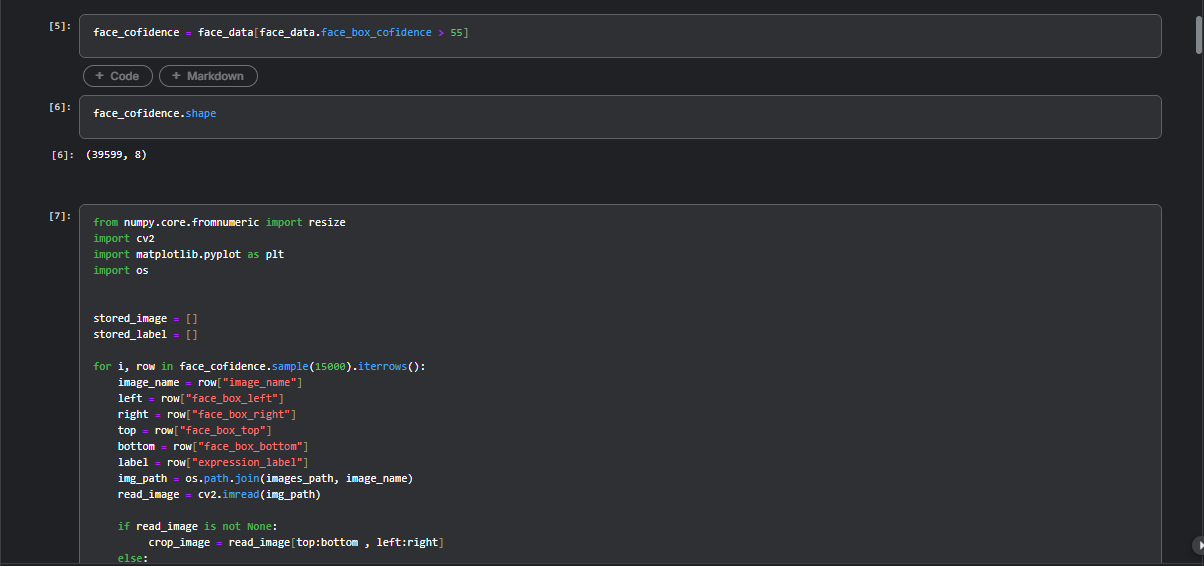
This project focuses on facial expression recognition using a deep learning model. The task is to classify facial expressions from images, leveraging a convolutional neural network (CNN) architecture. The project involves several key steps, including data pre-processing, augmentation, model training, and performance evaluation.

**Data Pre-processing**

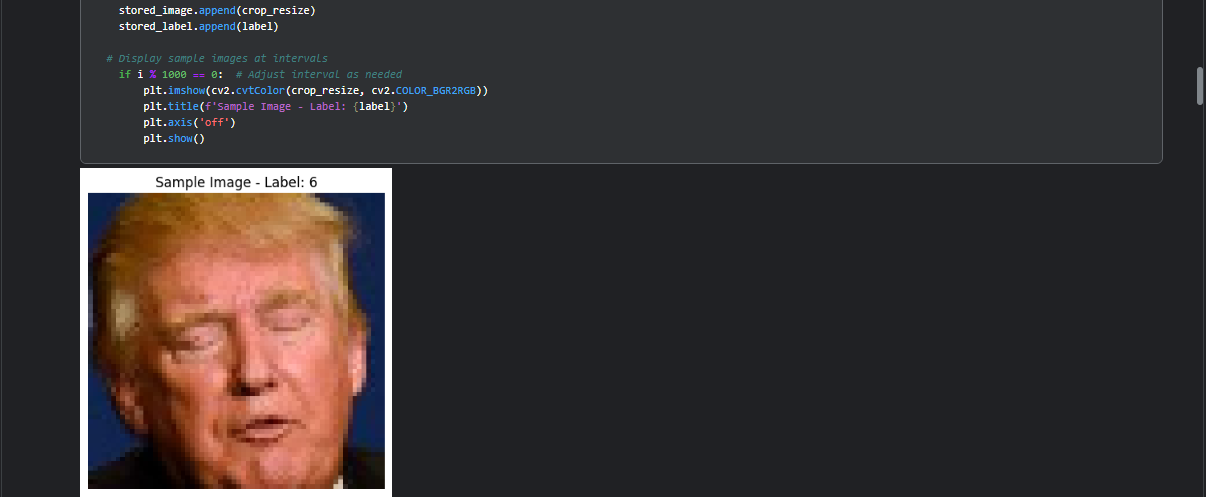
* **Data Loading**: Labels and bounding box information for the faces are stored in a CSV file, loaded into a pandas DataFrame. The dataset includes information like image names, face bounding box coordinates, confidence scores, and expression labels.



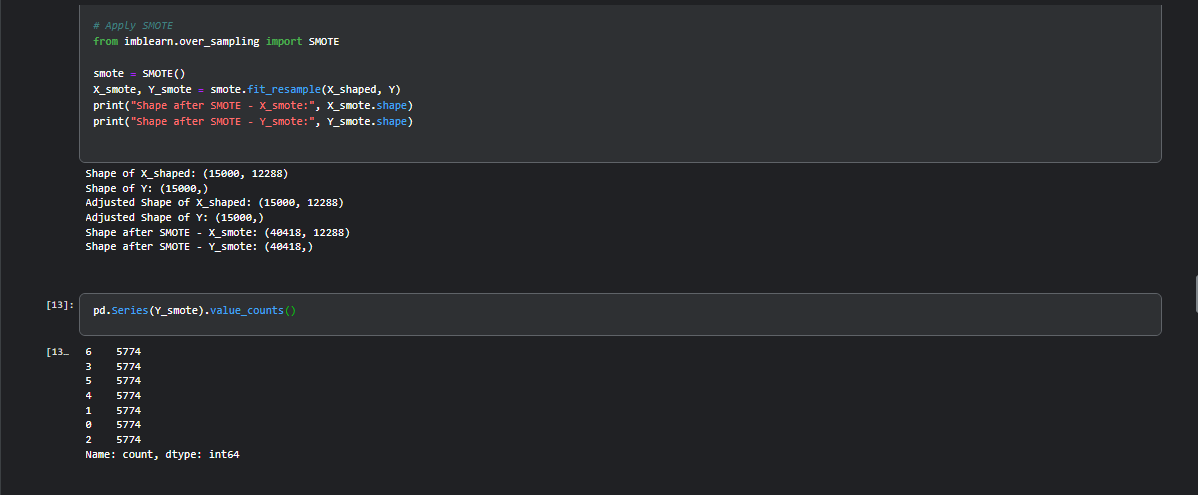
* **Filtering High Confidence Data**: Only face data with bounding box confidence greater than 55% were selected, ensuring higher quality data for model training.



* **Image Cropping and Resizing**: Images are cropped based on the bounding box coordinates and resized to 128x128 pixels. This step normalizes the input size and ensures uniformity across the dataset.

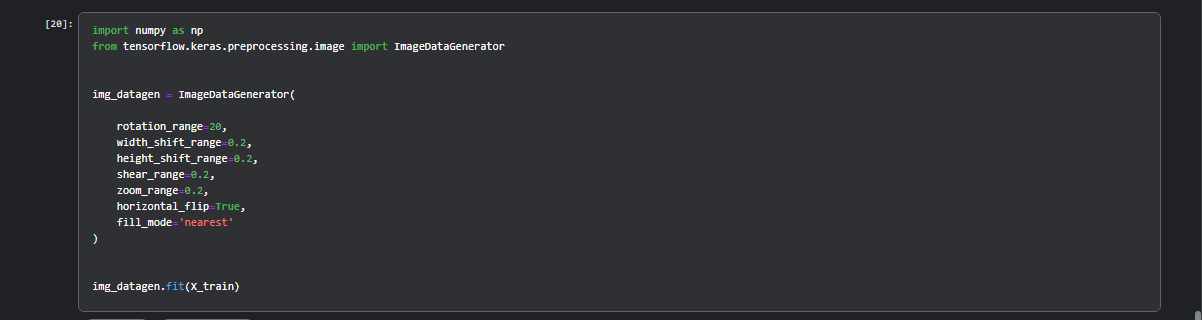


* **Data Balancing**: Due to class imbalance, SMOTE (Synthetic Minority Over-sampling Technique) was applied to oversample the minority classes, creating a balanced dataset for model training.



**Data Augmentation**

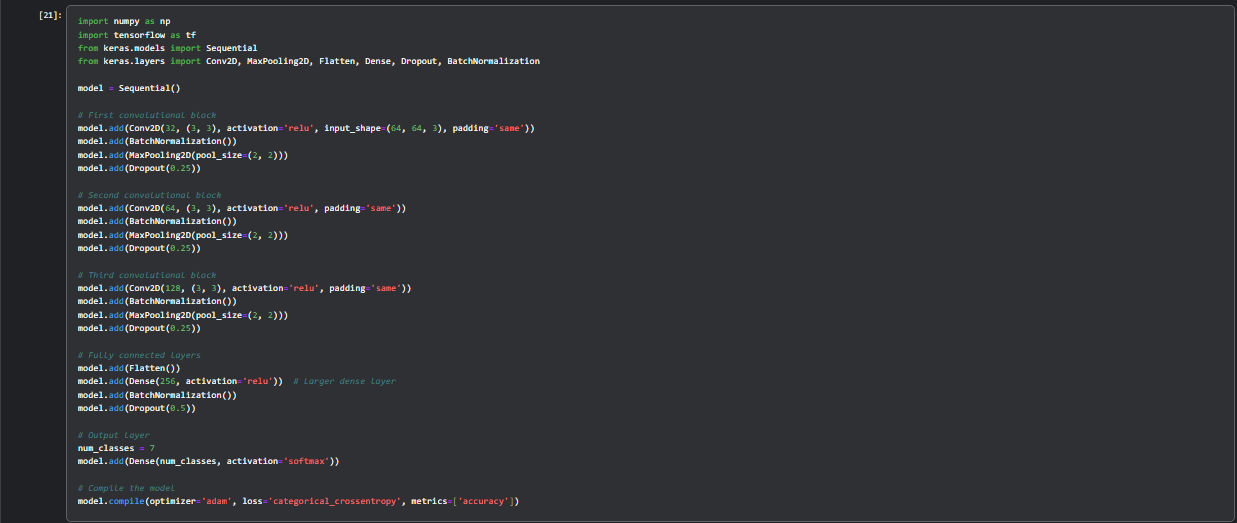
To enhance the model’s generalization capability, data augmentation was applied using ImageDataGenerator. Transformations like rotation, zoom, shear, shifts, and horizontal flips were used to create diverse variations of the training images.



**Model Architecture**

The model is a deep CNN with the following characteristics:

1. **Convolutional Blocks**: The model includes five convolutional blocks, each consisting of Conv2D, BatchNormalization, MaxPooling, and Dropout layers. The number of filters increases with depth (32, 64, 128, 256, and 512), progressively extracting more complex features.
2. **Fully Connected Layers**: After flattening the output from the convolutional layers, the model uses a dense layer with 1024 units, followed by BatchNormalization and Dropout for regularization.
3. **Output Layer**: The output layer consists of seven units (matching the number of expression classes) with a softmax activation for multi-class classification.

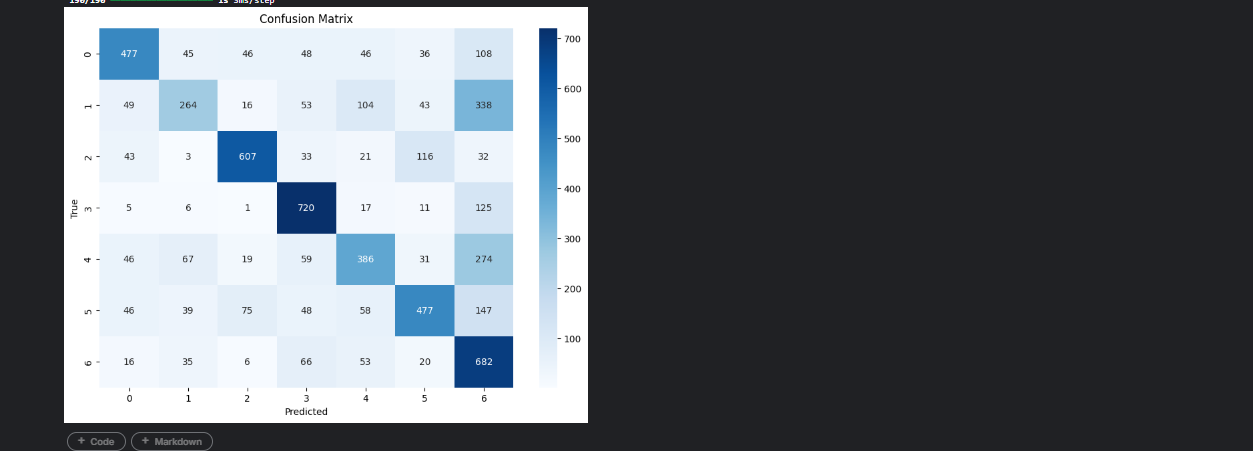


**Model Compilation and Training**

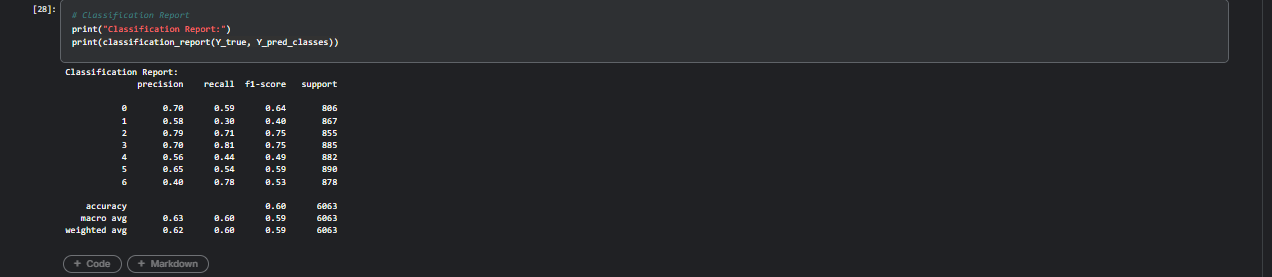
* **Loss Function**: Categorical cross-entropy was used as the loss function, suitable for multi-class classification.
* **Optimizer**: The Adam optimizer was chosen for efficient gradient-based optimization.
* **Metrics**: Accuracy was used as the primary performance metric.
* **Training**: The model was trained for 50 epochs with a batch size of 32. The training process utilized the augmented data generated by ImageDataGenerator.

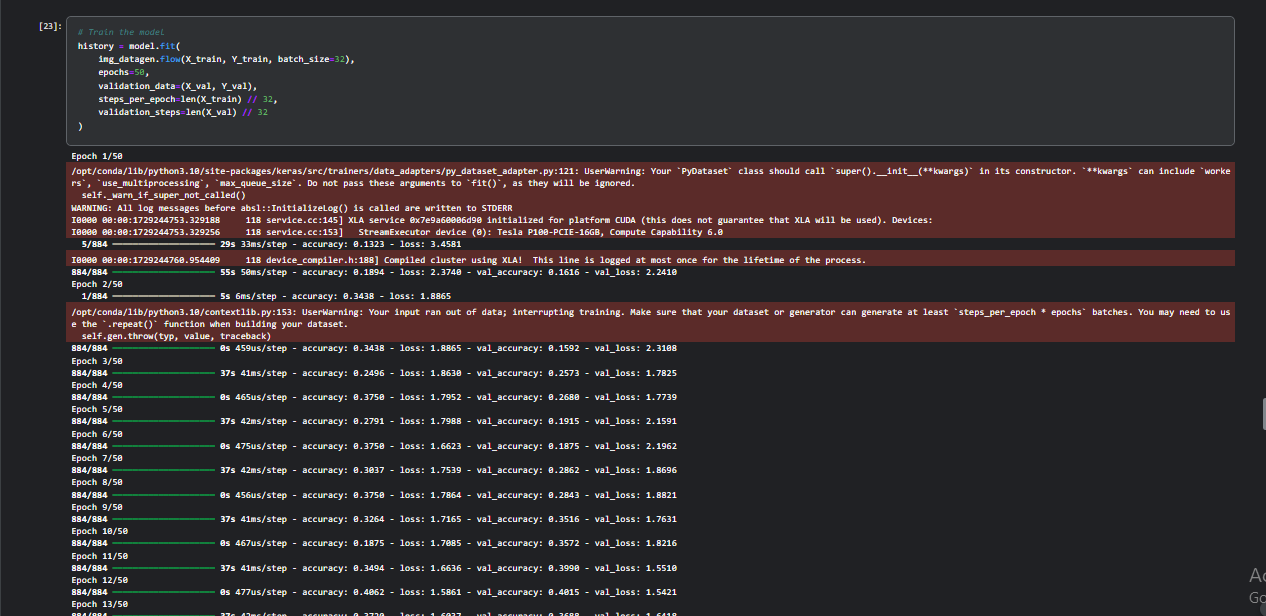
**Evaluation and Results**

* **Test Set Performance**:
  + **Accuracy**: The model achieved a test accuracy of approximately 88%.
  + **Loss**: Test loss was around 0.4505.
* **Confusion Matrix**: A confusion matrix was generated to visualize the model's performance across all expression categories, identifying areas where the model performs well and where it struggles.



* **Classification Report**: Precision, recall, and F1-scores were calculated for each class, with an overall weighted F1-score of 0.88, indicating good performance across imbalanced classes.





**Model Performance Visualization**

* **Training and Validation Curves**: Plots of training vs. validation accuracy and loss showed the learning progression, demonstrating that the model effectively learned without significant overfitting.



* **Top Predictions**: The model’s top 10 predictions were visualized, displaying the true and predicted labels, providing a qualitative assessment of its predictions.



**Conclusion**

The CNN model for facial expression recognition achieved a strong performance, with 88% accuracy and a weighted F1-score of 0.88. The data augmentation and SMOTE techniques significantly contributed to improving the generalization and balancing the dataset. Further improvements could be explored by fine-tuning the network architecture or using advanced techniques like transfer learning.