Kashish Gandhi[1,2]
Prutha Kulkarni[1,3*]
Taran Shah[1,4*]
Piyush Chaudhari[1,5*]
Meera Narvekar[1,6]
Kranti Ghag[1,7]

A Multimodal Framework for Deepfake Detection

Journal of Electrical Systems

Abstract :- The rapid advancement of deepfake technology poses a significant threat to digital media integrity. Deepfakes, synthetic media created using AI, can convincingly alter videos and audio to misrepresent reality. This creates risks of misinformation, fraud, and severe implications for personal privacy and security. Our research addresses the critical issue of deepfakes through an innovative multimodal approach, targeting both visual and auditory elements. This comprehensive strategy recognizes that human perception integrates multiple sensory inputs, particularly visual and auditory information, to form a complete understanding of media content. For visual analysis, a model that employs advanced feature extraction techniques was developed, extracting nine distinct facial characteristics and then applying various machine learning and deep learning models. For auditory analysis, our model leverages mel-spectrogram analysis for feature extraction and then applies various machine learning and deep learning models. To achieve a combined analysis, real and deepfake audio in the original dataset were swapped for testing purposes and ensured balanced samples. Using our proposed models for video and audio classification i.e. Artificial Neural Network and VGG19, the overall sample is classified as deepfake if either component is identified as such. Our multimodal framework combines visual and auditory analyses, yielding an accuracy of 94%.

Keywords**:** Deepfake Detection, Deep Learning, Multimodal, Machine Learning, Feature Extraction

1 Introduction

Recent advances in deep learning and media technologies have made synthetic generation of media more accessible than ever. It requires minimal effort to allow consumers to manipulate all kinds of media and spread misinformation. This can lead to fraudulent activities, scams, and spreading of misinformation [1], [2]. Several approaches have been proposed gravitating towards a unimodal approach, meaning they take only one modality into account, either audio or video [3],[4]. However, these unimodal approaches fall short in addressing complexity and nuance of sophisticated deepfakes. Although these detectors have shown impressive performances, video-only detectors can be deceived by synthetics audios and vice-versa. Hence, these prove to be ineffective for robust deepfake detection. To overcome these limitations, a multimodal analysis framework is proposed, integrating visual, auditory, and textual features to provide a holistic view of the media content. This approach not only enhances detection capabilities but also offers a more resilient solution against the evolving landscape of deepfake technologies. Through this paper, a new method that combines audio-visual information is presented. Custom features from the video dataset and spectrograms from the audio were extracted dataset during the training phase. Our models were trained on unimodal datasets which helps the developed detector to not overfit on a single data type and generalize as much as possible.

2 Literature Review

Deepfake technology has become increasingly prevalent, with various tools available online for creating synthetic videos. Popular applications such as FakeApp, Faceswap, DeepFaceLab, and Face Swap Generative Adversarial Network utilize autoencoder-decoder and Generative Adversarial Network architectures to produce realistic deepfakes. In audio deepfakes, Google's text-to-speech technology has driven a rise in usage, supported by tools like WaveNet by DeepMind and TacoTron by Google as mentioned in the survey [5]. These technologies often use Variational Autoencoders (VAEs), which compress and reconstruct audio to mimic target speakers, and Generative Adversarial Networks (GANs) for audio manipulation. This proliferation of deepfake technology presents significant cybersecurity threats, prompting research into effective detection methods.

Unimodal approaches to deepfake detection—those focusing solely on either audio or visual analysis—often fall short in handling the complexities of sophisticated deepfakes. Studies below by various researchers demonstrate that relying solely on visual features, such as facial characteristics or eye blinking patterns, can lead to inaccuracies as these methods may be vulnerable to variations in data or specific attack vectors that manipulate only one modality.

[1]Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
[2]Corresponding author: kashishgandhi6112003@gmail.com
[3]kulkarniprutha1@gmail.com
[4]taran.shah9@gmail.com
[5]piyush300504@gmail.com
[6]meera.narvekar@djsce.ac.in
[7]kranti.ghag@djsce.ac.in

[*] *Equal authorship*

However, their approach's reliance on fixed feature extraction limits its effectiveness on new data. Similarly, [7] tracked eye blinking patterns to differentiate real and deepfake videos using Fast-Hyperface, but this method is sensitive to individual blinking variations and environmental factors, leading to false positives or missed detections. Our method will mitigate this by combining more than one visual features, providing a more comprehensive detection framework that is less reliant on a single feature.

Computational limitations and the use of incompatible models can significantly impact the effectiveness of deepfake detection systems. Many existing studies, such as those by [8],[9] and [10], highlight how the choice of models and computational resources can constrain the performance of detection algorithms. For instance, [8] combined resource-intensive models like ConvNeXt, Swin Transformer, Autoencoder, and Variational Autoencoder, requiring substantial computational power, which is a barrier for real-time applications. Additionally, [9] used a 3D CNN to process videos directly, capturing spatial and temporal information but leading to high computational costs and significant memory use. Concurrently, in the study [10] they have used 7 different models including CapsuleForensics and Xception achieving up to 77% accuracy. Even with a custom-curated dataset, their approach only achieved limited success due to the use of multiple models that were not well-integrated with the type of data they analyzed, leading to inefficiencies and reduced accuracy.

Our research journey in deepfake detection was significantly shaped by insights from various key papers. The study began by exploring [11], which utilized MTCNN for face detection and EfficientNet-B5 for feature extraction, illustrating the importance of accurate face detection in our pipeline. This led us to integrate reliable face detection with advanced feature extraction. Considering [12], which employed a 3D CNN to capture spatial-temporal features directly from video frames, highlighting the value of capturing video frames to reduce the processing time and computational resources. Inspired by [13], which combined CNN-generated features with LSTM for sequential analysis, recognizing the importance of integrating sequential models for detecting long-term dependencies in videos. Also, after reviewing [14], which used CNN architectures like VGG19 for detecting facial artifacts, guiding us to evaluate and fine-tune these models. Furthermore, [15] emphasized the critical role of facial features such as eyes and mouth, leading us to focus on feature extraction methods sensitive to these regions. Additionally, [16] demonstrated the benefits of Convolutional Neural Networks (CNNs) to capture spatial-temporal dynamics and extract detailed features, inspiring us to apply CNN to for deepfake detection.

In recent research, several advancements have been made in the field of audio processing and deepfake detection. [17] explored deepfake audio detection by leveraging Explainable Artificial Intelligence (XAI). Their study emphasized the need for interpretability in deepfake detection systems, which is crucial for understanding and justifying the model's predictions. Study in [18] compared Mel Frequency Cepstral Coefficients (MFCC) and Mel spectrograms for raga classification using CNNs. Their comparative analysis provided insights into which feature extraction technique yields better results for audio classification tasks. They found that while both MFCC and Mel spectrograms are effective, Mel spectrograms offer more detailed temporal and spectral information, which can be advantageous for tasks like raga classification and potentially for deepfake audio detection as well.

A more advanced method for creating Mel-spectrograms involved converting MFCC to Mel-spectrograms. [19] implemented a custom 7-layer CNN model but couldn't determine its real-world performance. Observing that tokenizing words for classification was challenging as identical words were spoken in human and fake voices. This led us to focus on pitch, a key differentiating factor, and thus adopted Mel-spectrograms for better pitch representation. The work by [20] applied MFCC feature extraction and classified audio samples using SVM and VGG-16 transfer learning technology. Additionally, [21] demonstrated that transfer learning with the VGG19 model, a pre-trained 19-layer network augmented with additional layers, effectively classifies sound images and their corresponding audio. Thus, inspired by these approaches, VGG19 technology and mel-spectograms for audio analysis were explored. Recent studies have demonstrated the potential of multimodal methods in improving the detection of synthetic media. For instance, in [22],[23] and [24], the authors used a multimodal approach, extracting and fusing facial and speech features to improve deepfake detection accuracy, showing that multimodal systems can outperform unimodal counterparts in detecting deepfakes, inspiring us to adopt a similar strategy.

By integrating these insights, a deepfake detection system that addresses the limitations of unimodal approaches by combining both visual and audio features was developed. Our system is designed to be computationally efficient, leveraging advanced feature extraction and sequential models, and is capable of generalizing across different datasets and deepfake types. This comprehensive approach ensures robust and accurate detection, overcoming the challenges identified in previous research.

3 Data Collection

3.1 Video Dataset

A dataset that matched the characteristics of contemporary Deepfake videos was selected. The training subset, derived from DFDC Dataset, [25] consists of 2,619 deepfake videos and 515 real videos, representing different age, race, gender, and demographic groups, ensuring a broad and inclusive training dataset. This diversity is important very to develop a robust model that can accurately detect deepfake in different populations segments.

3.2 Audio Dataset

For audio dataset, the Fake-or-Real (FoR) dataset [26] was utilized, which comprises over 195,000 utterances from both real human speech and computer-generated synthetic speech. The dataset is available in four versions: for-original, for-norm, for-2sec, and for-rerec. The for-norm version, containing 26,927 fake and 26,941 real audio samples, and the for-rerec version, which includes an equal split of 5,104 fake and 5,104 real samples was selected. Our model was trained on a subset of 4,000 samples from this dataset ranging from 2 to 4 seconds in duration, providing a diverse range of audio for effective detection of synthetic speech patterns.

4 Proposed Methodology

The proposed methodology includes a multi modal approach to make one of 4 classifications whether it is:

a) Real video with Real audio
b) Real video with Deepfake audio
c) Deepfake video with Real audio
d) Deepfake video with Deepfake audio.

The input video will first have its audio extracted and passed through our audio deep fake detection model which generates and extracts the perfect Mel-Spectrogram of the audio samples as well as MFCC [27] for the same. These were passed through various models for classification as real and deep fake audio. The video on the other hand was passed through the feature extraction model which was stored as an array and consequently passed through the classification model. This resulted in an output from the mentioned outcomes

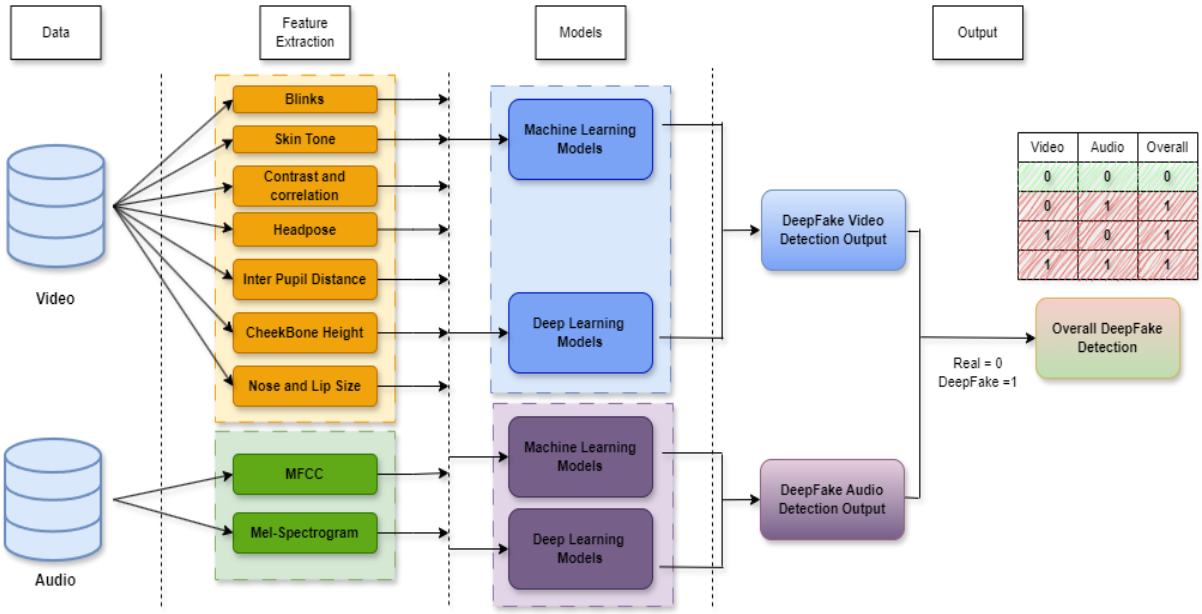


Fig.1: Pipeline for proposed methodology

4.1 Feature Extraction

4.1.1 For Video As the initial step in preprocessing our data, nine features were extracted from the videos. Seven of these features involved detecting facial landmarks and extracting pertinent attributes. To achieve this, the Haar Cascade algorithm, since it can easily detect objects in images irrespective of their scale and location. Hence, it was employed to accurately identify faces within the frame, defining our region of interest (ROI). Subsequently, FaceMesh's Face Landmark Model was utilized to detect specific features within the identified ROI:

- **Nose and Lip Size:** Research shows in [28] that size of facial features is often tampered with when a deepfake is generated. Hence, it was important to analyze the same for real and deepfake videos. Landmark indices used were 1 and 197 for the base and tip of the nose and 61 and 291 for the left and right corners of the nose, respectively, obtained using facemesh. The distance between the two points was found using the distance

formula. A similar approach was used to calculate the lip size where the distance between either corner of the lips was found and computed using the Euclidean distance formula.

(a) (b) (c)

Fig.2: The image shows the process of detecting deepfake alterations in facial features by analyzing the distances between key nose and lip landmarks. Subfigures are labelled as follows (left-to-right): (a) Original image, (b) Lip Indices, and (c) Nose Indices.

- **Contrast and Correlation:** Textural features are complex features that indicate roughness and regularity of an image. Textural features based on the Gray Level Co-Occurrence Matrix were extracted. It describes texture using the spatial distribution of pixels in an image. Using this matrix, the probability of two gray pixels being adjacent is found by computing distance and direction. The probability on different gray levels constitutes the gray level co-occurrence matrix. According to a previous study by Xu, Bozhi et al in [29], a total of 14 features can be derived from the GLCM. In our study, two features from this matrix: contrast and correlation were selected since they help extract most valuable spatial relationships between pixels.

  Contrast measures the richness and depth of texture details, with higher values indicating a greater gray-scale difference between pixels. The formula for calculating contrast is as follows:

$$f_{Con} = \sum_{i,j=1}^{N} P_{i,j}\,(i-j)^2 \tag{1}$$

  Correlation measures the degree of correlation between elements of the gray level co-occurrence matrix.

$$f_{Cor} = \frac{\sum_{i,j=1}^{N}(i-\mu_i)(j-\sigma_j)P_{i,j}}{\sigma_i\sigma_j} \tag{2}$$

  Where,

$$\mu_i = \sum_{i,j}^{N} iP_{i,j} \tag{3}$$

$$\mu_j = \sum_{i,j}^{N} jP_{i,j} \tag{4}$$

$$\sigma_i = \sqrt{\sum_{i,j=1}^{N} P_{i,j}\,(i-\mu_i)^2} \tag{5}$$

$$\sigma_j = \sqrt{\sum_{i,j=1}^{N} P_{i,j}\,(i-\mu_j)^2} \tag{6}$$

N is the size of the gray-level co-occurrence matrix, and $P_{i,j}$ is the value of the i-th row and j-th column of the gray-level co-occurrence matrix. To implement the above. This ROI was divided into 9 blocks to effectively deal with areas of the face that had been tampered with. Textural features were accurately extracted from each of the 9 sub-blocks and stored.

- **Blinks:** As mentioned in the research [7] by T. Jung et al. often it is seen that deepfakes have an irregular pattern in the blinking of the eye. This method tracks the blinking of the eyes as one of the features used in determining whether the video is real or deepfake. A combination of 2 models to extract the eye blinking count during the duration of a video was used. The outline of lining of the eye was noted, which helped find a ratio between the vertical height between the eyelids and the horizontal distance between the opposite corners of the eyes. The landmarks marked: 22, 23, 24, 26, 110, 157, 158, 159, 160, 161, 130, 243 were used to delineate the lining of the eye, as shown in the Fig 3.

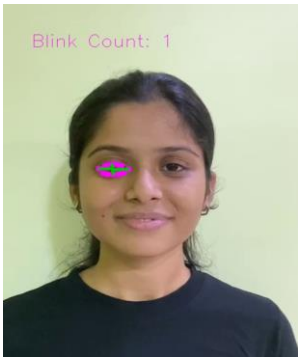


Fig.3: Blink feature extraction by marking key eye landmarks and tracking blinking patterns.

- **Inter Pupil Distance:** Often when deepfakes are created, the normal interpupil distance is tampered with, making it a notable feature that can be used to distinguish between real and deepfake videos. The landmarks at the center top and center bottom of each eye were noted. Using each of those distances and taking their average ,a very accurate estimate of the inter-pupil distance was found.

Fig.4: Inter-pupil distance feature extraction, using landmarks at the center top and bottom of each eye to accurately measure the distance.

- **Cheek Bone Height:** Cheekbone height helps analyze whether the video is a deepfake or real as abnormal distance from the chin to the cheekbones is a common occurrence in deepfake videos. For calculating the height from the chin to the cheekbones, use of basic geometry was made. A combination of the sine and cosine rules was used to find the cheekbone height. Figure 5 shows an outline of the skeleton used to mark the cheekbone height, along with the equations used to determine the actual cheekbone height.

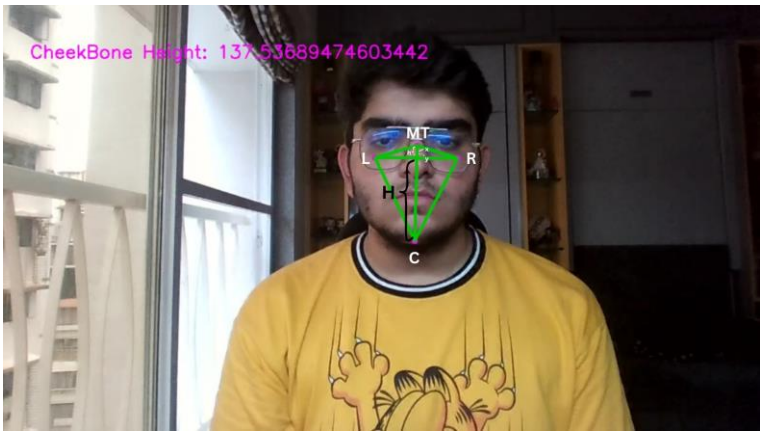


Fig.5: The extraction of cheekbone height feature, including the diagram used for implementing the mathematical calculations.

Our proposed analytical calculations:

$$cos\ R = \frac{LR^2 + MTR^2 - MTC^2}{2.LR.MTR} \tag{7}$$

$$cos\ x = \frac{MTR^2 + MTC^2 - RC^2}{2.MTR.MTCR} \tag{8}$$

$$y = 180° - (x + R) \tag{9}$$

$$\frac{sin\ R}{h} = \frac{sin\ y}{MTR} \tag{10}$$

$$h = \frac{sin\ R\ .MTR}{sin\ y} \tag{11}$$

$$H = MTC - h \tag{12}$$

Where: LR is the horizontal distance between the two cheekbones, MTC is the distance between the middle of the nose and the chin. R is the angle between MTR and LR, x is the angle between MTR and MTC and H is the height difference between the cheekbones and the chin.

To implement the above. The face region was divided into 4 sections using the quadrilateral kite. The cosine rule is used to find out angle R and angle x. These are the used to find angle y. Further, sine rule is used to find h and subtracting h with MTC results in H which is the cheekbone height.

- **Headpose:** The head movements along the x, y, and z axes are not uniform or consistent in a deepfake compared to a real video. A tendency to move our heads while talking is common; however, upon conducting research, two prominent studies by [30] and [31] that this is not the case for deepfakes. Often in a deepfake, the head movement does not represent the natural face as it looks more like a mask. This leads to irregular movements where even though the head is completely on the other side, the face is not. Hence, head movement can be used as a parameter to gauge whether the video is a deepfake or not. The camera matrix was utilized to gain depth perception in the frame. The solvePnP function was used to get the rotational and translational vectors that describe the 3D pose of the face relative to the camera. Then cv2's Rodrigues transformation was used to convert the rotational vector into a rotational matrix, following which the RQDecomp3x3 function was used to convert the rotational matrix into Euler angles which were then used to calculate and estimate the headpose.

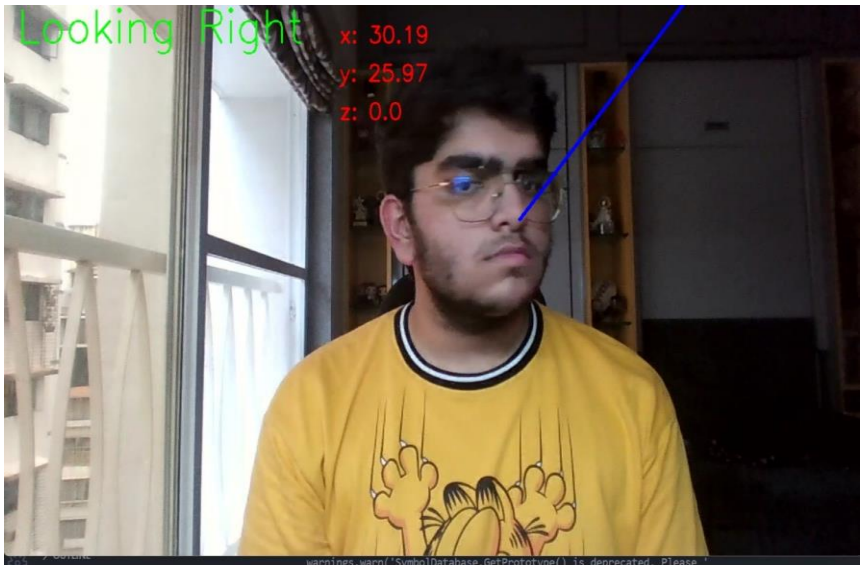


Fig.6: Headpose estimation, using camera matrix and 3D pose calculations to analyze head movement inconsistencies.

- **Skin Tone:** Skin tone helps in facial color analysis by detecting changes in blood flow and its concentration. It has been proven that color spaces other than RGB perform better while detecting color according to Hadas, Shahar et al [32]. Hence, use of the oRGB color space was made. The oRGB color space is a color space that separates color information into different channels to enhance features for skin detection. It models color perception using three channels:
  1. Luminance [L]: A grayscale value representing brightness.
  2. Chrominance [C1]: The difference between red and green color channels.
  3. Chrominance [C2]: The difference between blue and yellow colors.

To convert an RGB pixel to an oRGB pixel, the following linear transformation is performed:

$$\begin{bmatrix} L \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & 0.500 & -1.000 \\ 0.866 & -0.866 & 0.000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

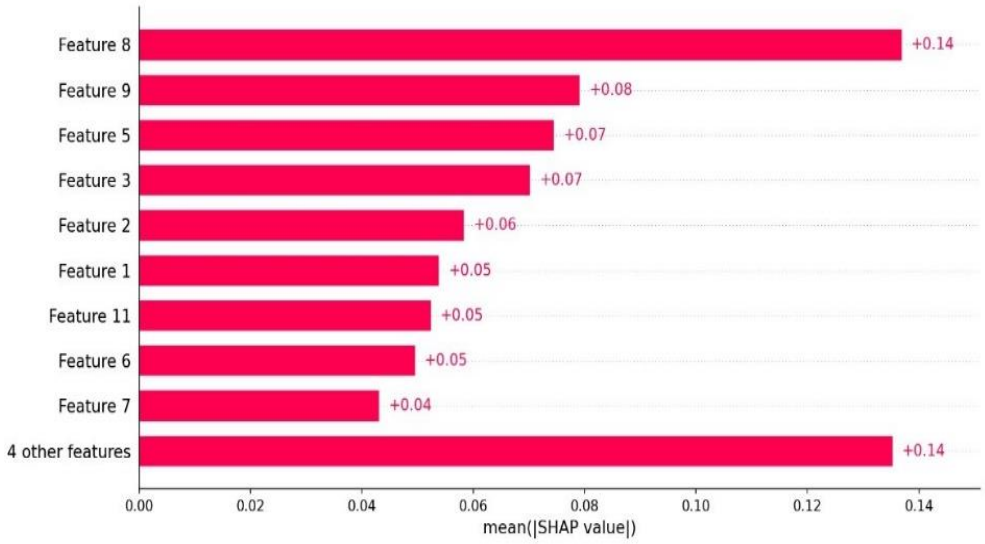


Fig.7: Feature importance for deepfake video detection. The x-axis represents the mean absolute SHAP (SHapley Additive exPlanations) values, indicating the average impact of each feature on the model's output. The features include cheekbone height, inter-pupil distance, number of blinks, headpose angles (x, y, z), nose size, lip size, contrast correlation, luminance, chrominance1, chrominance2, and others, listed from 1 to 13, respectively. Higher SHAP values indicate greater importance of a feature in the model's predictions.

4.1.3 For Audio To understand and visually study audio signals, mel-spectrograms were utilized. ”Mel” is an abbreviation for ”melody.” Mel-spectrograms [27] provide a time-frequency representation of audio signals with perceptually relevant amplitude and frequency representations. Both amplitude and frequency perceptions are nonlinear and can be expressed in logarithmic form. The mel scale is derived from a perceptually informed scale for the pitch of sound. The pitch for a 1kHz frequency of sound is perceptually similar to 1000 mels, making the pitch of the sound equivalent. The studies in [33], [34], and [35] provided us with a deep understanding of mel-spectrograms, which was effectively applied in our research.
The formula was from [34] to conduct the following:

$$m = 2595 \, . \, log_{10}(1 + \frac{f}{700}) \tag{13}$$

$$f = 700 \, (10^{\frac{m}{2595}} - 1) \tag{14}$$

Steps for Mel-Spectrogram Generation
1) Perform Short-Time Fourier Transform (STFT)
2) Convert Amplitude to Decibels (dB)
3) Convert Frequencies to Mel Frequency Representation

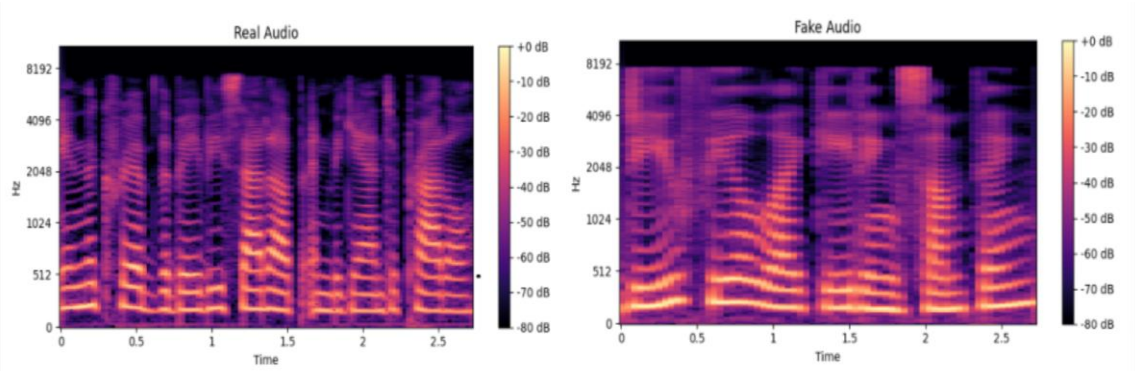


Fig.8: Mel-spectrograms comparing real (left) and deepfake (right) audio signals reveal distinct differences in time-frequency representation and amplitude. The fake audio often exhibits a broader frequency range and unique spectral signatures, with more harmonics and clearer patterns, unlike the real audio, which includes background noise and vocal imperfections.

Choosing the Appropriate Number of Mel Bands
The number of mel bands depends on the specific problem. For the research, 128 bands were used. This choice is informed by the 88 notes on a piano, which correspond to approximately 90 mel bands, aligning with the notes of Western music state in [18].

Construction of Mel Filter Banks
1) Convert the lowest and highest frequencies to their mel representations using the formula for m.
2) Create bands with equally spaced points, based on the desired number of mel bands.
3) Convert these points back to Hertz.
4) he frequency bins were rounded to the nearest value due to the discrete nature of signals and the constrained resolution imposed by the frame size of the Short-Time Fourier Transform (STFT).
5) Create triangular filters, which are the building blocks of the mel scale.
6) Higher frequencies have larger gaps between points to achieve the same pitch compared to mel frequencies, which have similar pitch.
7) When plotted, this shows triangular-shaped filters.
8) The shape of mel filter banks is geometric, but the calculations are algebraic.

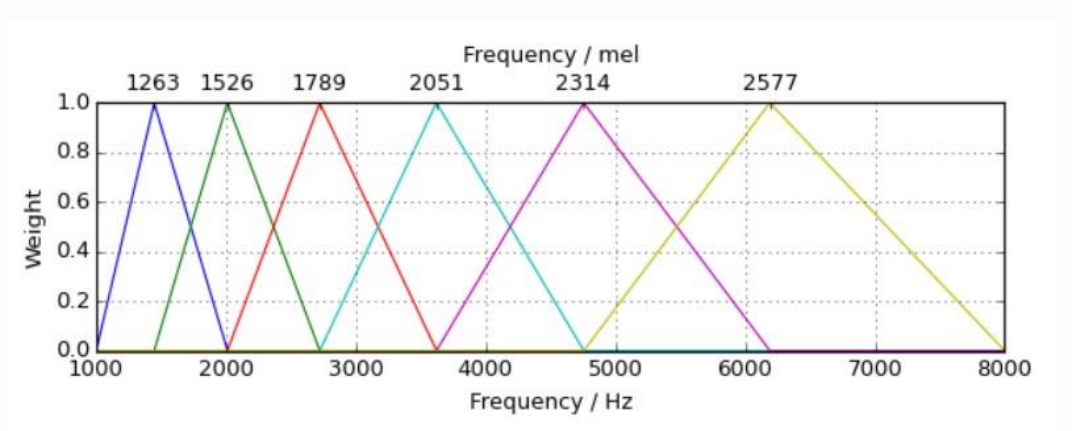


Fig.9: Plot of mel filter bank weights against mel frequencies and Hertz frequencies. The graph visualizes how triangular mel filters map frequency bands from Hertz to the mel scale, demonstrating the mel scale's frequency distribution [36].

Applying mel filter Bank to Matrix to Normal spectrograms

$$M_{MelFilterBank} = (number\ of\ bands, (frame\ size/2) + 1) \tag{15}$$

$$Y_{Spectogram} = ((frame\ size/2) + 1, number\ of\ frames) \tag{16}$$

$$Mel - Spectogram = M.Y_{(number\ of\ bands\ ,number\ of\ frames)} \tag{17}$$

Matrix multiplication is possible as the rows of *M* and *Y* are the same.
Each point in the graph indicates the presence of a mel band at a specific point in time. This is represented using different color combinations based on dB values. Mel spectrograms find applications in various fields such as audio classification, music genre classification, music instrument classification, and automatic mood recognition systems.

4.2 Models Used

4.2.1 For Video Initially, the videos were processed through our feature extraction model, resulting in a final feature dataframe with 2,590x13. Given the high imbalance between fake and real videos, the SMOTE (Synthetic Minority Over-sampling Technique) method was employed to upsample the dataset. After upsampling, the feature dataframe expanded to 4,342x13. The data was subsequently split into training and testing sets in an 80:20 ratio. This was then fed to the various models mentioned below.

Decision Trees were used to identify deepfake videos because it effectively handles various features extracted from videos, such as facial landmarks, texture characteristics, and skin tone. However, they fail with high-dimensional and complex data which might lead to overfitting and poor generalization. To tackle this, Random Forest was used to identify deepfake videos because of its ability to improve classification accuracy and robustness through ensemble learning. By constructing multiple decision trees and pooling their predictions, random forests reduce overfitting and increase generalization. This approach efficiently handles diversity and complexity from video data, combining the strengths of individual trees to provide more reliable classification.

Since this approach did not effectively capture intricate patterns and temporal dependencies, bagging was employed to enhance model stability and accuracy by reducing variance. By training multiple models on different bootstrap samples of the data and comparing their predictions, bagging reduces the risk of overfitting any one sample. To increase the accuracy, XGBoost was used as it focuses on patterns that are difficult to classify. This method iteratively refines the model by emphasizing misclassified instances, leading to a more robust and accurate detection system for distinguishing between real and deepfake videos.

All the above methods used machine learning techniques which can't handle complex patterns as effectively as deep learning models. With the intention of creating our own deep learning model and not using the pre-trained ones, Artifical Neural Network (ANN) for Deepfake video detection was chosen as they excel in recognizing complex patterns and representations from data. Inspired from the human brain, ANNs consist of multiple neural networks that learn certain features through adaptive training. This capability is crucial for detecting subtle and complex differences in real and deepfake video.

The feedforward structure of ANNs allows them to process input data across multiple layers, capturing nonlinear relationships and high abstractions that are missing in simple models. The use of activation functions helps detect nonlinearities, enabling the network to model complex patterns. Furthermore, the binary cross-entropy loss function efficiently handles the classification task by considering the difference between the predicted probabilities and the actual labels, which guides the network to improve the accuracy of the difference between real and deepfake video.
For a feedforward neural network, the output of a single neuron j in layer l is:

$$a_j^{(l)} = \sigma\left(\sum_{i=1}^{n_{l-1}} w_{ij}^{(l-1)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{18}$$

Where σ is the activation function, w_{ij} are the weights, a_i are the activations from the previous layer, and b_j is the bias term.
Loss Function For binary classification, the loss function (binary cross-entropy) is:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i})] \tag{19}$$

Where y_i is the true label, $\hat{y_i}$ is the predicted probability, and N is the number of samples.

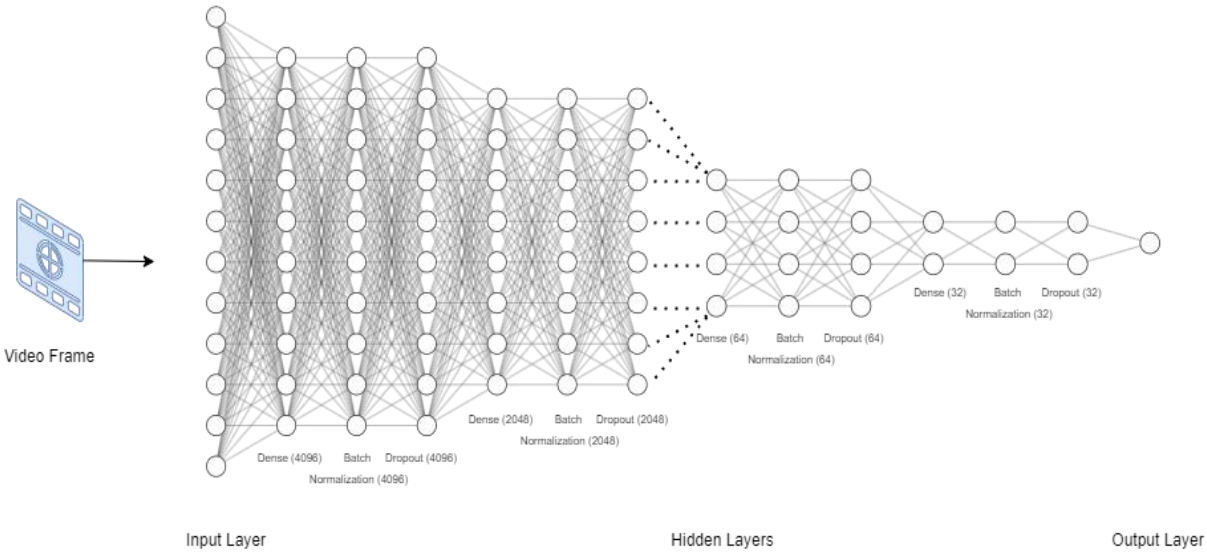


Fig.10: Architecture of the Artificial Neural Network (ANN) used for Deepfake video detection, illustrating the feedforward structure with multiple layers and activation functions to capture complex patterns.

4.2.2 For Audio After extracting the features split the data into training-testing ration of 80:20. During training, validation data comprising a randomly distributed set of samples was used to monitor the model's performance.
Random Forest and XGBoost were used to classify labelled audio samples due to their strong performance in complex data processing. Random forest with a sufficient number of estimators effectively reduces overfitting and provides robust classification results. XGBoost provided increased flexibility with its wide range of parameters, allowing to fine-tune the model for improved accuracy. Despite achieving adequate accuracy with these techniques, deep learning methods were selected to further reduce loss and enhance model performance, leveraging their capability to capture complex patterns in audio data.

Convolutional Neural Networks (CNNs) were used for classifying mel spectrograms because CNNs excel at identifying and learning spatial hierarchies in image-like data. In our case, mel-spectrograms are 2D representations of audio signals, where spatial patterns can reveal important features for classification. CNNs are particularly effective at detecting these patterns through their convolutional layers, which apply filters to capture features such as edges, textures, and shapes. The architecture of our CNN, with layers like Conv2D and MaxPooling2D, is designed to extract and downsample features from the mel-spectrograms, creating feature maps that highlight relevant information. The use of layers like Flatten, Dense, and Dropout further helps in combining these features, preventing overfitting, and improving classification

performance. By leveraging CNNs, complex patterns were easily learned and achieved higher accuracy in distinguishing between different audio samples.

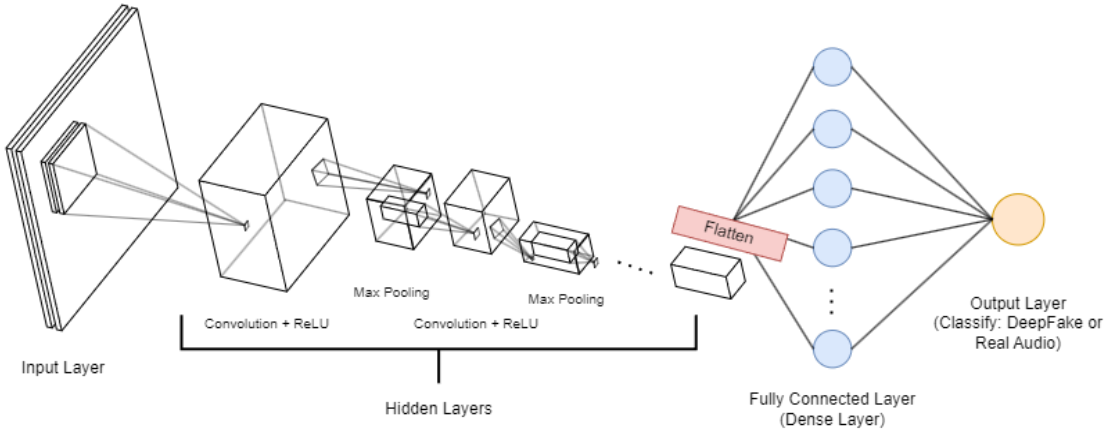


Fig.11: Architecture of the Convolutional Neural Network (CNN) used for classifying mel-spectrograms, highlighting layers such as Conv2D and MaxPooling2D for feature extraction and classification.

VGG19 was used to achieve the highest accuracy for our classification task due to its proven performance and the advantages of transfer learning. VGG19 is a well-established convolutional neural network architecture known for its deep and uniform layer structure, which allows it to capture intricate features from images. By utilizing pre-trained weights from ImageNet, VGG19 provides a strong starting point with learned features that can be adapted to our specific problem.

VGG19 was configured with an input shape of 224×224×3 to match the size and color channels of our mel-spectrogram images. By freezing all but the last 4 layers of the pre-trained VGG19 model, the valuable feature extraction capabilities were preserved while allowing fine-tuning on our specific dataset. This approach leverages the robust features learned from a large, diverse dataset while adapting the model to our task. The inclusion of additional dense and dropout layers refined the model and mitigated overfitting, ultimately boosting its performance for binary classification.

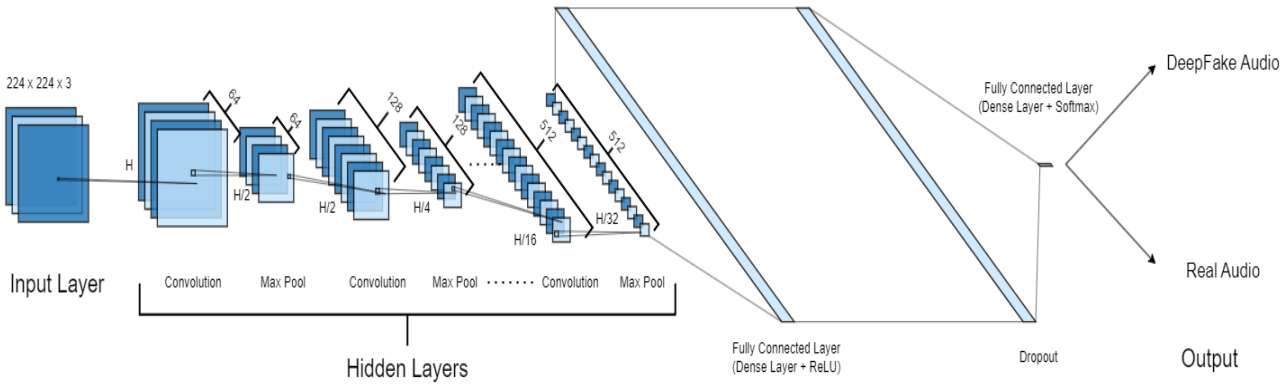


Fig.12: VGG19 architecture adapted for mel-spectrogram classification, showcasing transfer learning with pre-trained weights and fine-tuning for enhanced performance.

5 Experimental setup

The devices used for data collection and video feature extraction contained processors using i5 12th generation with Nvidia RTX 3050 16 GB ram. Similarly, devices used for audio feature extraction contained processors using i7 12th generation with Nvidia RTX 3060, ddr6 GPU.

6 Results

The model's performance is summarised in a classification report, which includes metrics such as precision, recall, F1 score, and accuracy. Table 1 presents a detailed classification report for Deepfake video detection and Table 2 presents the detailed classification report for Deepfake Audio detection

Table 1: Performance Metrics for Various Methods for Deepfake Video

Method	Precision	Recall	F1-Score	Accuracy
Decision Tree	0.80	0.80	0.80	0.80
Random Forest	0.88	0.88	0.89	0.89
Bagging	0.94	0.90	0.92	0.92
XGBoost	0.94	0.91	0.92	0.92
ANN	0.88	0.96	0.93	0.93

Table 2: Performance Metrics for Various Methods for Deepfake Audio

Method	Precision	Recall	F1-Score	Accuracy
Random Forest	0.83	0.82	0.82	0.82
Gradient Boosting	0.86	0.86	0.86	0.86
CNN	0.90	0.91	0.90	0.90
VGG19	0.98	0.97	0.98	0.98

Proposed Multimodal Approach Results: The original dataset was utilized in splits for training and testing, while randomly swapping real audio with deepfake audio. During testing, a balanced distribution of samples is ensured across the following categories: 'real-deepfake' (real video with deepfake audio), 'deepfake-real' (deepfake video with real audio), 'real-real' (real video with real audio), and 'deepfake-deepfake' (deepfake video with deepfake audio). The models with the highest accuracy were used for video and audio classification to generate the final output for the video. If either the video or audio component is identified as deepfake, the overall sample is classified as deepfake.

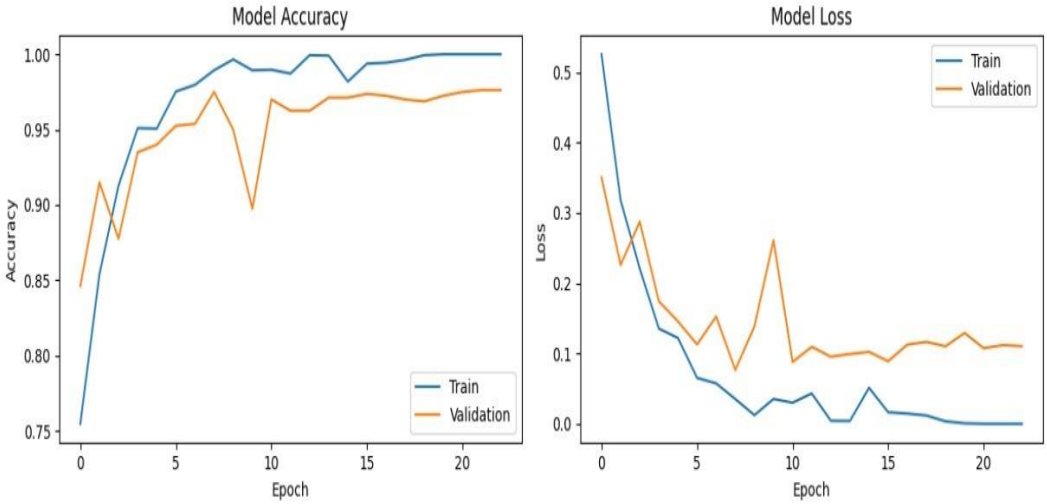


Fig.13: Training performance of the VGG19 model, displaying accuracy (left) and loss (right) versus epochs, illustrating the model's learning progress over time.

Table 3 shows that the approach correctly classified 1955 samples out of 2079 samples giving it an accuracy of 94%.

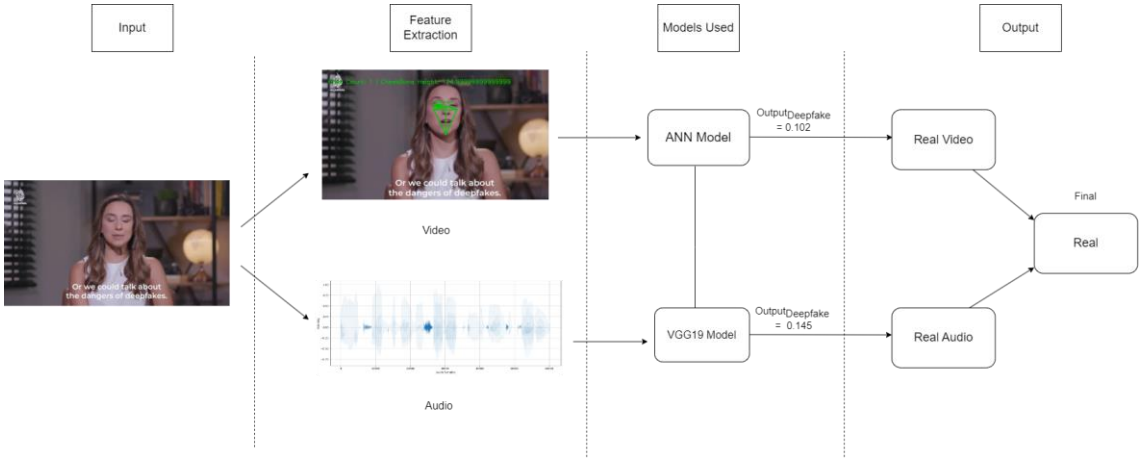
Table 3: Correctly Classified Data for Multimodal Data

Video	Audio	Number of Samples	Correctly Classified
0	0	528	502
0	1	523	496
1	0	513	477
1	1	515	480

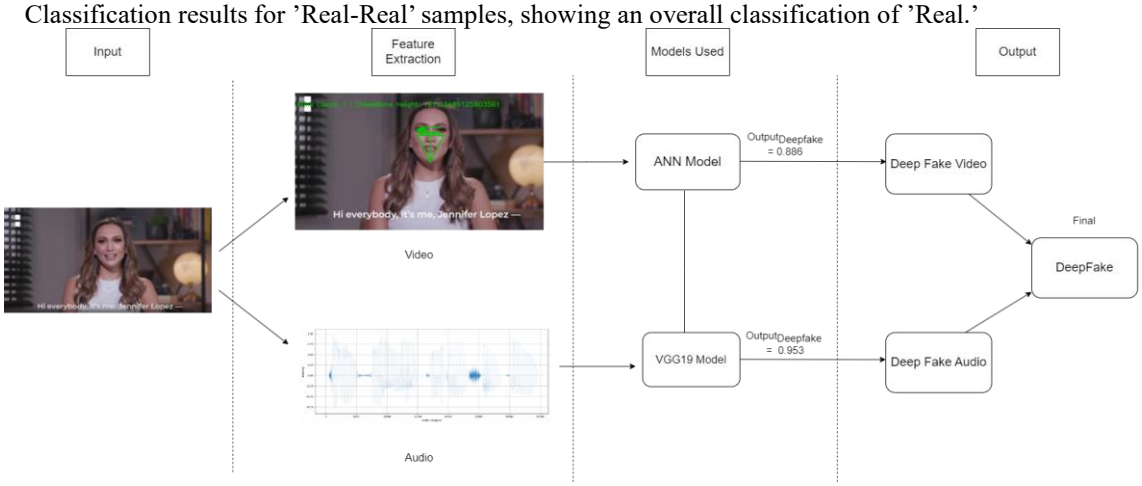
Here, '0' denotes real media whilst '1' denotes deepfake media.

7 Conclusion

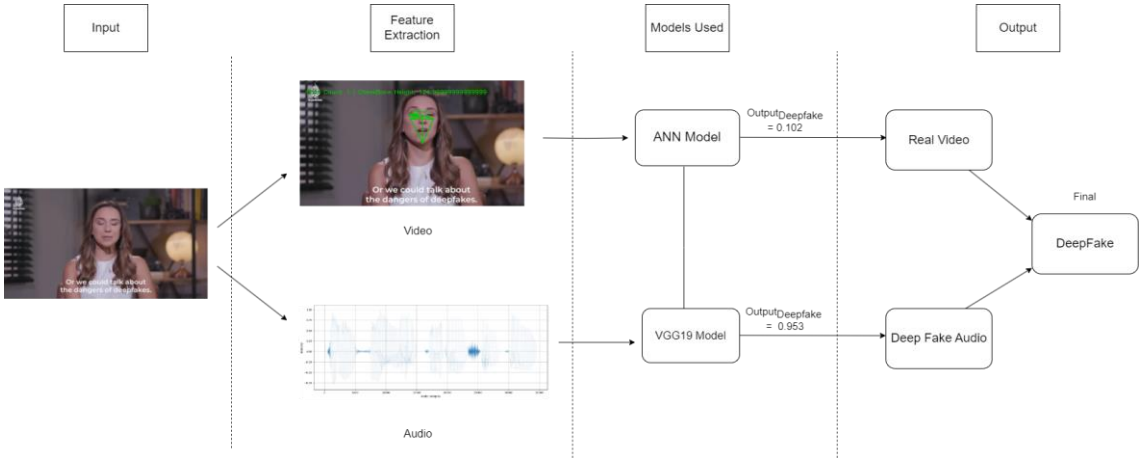
The deepfake detection methodology employs advanced deep learning techniques for both video and audio analysis. Specifically, ANN is utilized for video classification with an accuracy of 93%, while transfer learning with VGG19 is used for audio classification, achieving an accuracy of 98%. These models outperform traditional algorithms like Random Forest, Decision Tree, XG-Boost, and Bagging, which were less effective in the comparative analysis. The combined approach results in an overall accuracy of 94%, demonstrating its robustness and effectiveness. Our approach improves on previous works, such as the 86.13% accuracy with XG-Boost in [6], by integrating multiple visual features into a comprehensive multimodal framework. While studies like [29], [30], and [32] focus on single features—such as textural features or head poses—they often lack reliability on unseen data where other features may vary. By leveraging multiple features, our method offers greater robustness in detecting deepfakes across diverse scenarios.



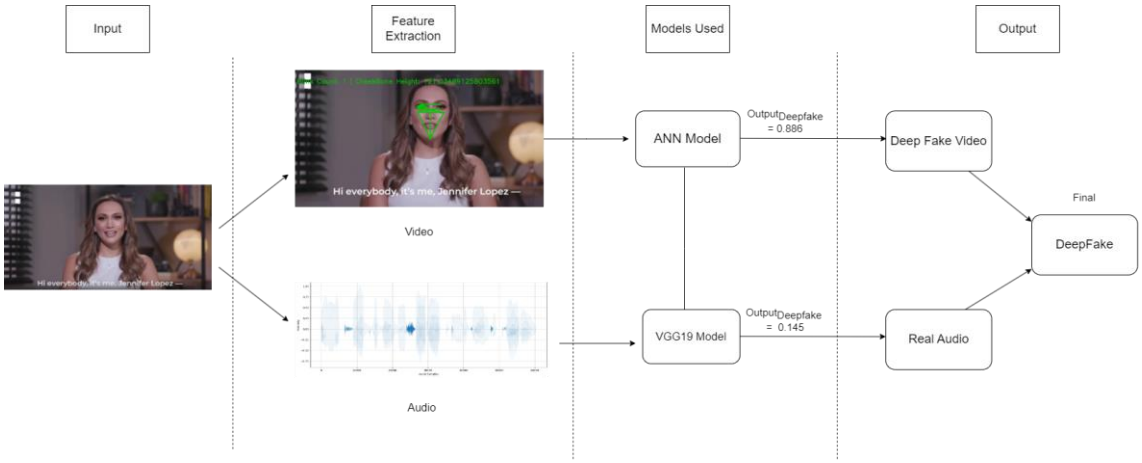
a) Classification results for ’Real-Real’ samples, showing an overall classification of ’Real.’



b) Classification results for ’Deepfake-Deepfake’ samples, showing an overall classification of ’Deepfake.’



c) Classification results for ’Real-Deepfake’ samples, showing an overall classification of ’Deepfake.’



d) Classification results for 'Deepfake-Real' samples, showing an overall classification of 'Deepfake.'

In contrast, studies like [8], [9], and [10] face challenges due to computational constraints and model inefficiencies. For example, [8] combined resource-intensive models, leading to high computational costs, while [9] used a 3D CNN with high memory usage. Despite using multiple models, [10] achieved only 77% accuracy, underscoring the need for a more streamlined and effective approach like ours.

The detection of deep fake audio in our work surpasses previous methods, such as in [33], where a custom CNN achieved 88.9% top-5 accuracy and VGG19 reached 88.5%, both significantly lower than our model's 98.0%. Similarly, [19] implemented a 7-layered CNN with 91% accuracy using MFCC features, but our model outperformed it without needing MFCC extraction, enabling faster processing. Additionally, they lacked real-time evaluation results, which our model successfully achieved.

By combining visual and audio features, our multimodal approach addresses the limitations of unimodal systems, ensuring comprehensive feature extraction and robust detection. This method demonstrates superior accuracy and reliability, making it a significant advancement in the field of deepfake detection.

8 Future Scope

The DFDC dataset for deepfake video detection had 50 zip files, but our model was trained on just 1 zip file with 3,135 videos. With enhanced computational resources, this approach could be extended to the entire dataset and other datasets, improving deepfake detection and preventing the spread of fake media. For audio detection, future improvements could include real-time audio monitoring alongside visuals. Our models were trained on a reduced Fake-or-Real dataset with 4,000 audio samples. Better extraction methods and data creation with appropriate dimensions could enhance performance, as computational time for optimal input files was higher than expected.

In combined analysis, the real and deepfake video and audio were swapped to create test data. A more comprehensive dataset with balanced samples of all combinations (deepfake-real, real-deepfake, real-real, deepfake-deepfake) would improve evaluation accuracy and refinement of our multimodal approach.

Acknowledgement We would like to express our deepest gratitude to our mentor, Chaitya Shah [chaitya0623@gmail.com], for his unwavering support and insightful feedback throughout this research. His guidance has been instrumental in refining the methodology and scope of the study, significantly contributing to the development and success of this work.

References

1. A. de Rancourt-Raymond, & N. Smaili (2023). The unethical use of deepfakes. Journal of Financial Crime, 30(4), 1066-1077. Emerald Publishing Limited.
2. Westerlund, M. (2019). The emergence of deepfake technology: A review. Technology Innovation Management Review, 9(11).
3. Hu, J., Liao, X., Liang, J., Zhou, W., & Qin, Z. (2022). Finfer: Frame inference-based deepfake detection for high-visual-quality videos. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(1), 951-959.
4. Ganguly, S., Mohiuddin, S., Malakar, S., Cuevas, E., & Sarkar, R. (2022). Visual attention-based deepfake video forgery detection. Pattern Analysis and Applications, 25(4), 981-992. Springer.
5. Yi, Jiangyan, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. "Audio deepfake detection: A surveyF." arXiv preprint arXiv:2308.14970 (2023).

6. Ismail, A.; Elpeltagy, M.; S.Zaki, M.; Eldahshan, K. A New Deep Learning-Based Methodology for Video Deepfake Detection Using XGBoost. Sensors 2021, 21, 5413.
7. T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," in IEEE Access, vol. 8, pp. 83144-83154, 2020, doi: 10.1109/ACCESS.2020.2988660.
8. Deressa Wodajo and Solomon Atnafu and Zahid Akhtar. Deepfake Video Detection Using Generative Convolutional Vision Transformer.(2023)
9. De Lima, Oscar & Franklin, Sean & Basu, Shreshtha & Karwoski, Blake & George,Annet. (2020). Deepfake Detection using Spatiotemporal Convolutional Networks.
10. Jiameng Pu and Neal Mangaokar and Lauren Kelly and Parantapa Bhattacharyaand Kavya Sundaram and Mobin Javed and Bolun Wang and Bimal Viswanath. Deepfake Videos in the Wild: Analysis and Detection (2021)
11. Montserrat, D.M.; Hao, H.; Yarlagadda, S.K.; Baireddy, S.; Shao, R.; Horváth, J.;Bartusiak, E.; Yang, J.; Guera, D.; Zhu, F. Deepfakes detection with automatic face weighting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 668–669.
12. Nguyen, X.H.; Tran, T.S.; Nguyen, K.D.; Truong, D.T. Learning spatio-temporalfeatures to detect manipulated facial videos created by the deepfake techniques. Forensic Sci. Int. Digit. Investig. 2021, 36, 301108.
13. D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent NeuralNetworks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6, doi: 10.1109/AVSS.2018.8639163.
14. Li, Yuezun & Lyu, Siwei. (2018). Exposing DeepFake Videos By Detecting FaceWarping Artifacts.
15. Afchar, Darius & Nozick, Vincent & Yamagishi, Junichi & Echizen, I..(2018). MesoNet: a Compact Facial Video Forgery Detection Network. 1-7. 10.1109/WIFS.2018.8630761.
16. El-gayar, M.M. & Abouhawwash, Mohamed & Askar, S.S. & Sweidan, Sara. (2024). A novel approach for detecting deep fake videos using graph neural network. Journal of Big Data. 11. 10.1186/s40537-024-00884-y.
17. Govindu, Aditi & Kale, Preeti & Hullur, Aamir & Gurav, Atharva & Godse, Parth.(2023). Deepfake audio detection and justification with Explainable Artificial Intelligence (XAI). 10.21203/rs.3.rs-3444277/v1.
18. Joshi, D., J. Pareek, and P. Ambatkar. "Comparative study of Mfcc and Mel spectrogram for Raga classification using CNN." Indian J Sci Technol 16, no. 11 (2023): 816-822.
19. Massoudi, Massoud, Siddhant Verma, and Riddhima Jain. "Urban sound classification using CNN." In 2021 6th international conference on inventive computation technologies (icict), pp. 583-589. IEEE, 2021.
20. Hamza, Ameer, Abdul Rehman Rehman Javed, Farkhund Iqbal, Natalia Kryvinska, Ahmad S. Almadhor, Zunera Jalil, and Rouba Borghol. "Deepfake audio detection via MFCC features using machine learning." IEEE Access 10 (2022): 134018-134028.
21. Shetty, V. (2024). Python · DEEP-VOICE: DeepFake Voice Recognition, [Private Datasource]. Kaggle. https://www.kaggle.com/code/pyknight73/ my-version-of-deepfake-detection
22. Salvi D, Liu H, Mandelli S, Bestagini P, Zhou W, Zhang W, Tubaro S. A RobustApproach to Multimodal Deepfake Detection. Journal of Imaging. 2023; 9(6):122. https://doi.org/10.3390/jimaging9060122
23. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2020). ProtectingWorld Leaders Against Deep Fakes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
24. Korshunov, P., & Marcel, S. (2018). DeepFakes: a New Threat to Face Recognition?Assessment and Detection. arXiv preprint arXiv:1812.08685.
25. Kaggle. (2020). Deepfake Detection Challenge Dataset. Retrieved from https: //www.kaggle.com/c/deepfake-detection-challenge/data
26. MOHAMMED ABDELDAYEM (2024). The Fake-or-Real (FoR) Dataset (deepfakeaudio) https://www.kaggle. com/datasets/mohammedabdeldayem/ the-fake-or-real-dataset
27. Vimal, B. & Surya, Muthyam & Darshan, & Sridhar, V.S. & Ashok, Asha. (2021). MFCC Based Audio Classification Using Machine Learning. 1-4. 10.1109/ICCCNT51525.2021.9579881.
28. Tolosana, Ruben, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, andJavier Ortega-Garcia. "Deepfakes and beyond: A survey of face manipulation and fake detection." Information Fusion 64 (2020): 131-148.
29. Xu, Bozhi & Liu, Jiarui & Liang, Jifan & Wei, Zhuo & Zhang, Yue. (2021). DeepFake Videos Detection Based on Texture Features. Computers, Materials & Continua. 680. 1375-1388. 10.32604/cmc.2021.016760.
30. X. Yang, Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent HeadPoses," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 8261-8265, doi: 10.1109/ICASSP.2019.8683164.
31. Lutz, Kevin, and Robert Bassett. "Deepfake detection with inconsistent headposes: Reproducibility and analysis." arXiv preprint arXiv:2108.12715 (2021).
32. Shahar, Hadas & Hel-Or, Hagit. (2020). Fake Video Detection Using Facial Color.Color and Imaging Conference. 2020. 175-180. 10.2352/issn.2169-2629.2020.28.27.
33. Zhang, B., Leitner, J., & Thornton, S. (2023). Audio recognition using mel spectogram and Convolution Neural Network. Noise Lab. https://noiselab.ucsd. edu/ECE228_2019/Reports/Report38.pdf
34. Carvalho, S., & Ferreira Gomes, E. (2022). Automatic Classification of BirdSounds: Using MFCC and Mel Spectrogram Features with Deep Learning. https://www.worldscientific.com/doi/pdf/10.1142/S2196888822500300

35. Roberts, Leland. (2020). Understanding the Mel Spectrogram. Medium. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53.
36. SIG Gigue. (n.d.). Mel filter bank visualization. Retrieved August 14, 2024, fromhttps://siggigue.github.io/pyfilterbank/melbank.html