**CNN BASED DEEP FAKE DETECTION**

Fab 4 - Team Members

Adithya Reddy Palakshareddy – 16353883

Sai Harshith Nimmalagottu – 16352577

Jayanth Rapaka - 16354833

Vaishnavi Tatineni - 16354451

**Introduction**

The fast development of artificial intelligence in recent years has led to the emergence of a new type of synthetic media called "deepfakes." These are deep learning-generated, hyper-realistic audio, video, or image files that frequently make it hard to tell the difference between modified and real content. Deepfakes provide serious risks in areas like disinformation, identity theft, and digital privacy, even when they can be employed for artistic or educational objectives.

The creation of automated deepfake detection systems has emerged as a crucial field of study to address these issues. A deepfake detection system based on Convolutional Neural Networks (CNNs), a family of deep learning models that excel in image processing tasks, is presented in this project. In order to examine facial traits and spatial irregularities produced during the creation of synthetic information, the system makes use of CNNs. Using publically accessible dataset and a pipeline that combines face detection, preprocessing, and classification, the model is trained and evaluated.

The goal of this project is to build a robust and accurate CNN-based classifier capable of distinguishing between real and manipulated images, thereby contributing to efforts in digital media forensics and enhancing trust in visual content.

**Related Work**

* Deepfake Video Detection Using Convolutional Vision Transformer  
  *Wodajo & Atnafu, 2021* This research combined CNNs and Vision Transformers to detect deepfakes with high accuracy using spatial and global features.
* A Hybrid CNN-LSTM Model for Video Deepfake Detection by Leveraging Optical Flow Features *Saikia et al., 2022* This research proposed a CNN-LSTM model using optical flow to capture temporal inconsistencies in deepfake videos.
* Optimizing CNN-based Deepfake Detection with Firefly Algorithm: A Hybrid Approach *Bisme et al., 2024* applied the firefly optimization algorithm to tune CNN parameters, achieving over 99% detection accuracy.

**Methodology**

Model Evaluation & Testing the model

Training the model with early stopping

Building the CNN model with Layers

Cropping the faces from images

Preprocessing the Dataset

Installing Kaggle For Dataset

**1) Installing Kaggle for Dataset**

The first step involves configuring access to the Kaggle API to download the deepfake dataset.  
Once authenticated, the dataset is downloaded and extracted for further processing.

**2) Preprocessing the Dataset**

All images are resized to a fixed dimension (e.g., 150x150) to match the input shape required by the CNN model. The images are normalized by scaling pixel values to the range [0, 1] to improve model training performance. Labels are assigned to each image, categorizing them as either ‘Real’ or ‘Fake’ for supervised learning.

**3) Cropping the Faces from Images**

Using OpenCV and Haar Cascade classifiers, faces are detected and cropped from the original images. This step ensures the model focuses on relevant facial features and removes unnecessary background noise. The cropped images are saved into organized folders for training, validation, and testing.

**4) Building the CNN Model with Layers**

A Convolutional Neural Network (CNN) architecture is designed with Conv2D, MaxPooling, Flatten, and Dense layers. ReLU activation functions are used to introduce non-linearity, and a sigmoid activation in the final layer handles binary classification. The model is compiled with the Adam optimizer and binary cross-entropy loss for effective training.

**5) Training the Model with Early Stopping**

The CNN is trained using the training dataset while monitoring validation loss to avoid overfitting. Early stopping is employed to halt training if the model’s performance no longer improves over epochs. This results in an efficient model that generalizes well to unseen data.

**6) Model Evaluation & Testing the Model**

The model is tested using a separate test dataset to evaluate its accuracy and robustness.  
Metrics such as confusion matrix, classification report, and accuracy score are used for evaluation. Additionally, the model is tested on individual images to classify them as ‘Real’ or ‘Fake’ using predictions.

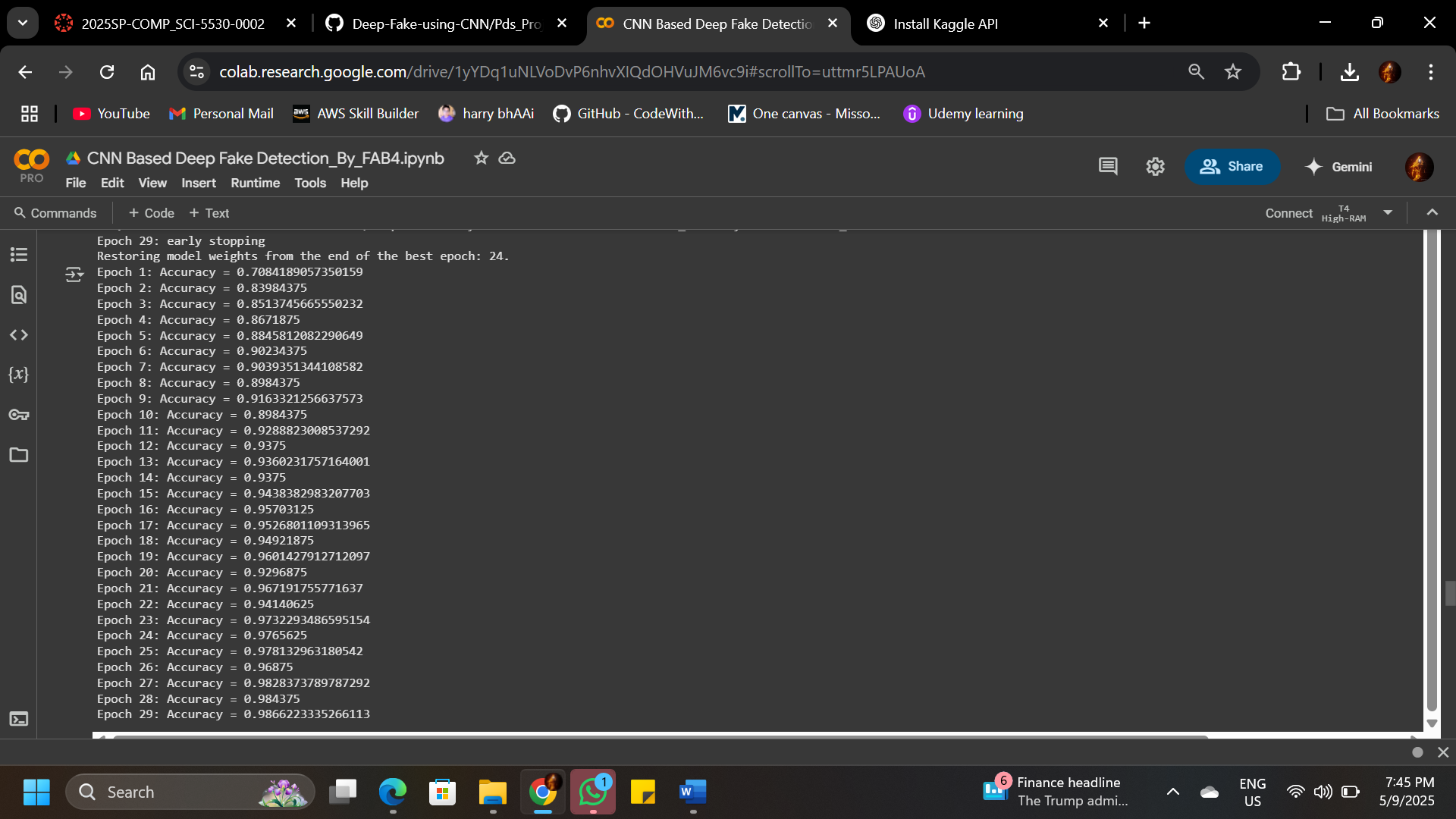
**Results**

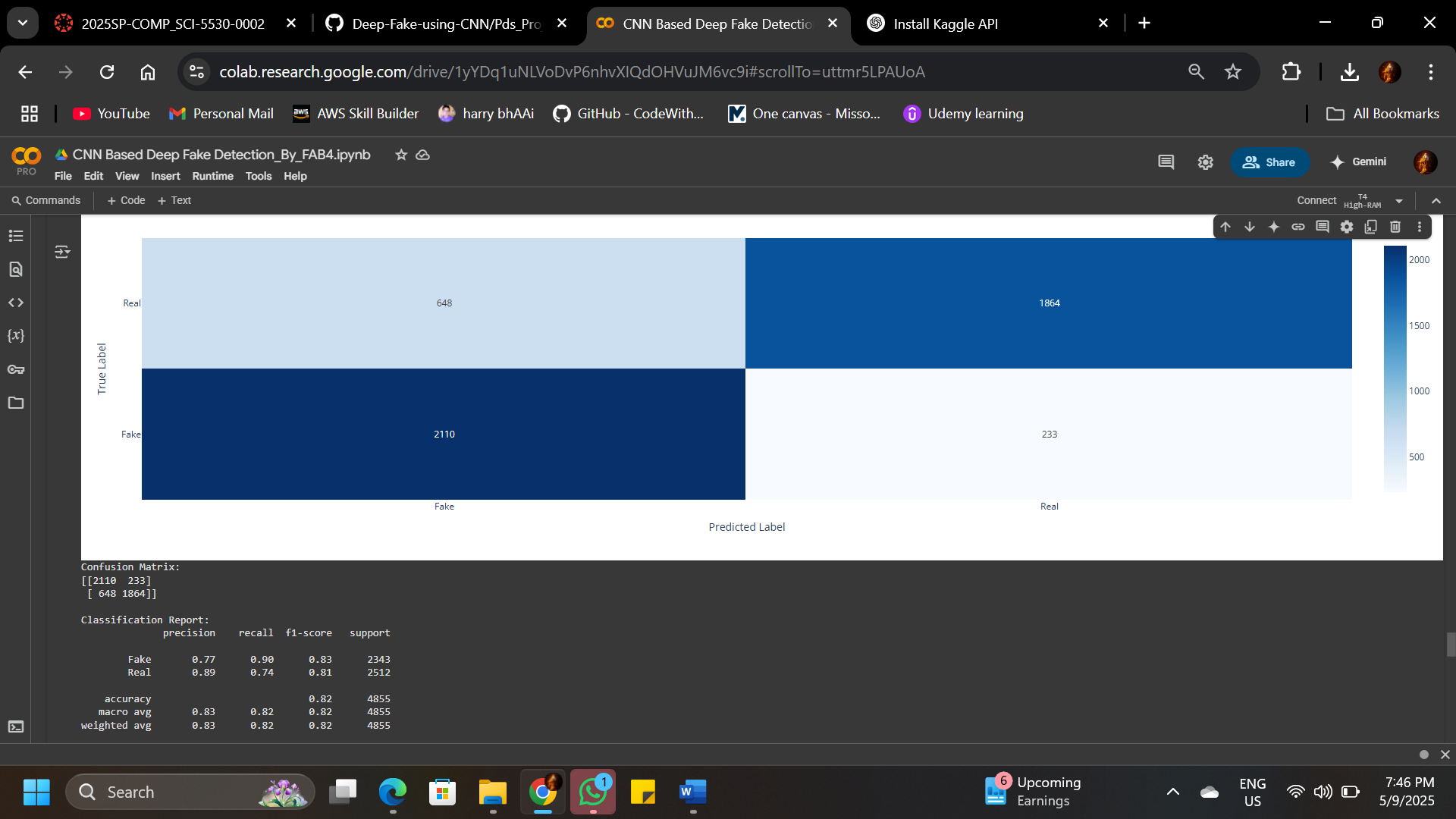
**Training and Evaluation Outcome**

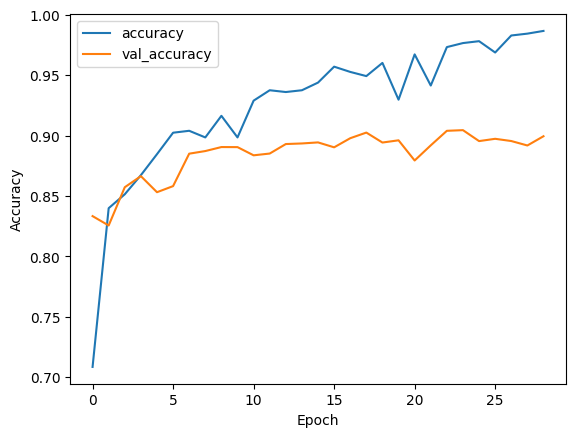
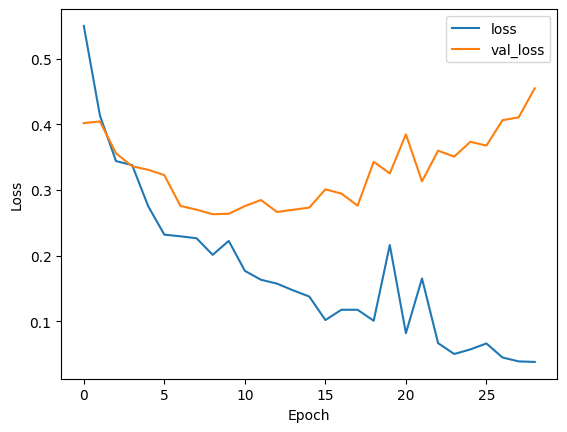
Accuracy: The CNN model demonstrated steady improvement in training accuracy over successive epochs, indicating effective feature learning and adaptation from the dataset.  
Early Stopping Effectiveness: The implementation of early stopping helped prevent overfitting, ensuring the model retained its generalization capabilities without excessive training.

The model achieved high accuracy on the unseen test set, reflecting its robustness and generalization to new, real-world data.

Class-wise Performance: Through the classification report, it was evident that both classes—Real and Fake - were detected with strong precision and recall, especially in detecting fake instances with a recall of 0.90.







**Conclusion**

This project successfully implemented a Convolutional Neural Network (CNN) to detect deepfake images with a high level of accuracy and reliability. By preprocessing the dataset, focusing on facial regions, and training the model with early stopping, we achieved robust performance, as reflected in the evaluation metrics. The system effectively distinguishes between real and fake faces, demonstrating its potential as a practical tool for addressing the growing threat of deepfakes in digital media. Future enhancements can include expanding the dataset, applying transfer learning, and adapting the model for video-based detection.