Audio Deepfake Detection Project Documentation

Overview

This documentation provides a detailed look at the methodology and steps involved in the audio deepfake detection project. The goal is to classify audio as either real or fake by processing a dataset, extracting relevant audio features, and using a pre-trained machine learning model.

Methodology

Dataset Preparation

Traverse Input Directory

The project begins by traversing the input directory to list and count all available files. This initial step is crucial for understanding the structure of the dataset and ensuring that all necessary files are accessible for further processing. It helps in identifying the various types of files and their locations, which is important for subsequent steps in the project.

Count Files

After listing the files, the next step involves counting the directories and files within the dataset. This count helps provide a clear view of data distribution across different folders. Understanding this distribution is essential for efficient dataset management and organization, ensuring that data is correctly categorized for processing.

Load CSV

The core of the dataset is loaded from a CSV file into a pandas DataFrame. This updated dataset includes 29,965 rows, a significant increase from the previous dataset of 11,778 rows. The expansion allows for a more robust training and evaluation process, providing a larger and more varied set of data for the model to learn from.

Data Exploration

Examine DataFrame Shape

Exploration of the dataset begins by examining the shape of the DataFrame to understand the number of samples and features available. Knowing the dimensions of the dataset helps in planning the subsequent data processing steps and ensures that the dataset is suitable for the intended machine learning tasks.

Review Samples

Samples labeled 'FAKE' and 'REAL' are reviewed to inspect the content and distribution. This review ensures that the dataset is representative of the task at hand and that there are no obvious errors or anomalies in the data. It provides a preliminary check on the quality and relevance of the data.

Generate Summary

A summary of the DataFrame is generated to check data types and non-null values. This summary ensures data integrity and confirms that the dataset is ready for processing. It provides an overview of the data's structure, helping to identify any missing or inconsistent values that need to be addressed.

Descriptive Statistics

Descriptive statistics are provided to offer insights into the data distribution and key metrics. These statistics help in understanding the central tendencies, dispersions, and overall distribution of the data, which are important for making informed decisions during data preprocessing and model training.

Data Processing

Encode Labels

To facilitate model training, the labels 'FAKE' and 'REAL' are encoded as 0 and 1, respectively. This encoding is necessary for the machine learning model to interpret the labels correctly. It transforms categorical data into a numerical format that the model can process effectively.

Calculate Correlation Matrix

A correlation matrix is calculated, focusing on the 'LABEL' column to identify features that strongly correlate with the labels. This helps in understanding which features are most relevant for classification and provides insights into potential feature selection and engineering steps.

Feature Extraction

Define Feature Extraction Function

Feature extraction is a critical part of the project. A function is defined to extract various audio features from the files, such as chromagram, spectral centroid, and MFCCs. These features are essential for training the model, capturing different aspects of the audio signals that are relevant for distinguishing between real and fake audio.

Test Feature Extraction

The feature extraction function is tested on sample audio files to ensure accurate extraction. This testing phase verifies that the function works correctly and that the extracted features are meaningful and useful for the classification task.

Features:

- **Chromagram:** Captures the harmonic content and pitch class distribution.
- **Spectral Centroid**: Indicates the "center of mass" of the spectrum, related to brightness.
- **Spectral Bandwidth**: Measures the spread of the spectrum, indicating timbral texture.
- **Spectral Rolloff**: Represents the frequency below which a certain percentage of the total spectral energy is contained.
- **Zero Crossing Rate**: Measures the rate at which the signal changes sign, related to noisiness.
- **RMS (Root Mean Square)**: Represents the signal's loudness or energy level.
- **MFCCs (Mel-Frequency Cepstral Coefficients)**: Capture the timbral texture and phonetic content, crucial for speech and audio recognition.

Model Prediction

Verify Feature Shapes

Once the features are extracted, their shapes are verified to ensure they match the expected format for model input. This step ensures compatibility between the extracted features and the model, preventing potential errors during prediction.

Reshape Tester Data

A function is defined to reshape tester data to be compatible with the pre-trained model. This step is crucial for ensuring that the data fed into the model during prediction matches the format used during training.

Predict Real or Fake

The pre-trained model is then used to predict whether the audio is real or fake based on the extracted features. The results of these predictions are displayed to provide insights into the model's performance and to evaluate its accuracy in classifying the audio samples.

Audio from Video

Extract Audio Using MoviePy

The project includes a step to extract audio from video files using the `moviepy` library. This extracted audio is saved as a separate file, which can then be processed in the same way as other audio files in the dataset.

Extract and Prepare Features

Features are extracted from the saved audio file using the defined function. These features are prepared for model prediction, ensuring they match the expected format and contain all necessary information.

This documentation provides a thorough overview of the deepfake detection project, detailing the steps from dataset processing to feature extraction and model prediction. By leveraging an updated dataset and advanced techniques, the project aims to achieve accurate and reliable results in detecting deepfake audio.

Research paper of audio dataset used as reference:

https://arxiv.org/pdf/2308.12734

CONCLUSION:

To conclude, I will just link my Kaggle Notebook, that contains the model training and testing code.
https://www.kaggle.com/code/pyknight73/my-version-of-deepfake-detection/edit/run/189627138