IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

Deepfake and Voice Clone Identity Shield System

Harshitha M R[1], Jisha P[2], Navya K L[3], Nishmitha H P[4], Dhananjaya M[5]

Student, CSE, SJB Institute of Technology, Bengaluru, India[1]

Student, CSE, SJB Institute of Technology, Bengaluru, India[2]

Student, CSE, SJB Institute of Technology, Bengaluru, India[3]

Student, CSE, SJB Institute of Technology, Bengaluru, India[4]

Assistant Professor, CSE, SJB Institute of Technology, Bengaluru, India[5]

Abstract: Synthetic videos and cloned voices can be easily created nowadays, and are being used for impersonation, fraud and spreading misinformation. Manually determining whether video content or audio has been manipulated is not reliable. The Deepfake and Voice Clone Identity Shield System (DFVCISS) is a web-based solution that provides detection of video deepfakes, analysis of voice clones and combined audio/video analysis using a single Flask application. It will provide predictions, confidence and evidence from the model instead of just a category. All analysis will be logged within an SQLite history. A PDF can then be generated using the information stored. The video detection uses a trained convolutional neural network. Inspection of the FakeAVCeleb dataset was performed to determine audio/video examples. This project is a prototype to allow initial verification of media. The reliability of the results depend upon the models being used, training dataset and independent testing.

Keywords: Deepfake Detection; Voice Cloning; Artificial Intelligence; Multimedia Forensics; Convolutional Neural Networks; Risk Assessment.

I. INTRODUCTION

A. Background and Motivation

With the rise of video and audio, these have become an integral part of everyday life. They are used for teaching, reporting the news, online banking and proving identity. However, these have become much easier to falsify. With the advances in deep learning techniques it is now possible for a model to change faces between individuals, alter expressions, match lip movement to synthetic speech or create a voice that sounds like a specific person.

A deepfake is any technique that uses artificial intelligence to create or manipulate media. There are many benign uses of this technology. Some are even useful in entertainment, accessibility, education and content creation. Problems can occur when this technology is misused to impersonate people, conduct fraud, spread misinformation and harm someone's reputation. When combined with synthetic video and voice cloning there is a high risk of showing someone saying things they never actually said.

It can be difficult to determine if media has been manipulated by simply looking at it. The visual clues can often be very subtle and it can be difficult for the average person to determine if the audio has been created. Also, if the media has been compressed or poorly recorded it can hide any clues that there is something wrong. Therefore, using an automated system to verify the media using AI seems like a good idea.

B. Problem Statement

Most tools only provide a single method for verifying if media has been manipulated. This means users must upload to multiple tools to verify if the media has been faked using facial forgery (video) or synthetic speech (audio). Results have to be interpreted and they are all from different sources and will need to be documented by the user.
Having just a simple result of fake/real doesn't provide much information. Most reviewers want to know details about the file, what the model predicted, confidence levels and evidence.

We aim to develop a web based tool which allows detection of video deepfakes, voice clone analysis and joint analysis with a history of results and structured reports to make the process easier to execute and review.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511134

C. Objectives

The aim of this project is to create a tool for detecting deepfake and voice cloning in multimedia content. This will be achieved by: (i) creating a video detection module able to classify the uploaded video using a trained deep learning model. (ii) Creating a page for detecting synthetic or cloned speech in audio files. (iii) Creating a multi-modal detection module for video and audio files. (iv) Displaying the predictions, confidence and evidence in an easy-to-read format. (v) Storing results of the analysis into a database so previous results can be reviewed. (vi) Generate downloadable reports. (vii) Creating a user-friendly website. (viii) Providing a foundation for future improvements such as improved accuracy, transparency and security.

D. Scope and Contributions

This project has focused on creating an integrated system that uses video, audio and multi-modal analysis of web-based media. Features such as evidence presentation, history tracking and report creation are also included. Our key contribution was providing an application level system where users can upload media and obtain a prediction alongside the supporting evidence to store for later.

It should be noted that we have created a prototype integration system. This means thatWhilst modules for detection require individual testing to ensure they perform well in real-world circumstances, this system is not a replacement for expert forensic analysis and should not be the sole criteria for making any legal, financial or identity verification decisions.

II. LITERATURE REVIEW

A. Deepfake Video Detection

FaceForensics++ was introduced by Rössler et al. as a benchmark for detecting manipulations across various types of forgery and compression levels [1]. This highlighted the importance of using standard datasets and the key role visual features play in identifying forgery. A large dataset of high quality deepfake videos known as Celeb-DF was then released by Li et al. to overcome issues of previous datasets [2]. The more complex examples will give researchers a more challenging task to determine if a detector can distinguish between fake and real clips. More generalised surveys have been conducted on generation/detection methods, architectures and datasets [5], [6].

Most approaches focus on extracting spatial and facial anomalies, along with texture irregularities and temporal features. CNNs can be used to extract visual features from images so most video deepfake detectors will take some frames, detect the face within the frame, resize the image to focus on the face and feed into a trained classifier. However, when using a dataset to train a deepfake detector, the performance often decreases when using an unseen dataset. How the face was manipulated, the level of compression, lighting conditions, quality of camera and the appearance of the face along with the overall dataset distribution will affect how well a detector performs.

B. Voice Cloning and Synthetic Speech Detection

Voice clone technology can duplicate a speakers voice characteristics. Despite having legitimate applications it can also allow someone to impersonate another person. Detecting cloned voices can be challenging as the generated audio can be very similar to the original especially when short or high quality. ASVspoof have provided datasets and protocols for detecting spoofed audio including synthetic audio, voice converted audio and replayed audio. This has shown that different attacks and recording conditions affect performance [3].

Audio detectors generally focus on acoustic, spectral, speech patterns or neural network features. The reliability will depend on what training data it has been trained on, what type of synthetic audio it has been trained on and the conditions it encounters during testing. In this case, the voice clone page must be read correctly based on how the models and logic is set up. Unless an audio classifier is trained and tested to do so it cannot be considered a validated voice clone detector.

C. Multimodal Deepfake Detection

There is a wide variety of ways to manipulate videos and audio. These could be just video with real audio, just audio with a fake video or both audio and video being manipulated. For this reason, detecting just one modality would not detect all the problems. FakeAVCeleb was created by Khalid et a. to provide a resource for further research. This includes real/fake video and real/fake audio [4].

A multi-modal approach can be taken to look at the output of the video and audio detectors. Depending on how this is designed, it can also check that the audio matches what is being said with the movement of the mouth. This will involve specific audio video processing and can only be claimed if it has been created and tested. However there are many challenges which come with multi-modal approaches. This can include poor quality in one of the modalities, time delays, missing audio and discrepancies between the models. A reliable system must explain how these scenarios are dealt with.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

D. Research Gaps and Limitations

Multiple issues can be identified from this literature. Firstly, models may have generalisation issues when applied to instances they have not previously encountered. Secondly, models may not be very reliable when applied to diverse datasets due to a lack of variation in terms of people, faces, scenes and recording conditions. Thirdly, separate video and audio analysis tools make it difficult to compare results. Fourthly, there is no reasoning behind the label. Fifthly, analysis logs should be kept so the relationship between the uploaded file, model output, analysis time and report can be tracked. Lastly, confidence scores and labels can be easily misinterpreted as showing genuine or manipulated content.

The application issues have been addressed by having separate analysis modules, clear organisation of results, information on evidence, keeping an analysis history and downloadable reports. However, fundamental issues with AI based detection still remain. Further experimentation would be needed to understand how well the system will perform on unknown or challenging examples.

E. Summary of Reviewed Resources

In order to keep the review of work covered, Table I provides a summary of the datasets and benchmarks mentioned above and how they relate to the project. It can be seen that these have been narrowed down based on information found within the literature [1]-[4].

TABLE I DATASETS & BENCHMARKS WITHIN DFVCISS PROJECT

Resource	Modality	Focus	Relates to DFVCISS
FaceForensics++ [1]	Video (face)	Multiple methods with various compression levels	Reference benchmark for face forgery
Celeb-DF [2]	Video	Large set of high-quality deepfake videos	Challenging test cases for video detection
ASVspoof [3]	Audio	Datasets and protocols for synthetic speech, converted and replay attacks	Provide guidelines for audio detection
FakeAVCeleb [4]	Audio/video	Real/fake audio-video combinations	Dataset looked into for experimentation

Overall it can be seen that the video resources focus on face manipulation and the audio resource focuses on spoofed speech. The FakeAVCeleb dataset is the only one that includes both types of data. This means that multiple sets of guidelines need to be used when building a system to detect multiple types of manipulation. Care needs to be taken to ensure that a model trained with one type of data can be applied to the other.

The gaps identified in the previous subsection can be compared with design decisions made within this project. Table II outlines these, and the final column is honest about what is still to be carried out within the prototype.

TABLE II RESEARCH GAPS & HOW THEY HAVE BEEN ADDRESSED IN DFVCISS

Gap	Design Decision in DFVCISS	What is still to be done
Generalisation	Separate analysis modules so a new model can be used	Test on unseen manipulation methods
Multiple dataset	FakeAVCeleb used as a source of audio-video data	Test on additional diverse datasets
Individual modality analysis	Single interface for video, voice-clone and multimodal analysis	Documented method of combining information.
Interpretability	Evidence details shown alongside prediction	Experimental validation of explanations.
Traceable evidence	History related to evidence and reports	Hashes and model versions where not yet recorded.
Uncertainty	Confident shown in outputs from model with warnings	Confidence calibrated.

III. PROPOSED SYSTEM AND METHODOLOGY

A. System Architecture

The DFVCISS analysis tool is web-based. It uses the Flask framework to manage the Web requests, connect the front-end to the analysis tools and display outputs within the browser. The analysis data and evidence are stored using SQLite.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

The tool has eight components: the front-end, which allows the user to access the dashboard, video, voice clone and multi-modal modules; the upload and validation module, which validates files against the applications handling criteria; the video detection module; the voice clone analysis module; the multi-modal module; the output module; the database module; and the report generation module.

An example flow of the system would be, the user selects an analysis module then uploads a supported file to the application. The application validates the request and performs the relevant analysis. It gathers the output, stores it and displays the output page. A report can also be requested, which would retrieve the stored data and generate a structured PDF. An overview of the system can be seen in Fig. 1.

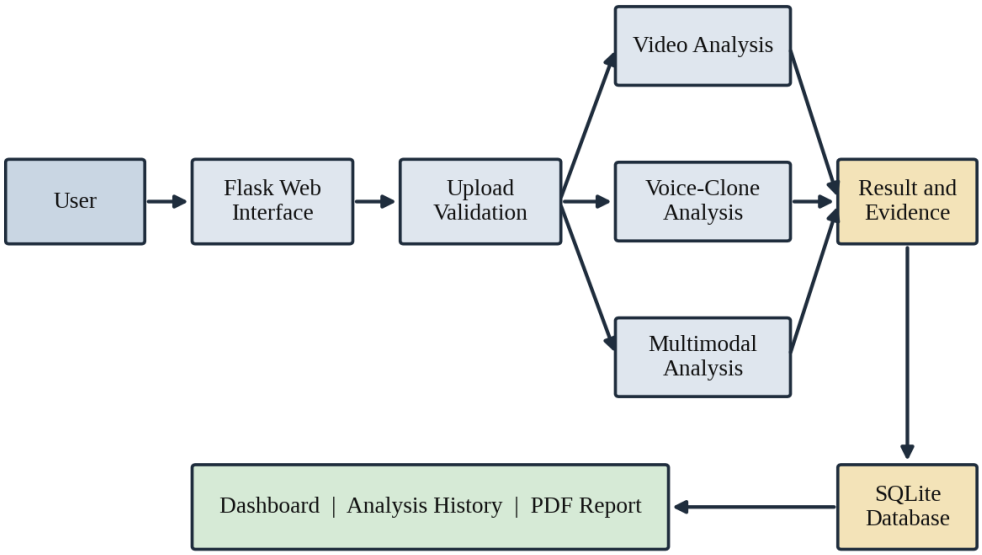


Fig. 1 Proposed architecture of the DFVCISS system

The role of each component is given in table III. This clearly shows how each component provides a well defined output to the following component. This makes it possible to use the same uploading, storing and reporting functionality for both the video, audio and multimodal paths.

TABLE III MAJOR COMPONENTS OF DFVCISS

Component	Responsibility	Output
User interface	Dashboard and pages for video, voice-clone and multimodal analysis	Upload forms and result views
Upload and validation layer	Receives the media file and checks it against file-handling requirements	Accepted or rejected file
Video detection module	Frame selection, face preprocessing and CNN prediction	Classification and prediction information
Voice-clone module	Preprocesses audio and runs the configured audio-analysis function	Classification with any confidence or supporting data
Multimodal module	Combined workflow for video and audio	Modality-specific or combined results
Result presentation layer	Displays prediction, confidence and available evidence	Result page
Database layer	Stores analysis metadata and evidence in SQLite	Retrievable records
Report generator	Builds a structured PDF from a saved record	Downloadable report

There are three design principles. Firstly, the modular nature of the application means that an individual detection technique can be refined or changed without affecting the other parts of the system. Secondly, there is full traceability between the uploaded files, the output of the model, date/time of analysis and the produced report. Thirdly, there is careful wording in presenting the results of the process. The results are presented as outputs from the model and any additional information. They are not presented as a definite determination on the authenticity of the file.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

B. Dataset Description and Preparation

Selecting the correct data is an important step in creating our media authenticity detection system. The data used in this work was provided by the FakeAVCeleb dataset. This provides a combination of real and fake videos with real and fake audio, this is useful for multi-modal work [4]. The data available locally is made up of a CSV file containing metadata about the media. It also contains sub-folders containing the media, with the CSV relating each media location to its label and category.

Looking at the metadata shows us what files we have available and how they are labelled. We have 21,566 records in total with 21,560 MP4 files verified. There are very few records that do not match a file. These will need to be looked into before using the data for our final experiments. The number of records for each category can be seen in table IV. This only shows us what was in the metadata we looked at. Not what was actually used in training and testing. We will need to note what records were used after pre-processing, excluding and duplicates.

TABLE IV FAKEAVCELEB METADATA CATEGORIES INSPECTED

Dataset category	Number of records
FakeVideo-FakeAudio	10,857
FakeVideo-RealAudio	9,709
RealVideo-RealAudio	500
RealVideo-FakeAudio	500
Total records inspected	21,566

The video preprocessing consists of selecting frames from each clip, detecting the faces in the frames, resizing the faces and creating appropriate images for the classifier. Audio preprocessing will depend on what the model expects (format, sampling rate, length, features etc.). The specifics of how frames are selected, faces detected and audio processed will depend on what is implemented in the code. This will be documented once the code has been confirmed.

It is very important to ensure that the training/test data is split properly. If there are multiple clips from the same source video/speaker, samples should not be too similar. Therefore, samples shouldn't be from the same source video/speaker. During the process of converting the metadata into an experiment, we generally follow a simple process. First, we match our metadata to the actual media files we have. This is why there is a slight difference between the number of records (21,566) and files (21,560). Then duplicates are removed. Finally the samples are divided into training, validation and test sets. The test set will be kept aside and only used for the final metric. We also ensure that clips from the same source video/speaker aren't too similar by grouping them together. We will state how many samples are in each set and what was excluded. This will make it easier to justify the final numbers.

C. Deepfake Video Detection Module

The video module takes the uploaded clip and passes it to the video prediction function. Within the video prediction function, select frames are passed to the face detector function. This identifies the faces within each image and resizes them to the required input shape. Once that has been done, the trained model (deepfake_model.keras) is used to make the prediction.

The video module will first load and validate the uploaded video before passing it to the analysis function. Then, select frames from the video will be passed to the face detection function. This will identify and prepare the facial regions in the frames. The faces will be resized to fit the input shape of the model. Then, the faces will be passed to the classifier. This will collate the predictions from each frame using the logic included within the implementation. Finally, the video will return the prediction with the relevant prediction information to the Flask application. This will save and display the results to the results page. A summary of this process can be seen in fig 2.

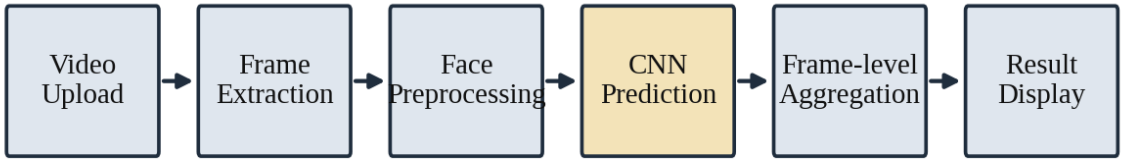


Fig. 2 Video deepfake detection workflow

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

Currently, the prototype uses a convolutional neural network (CNN) to make a prediction from an image. The convolutional layer will detect patterns from the pixel data. These patterns will then be translated into a prediction by the subsequent layers. In regards to detecting deepfakes, the patterns learnt by the model could be visual characteristics common in manipulated faces. The final prediction will be shown along with some confidence information. However, these values are the output of the model and are not necessarily probabilities of the file being real/fake. The model will need to be calibrated to determine how accurate the confidence is.

D. Voice Clone Detection Module

The voice clone module allows audio recordings to be analysed. This will determine if there are patterns within the recorded audio that are associated with the categories that the audio model has been configured for. Audio models for detecting synthetic speech will typically have some form of preprocessing applied to the audio, extracting either acoustic or learned features. How the model handles the input will depend on what type of model is being used. Therefore, the voice prediction function will need to have the processing done to the input, the model architecture, the data it was trained on and how the raw output maps to the label shown.

Once an audio file has been uploaded to the voice clone page, the audio will be validated to ensure it can be processed. Then, the audio will be converted/preprocessed to match what the audio model requires. This will be passed to the analysis function that has been configured. The output will then be saved and shown to the user. The benefit of this is that the audio validation is part of the same application as the video validation. However, how well this works will need to be measured with labelled real and synthetic audio.

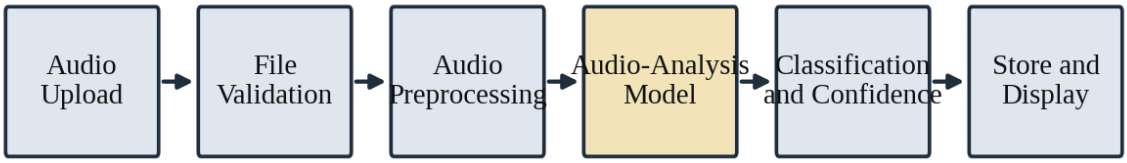


Fig. 3 Voice-clone analysis workflow

Although this seems like a straightforward process (fig 3), most of the technical aspects are in the middle. Whatever the audio model requires (format, sample rate, length, features etc.) must be reliably produced. Therefore, once the implementation has been confirmed, the processing done to the input, model architecture, training data and raw output to label mapping must be noted.

A voice that sounds unusual or unfamiliar should not automatically be classed as a clone. Real audio will have noise, compression, reverb, background speech and microphone artefacts which can affect the model. Therefore, the output should be seen as an indicator rather than confirmation of synthetic speech.

E. Multimodal Detection Module

This module is useful if you need to carry out a multi-modal analysis. It will make use of the video and audio analysis functions. Any combination rules should be explained here including handling of missing data, discrepancies between the predictions, and differing confidence levels. Once a supported file containing video (and audio if available) has been provided, the video and audio media can be accessed. The video can be analysed using the video module and the audio using the voice clone function. These can then be combined or displayed depending on what combination rules have been implemented. Evidence should also be displayed.

This could be useful if one media appears to be authentic while the other appears suspicious. An obvious example would be video that has been recorded but audio that has been synthetically generated. However, the accuracy of this approach will not necessarily be greater than just analysing each media individually. This must be proven by comparing predictions made by analysing video only, audio only and multi-modal. The current implementation just carries out each analysis and provides the results. This should be described as an integrated multimodal analysis rather than a trained fusion model. Only if a fusion mechanism has been learnt, or a sync detection method, can this be claimed.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

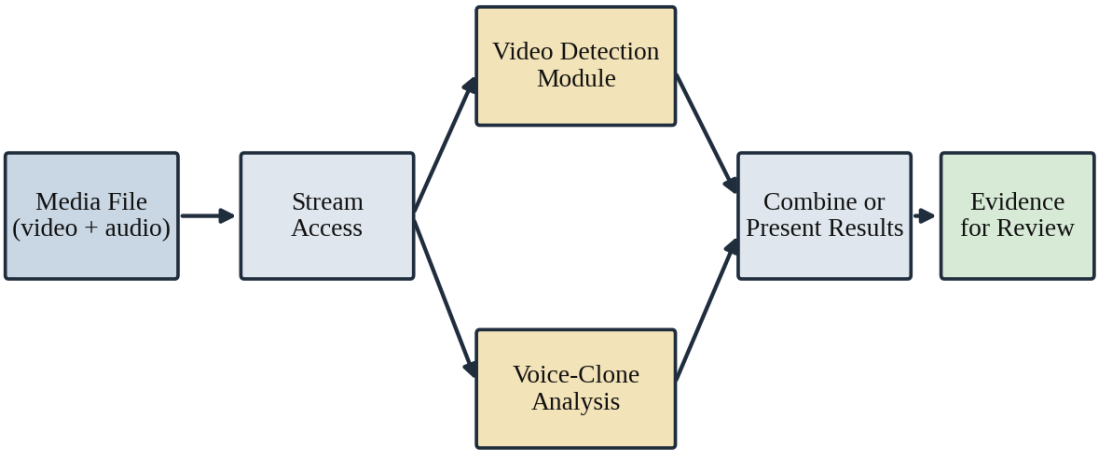


Fig. 4 Multimodal analysis workflow

As illustrated by Fig. 4 the two analyses are performed independently and only fused when presenting results. Since they both may give conflicting results, it is useful to have a common method to interpret the results. One example of this is given by Table V.

TABLE V READING COMBINED VIDEO AND AUDIO OUTPUTS

Video result	Audio result	How to read it
Genuine	Genuine	Neither module flagged the file; concern is lower, but this is not proof of authenticity
Flagged	Genuine	Visual manipulation is indicated while the audio appears unaltered
Genuine	Flagged	Audio manipulation is indicated; the video alone would have missed it
Flagged	Flagged	Both streams raise concern; this is the strongest indication but still needs human review
Available	Missing	Only the visual stream can be assessed, so the audio condition is unknown
Low confidence	Low confidence	Treat as inconclusive and verify through other means

The table also illustrates the need to decide how to handle missing audio and conflicting predictions. If there is no audio for a clip then the module has nothing to say about voice cloning. We do not want to assume silence means everything is okay. Similarly if the two modules disagree we would like to know this. It would be useful to show this in the results.

F. Evidence Extraction and Risk Assessment

DFVCISS also contains an evidence section to allow for easier examination of results. Depending on what data is returned by the prediction stage, this will contain information about the file, amount of frames sampled/analysed, frame by frame predictions, and raw model output amongst other things. A separate evidence area is stored so it can be accessed when opening a result/report. This keeps the data in a structured format and allows for additional details to be added by the report generator if available.

A useful piece of evidence will have the following information: analysis ID, time of analysis, filename uploaded, media type, prediction, confidence returned, frames sampled/analysed, frame by frame predictions, model ID or version (if known), and file integrity information such as SHA-256 hash (if calculated). All of these may not be present depending on how the current implementation works and should only be shown as measured evidence if they are actually being calculated and stored by the application.

Risk assessment within DFVCISS means presenting the data from the prediction stage in such a way that the user can easily see how much concern the file should raise. The high score from the model does not equate to a real world probability of fraud. If a risk score is to be created then how it is calculated needs to be defined. For example the fake class score can be mapped to categories using specific thresholds. Until this has been calibrated and validated using representative data this is just a rule. Evidence can be useful for traceability but having model generated evidence does not prove that the file has been tampered with. Therefore the measured metadata, model predictions and human interpretations are kept separate.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

A typical evidence field and what they represent are shown in Table VI. As mentioned above any fields should only be shown as measured evidence if they are confirmed to be calculated and stored by the application.

TABLE VI FIELDS OF AN EVIDENCE RECORD

Field	Purpose
Analysis identifier and timestamp	Links the history record, the evidence and the report
Uploaded filename and media type	Identifies exactly which file was analysed
Final model prediction	Headline output of the analysis
Confidence information	Model-reported value attached to the prediction
Frames sampled and analysed	Shows how much of a video was actually examined
Frame-level predictions	Shows how the prediction varied across frames
Model identifier or version	Ties the result to a specific model
File hash (SHA-256)	Allows file integrity to be checked later

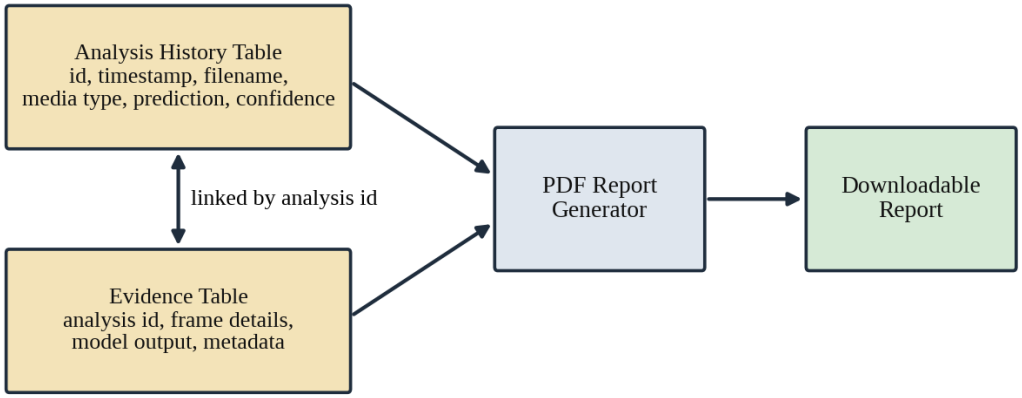


Fig. 5 Relationship between stored records and the generated report

The structure shown in Fig. 5 allows for the history table to remain small whilst allowing the evidence table to expand with the more detailed analysis results. When creating a report these tables are viewed based on their common identifier for analysis. This ensures that reports reflect the data that was stored during analysis.

G. Web Application and Database Integration

The web application has been created using Python and the Flask framework. Routes have been created which link the interface to the correct analysis functions. A homepage, login and sign up page have been created alongside a dashboard to be the main starting point for the application. There are separate pages for video detection, voice clone analysis and multimodal analysis. The main interface and reporting can be seen in Figs. 6-11.

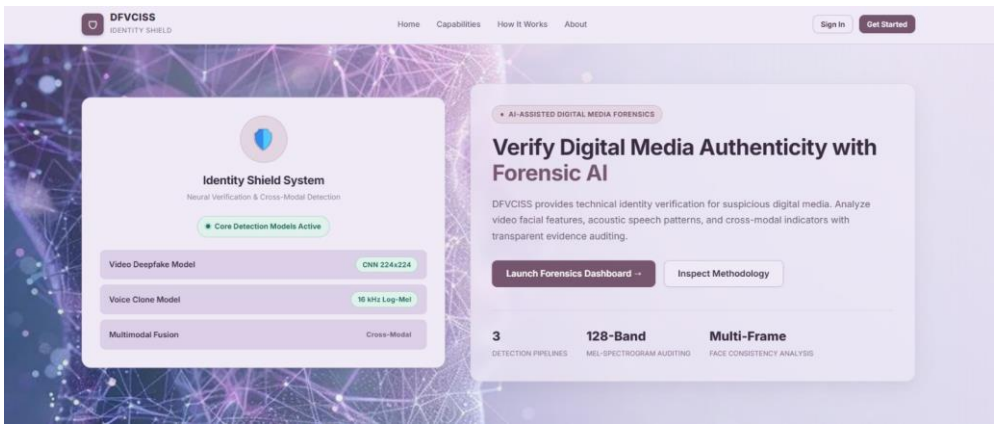


Fig. 6 DFVCISS landing page and overview of the analysis modules

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

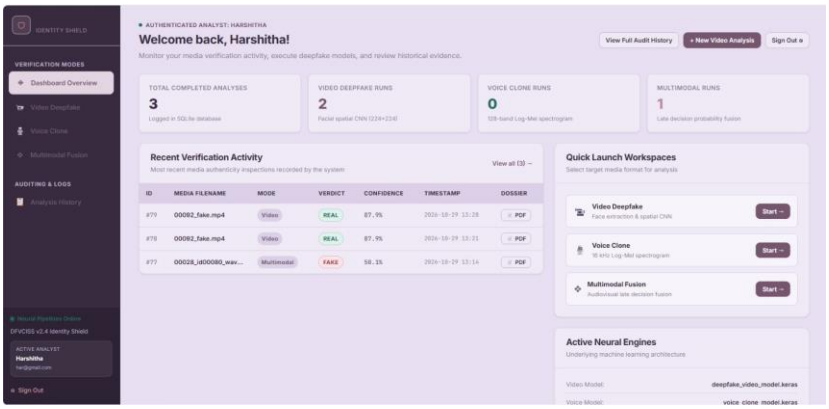


Fig. 7 Analyst dashboard and recent verification activity

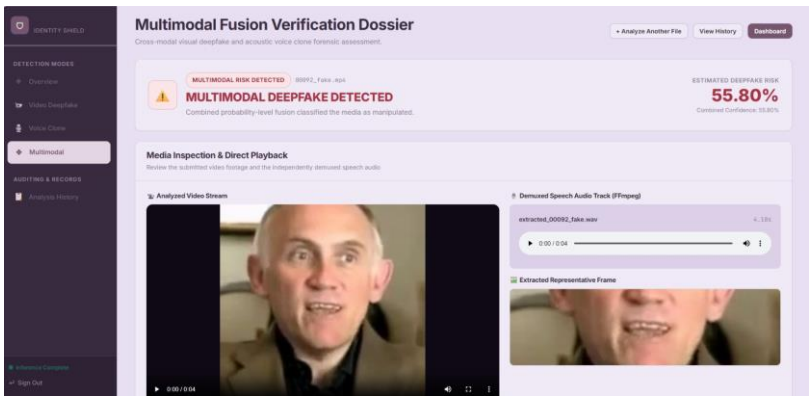


Fig. 8 Multimodal analysis result with the video and extracted audio streams

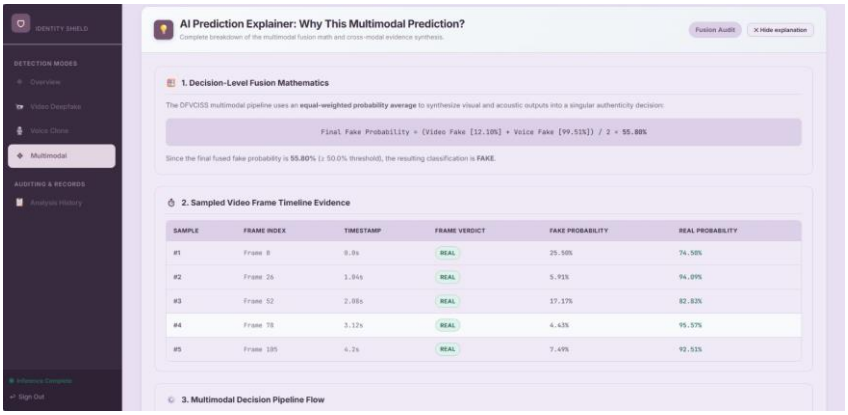


Fig. 9 Prediction explainer with fusion calculation and sampled-frame evidence

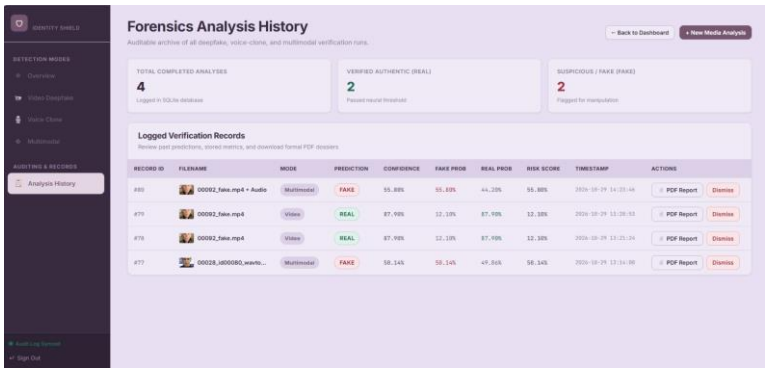


Fig. 10 Forensics analysis history with saved records and report actions

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

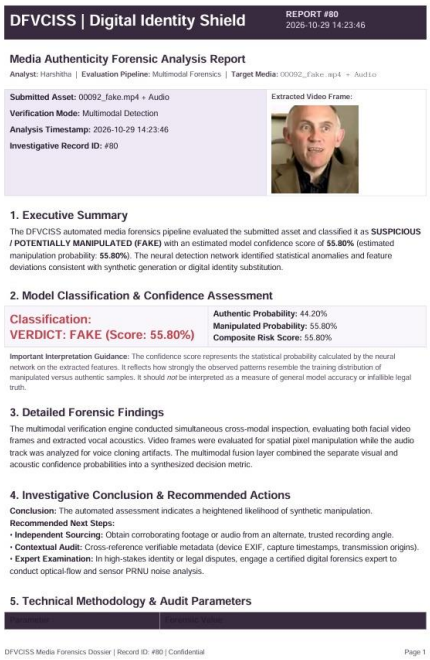


Fig. 11 Example PDF forensic report and model assessment

Analysis history and evidence will be stored in SQLite. The main data will be stored in the analysis history table. Additional data associated with the analysis ID will be stored in the evidence table. When an analysis has been completed, the results are saved to be accessed from the history page. The PDF generator uses the selected record and associated evidence to create a downloadable report. Having the presentation, analysis and persistence layers separated made it easy to add additional detection types and reporting types.

Additional features will be required in production. This includes secure authentication, secure upload handling, restrictions on file size and malware scanning (where applicable). Secure storage will be required. There will also be access controls on the database. All uploads will be carefully managed. If these features have been included they should be identified.

The software and hardware environment used during development is outlined in Table VII.
To give a good example of how the system works as a whole, let's look at one full analysis. The user will log into the application and navigate to the dashboard. They will select the page that corresponds to their file type (video, voice clone or multi-modal). Once the file has been uploaded and validated, the appropriate analysis function will execute and return the results. The application will save the prediction and evidence to SQLite and display the results page. The user can view the label, confidence information and evidence available. The record can be found in the analysis history and exported as a PDF. The user does not need to re-upload the file at any point or manually copy over results.

TABLE VII DEVELOPMENT ENVIRONMENT AND COMPONENTS

Item	Detail
Language and web framework	Python 3.11.x with Flask 3.0.x
Database	SQLite 3 (dfvciss.db); user-isolated history and evidence tables
Video model	Four-block 2D CNN; deepfake_video_model.keras; input 224 × 224 × 3
Audio model	DFVCISS-AudioCNN log-Mel spectrogram 2D-CNN; voice_clone_model.keras
Dataset	FakeAVCeleb v1.2; 21,566 metadata records inspected; 3,284 multimodal test clips reported
Report format	Forensic dossier PDF generated with ReportLab; SHA-256 digest reported
Development machine	Windows 11; Intel Core i5-1235U; 16 GB RAM; Intel Iris Xe graphics
ML and vision libraries	TensorFlow 2.15/2.16; OpenCV 4.x



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511134

H. Evaluation Metrics and Protocol

We want to assess two aspects of DFVCISS: the functionality of the software and the accuracy of our models. Accuracy is measured using traditional classification metrics. We consider manipulated files to belong to the positive class. True positives are when we have a manipulated file that we correctly identify as being manipulated. Similarly, true negatives are when we have a genuine file and we correctly identify it as being genuine. False positives are when we have a genuine file and incorrectly identify it as being manipulated. Finally, false negatives are when we have a manipulated file but fail to identify it. Using these values we can calculate:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1\text{-}score = 2 \times Precision \times Recall / (Precision + Recall)$$

The overall accuracy can be misleading as we may get a high accuracy whilst performing poorly on one class. This can be a problem given there is a large imbalance in the number of samples per category (as seen in Table IV). Precision tells us how often we correctly identify a manipulated file. Recall tells us how many manipulated files we are able to identify. F1-score provides a balance between these two metrics. For the audio module, common additional metrics are the equal error rate. This shows at what operating point we are getting equal numbers of false acceptances and false rejections. [3]. For the multimodal module, we report the results for all 4 combinations of real/fake. We compare this with the video only and audio only baseline.

In addition, we should look to see if the confidence values match the actual accuracy on the test set. This will allow us to determine if our model is well calibrated. We should also look to see how our model performs under different conditions during recording. For example, we may have a model that works well on clean recordings, but performs differently when the files have been compressed, resized or re-encoded.

I. Responsible Use

Using the tool to label peoples recordings as fake or real could have real consequences. As such DFVCISS is a decision support tool. The output should not be used as the sole reason to make decisions around legal, financial or identity checks. The predictions and confidence values cannot be taken as evidence and should be weighed in conjunction with other data surrounding the file and source.

Uploading media could contain sensitive data. Videos and voice recordings may include identifiable people. If we were to use this tool more widely there would need to be clear policies in place to determine who can view the file, how long it will be stored for and how to protect it. This is the same as the production use case and should be considered when designing the tool.

IV. RESULTS AND DISCUSSION

A. Experimental Setup

DFVCISS was developed using Python 3.11.x (Windows 11, Intel Core i5-1235U processor, 16GB RAM, Intel Iris Xe graphics). The software used within the environment is Flask 3.0.x, TensorFlow 2.15/2.16 and OpenCV 4.x. The video detection model consists of a 2D CNN with 4 blocks, named ‘deepfake_video_model.keras’. This model accepts 224x224x3 values from facial frames. The audio detection model is a CNN named DFVCISS-AudioCNN, named ‘voice_clone_model.keras’. Analysis history and evidence is stored using SQLite. Analysis of the video model has been performed using the ReportLab library to generate a forensic PDF report including the SHA-256 hash.

An analysis of the metadata within FakeAVCeleb was performed. This consisted of a total of 21,566 records being analysed. A total of 21,560 MP4 files were identified. During testing of the video model, an independent dataset of 3,024 clips (14,168 frames) were used. A balanced audio dataset consisting of 800 recordings was used. A multi-modal dataset consisting of 3,284 clips was used. Full results can be seen in Tables VII-X below. Software specifications can also be seen below.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

B. Video Detection Results

On the reported independently tested 'FakeAVCeleb' section of data, the video classifier predicted 3024 videos, and 14168 facial frames. On the video level, it had an accuracy of 96.30%. It demonstrated a high precision score of 99.82% and a high recall score of 96.28%. The F1 score was 0.9802 and its ROC-AUC score was 0.9957. On the video level, there were 5 false positives for the 144 real videos. The video classifier achieved an accuracy of 98.03% on the frame level. It had a high precision score of 99.70%, a high recall score of 98.24%, an F1 score of 0.9896 and a ROC-AUC score of 0.9885. These results can be seen in table VIII below.

These results show that the video classifier has a high precision at both the video and frame level. However, there are still errors within the results due to there being some false positives and false negatives. This means that these results should be taken as what the model predicted for this section of the test data.

TABLE VIII VIDEO DETECTION METRICS

Metric	Video-level result
Test samples	3,024 videos; 14,168 frames
True positives (TP)	2,773
True negatives (TN)	139
False positives (FP)	5
False negatives (FN)	107
Accuracy	96.30%
Precision	99.82%
Recall (sensitivity)	96.28%
F1-score	0.9802
ROC-AUC	0.9957
Frame-level accuracy / precision / recall / F1 / ROC-AUC	98.03% / 99.70% / 98.24% / 0.9896 / 0.9885

In practice, there are different costs associated with false positives and false negatives. False positives mean that a real recording has been identified as being fake, which could cause unfair suspicion towards a real individual or event. False negatives are where a manipulated recording passes through. This does not serve any purpose. By reporting both false positives and false negatives we can understand which type of classification is more important to us.

C. Voice Clone Detection Results

The voice clone module was tested with 800 audio recordings (400 real and 400 synthetic/cloned). This resulted in an accuracy of 95.25%, precision of 91.51%, recall of 99.75% and F1-score of 0.9545. It accurately classified 399 synthetic samples but failed to classify 1. In terms of real recordings it flagged 37. The equal error rate was 1.87% (0.0187). Table IX below shows all the results.

This gives an overview of how well the module performed in testing. However, it has yet to be seen how accurate it would be in real situations such as different synthesis methods and recording conditions.

TABLE IX VOICE-CLONE DETECTION METRICS

Metric	Measured result
Genuine audio test samples	400
Synthetic audio test samples	400
Correct genuine classifications (TN)	363
Correct synthetic classifications (TP)	399
False alarms (FP)	37
Missed synthetic samples (FN)	1
Accuracy	95.25%
Precision	91.51%
Recall	99.75%
F1-score	0.9545
Equal error rate (EER)	1.87% (0.0187)

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

D. Multimodal Analysis Results

The multimodal module was tested with 3284 clips. It was able to accurately classify 73 out of 75 real video/real audio clips, 1445 out of 1501 fake video/real audio clips, 75 out of 75 real video/fake audio clips and 1615 out of 1633 fake video/fake audio clips. This gave an overall accuracy of 3208 (97.69%). As the data is imbalanced it demonstrates that both modalities are important. The system was able to detect all synthetic audio in real video/fake audio clips (75 out of 75). Table X below shows how many were detected per category and what percentage. However, this doesn't necessarily show that the fusion is better than the individual modalities.

TABLE X MULTIMODAL EVALUATION CATEGORIES

Evaluation category	Test count	Correct classifications and reported result
Real video and real audio	75	73 (97.33%); concordant real
Fake video and real audio	1,501	1,445 (96.27%); discordant streams
Real video and fake audio	75	75 (100.00%); discordant streams
Fake video and fake audio	1,633	1,615 (98.90%); concordant fake
Overall multimodal set	3,284	3,208 (97.69%)

E. Web Application and Reporting

The functionality of this application was tested. The home page, authentication, uploading and analysing video/audio, multimodal analysis, evidence, Grad-CAM, search history, exporting to PDF and using SQLite have all been confirmed. Table XI below shows the results.

These results are taken from the summary of the project. They show what the results of the functionality tests were.

TABLE XI FUNCTIONAL TESTING CHECKLIST

Feature	Expected outcome	Test status
Home page	Opens the main application page	Verified
Login and signup	Handles the implemented authentication workflow	Verified
Video upload	Accepts supported video input	Verified
Video analysis	Returns and displays the model prediction	Verified
Voice-clone analysis	Processes supported audio input	Verified
Multimodal analysis	Runs the implemented combined workflow	Verified
Evidence details	Displays evidence fields returned by the analysis	Verified
Analysis history	Retrieves previously saved records	Verified
PDF report	Generates a report for a selected record	Verified
Database persistence	Retains and retrieves the saved analysis	Verified

F. Limitations

Limitations do exist. Detection methods may not work for all different types of deepfake creation methods or text-to-speech methods. Additionally, factors such as compression, background noise, lighting and other conditions during the recording of the audio file may affect the prediction of the algorithm. A file may be categorised differently depending upon if it has been compressed, the size or recording conditions. There will always be some false positives and negatives. It is also important to consider how much computing resources are required. The accuracy of results will vary depending on the training data. While showing evidence of the evidence will improve traceability, it does not guarantee the decision is correct. Confidence scores should not be considered calibrated probabilities unless there have been calibration experiments. A multimodal approach may help provide measurable improvements if the fusion methodology has been properly designed and tested. Future testing should be done using independent data sets, ensuring the training and test samples are well separated. Error analysis and comparison between single and fused modalities should be performed.

The main limitations, along with how they are likely to affect the results and how they can be mitigated are shown below in Table XII.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511134

TABLE XII LIMITATIONS AND MITIGATION

Limitation	Effect on results	Mitigation
Unseen manipulation methods	Accuracy may drop on new kinds of fakes	Test on independent datasets and unseen methods
Recording conditions	Compression, noise or lighting can change predictions	Evaluate across varied conditions
False positives and negatives	Genuine files flagged or fakes missed	Report precision, recall and an error analysis
Confidence interpretation	Scores misread as probabilities	Calibration experiments; present as model output
Multimodal design	No benefit unless the combination is designed and tested	Compare video-only, audio-only and combined results
Computational cost	Slower handling of larger files	Measure processing time and optimise
Dataset dependence	Results tied to the training data	Use diverse, carefully separated datasets

G. Discussion

We see high performance across the test splits for our task. We achieved a video level accuracy of 96.30%, a voice level accuracy of 95.25% and an overall multimodal accuracy of 97.69%. The performance on the multimodal task is particular important for the asymmetric manipulation scenario. Due to the different number of samples for video and voice compared to multimodal, we cannot combine these numbers. However, this is dependant on the datasets, labelling, models and testing methods used. For more general conclusions further independent tests would be required.

The real value of our DFVCISS system comes from the process surrounding our models. The user can upload a clip, receive a prediction (with evidence) and then refer back to the record. This has real advantages for any detection model. As better data and methods become available, we can improve our models.

H. Intended Use Scenarios

Given the context of this project (communication, education, banking, journalism, social media and online identity verification), there are many situations where such a tool could prove useful. For example, using this tool, an organisation could get an initial assessment and record of any suspicious video or audio messages they receive. This would help a newsroom or platform identify clips in need of more detailed analysis, or a user could verify a recording which appears to be out of character for them.

This system would act as an initial screening tool. If the results return concerning findings then this should be verified. If the results return reassuring findings then this should not be considered a guarantee. When high stakes are involved expert forensic analysis should be conducted and this tool is designed to support this.

V. CONCLUSION AND FUTURE WORK

This paper describes a web application for video deepfake detection, voice clone detection and multi-modal analysis, with an integrated history, evidence and PDF report. A summary of the tests performed is provided below. The video, audio, and multi-modal models were tested on different subsets of the data. The main parts of the application were also tested.

The results suggest good performance on the different partitions. However, there were errors found and generalisation is not proven. This should therefore be considered as a prototype for triage and decision support. Future work should be done to ensure the data splits and model configurations can be replicated. The metrics should also be replicated independently. Finally, the model should be tested on a wider range of unknown manipulation techniques. The multi-modal model decisions should also be compared to unimodal baselines. Only after this can it be used in high stakes scenarios.

There are several options for how to improve the application in the future. These are summarised in Table XIII. Improvements could be made to the models, evidence, and usability.

TABLE XIII PLANNED ENHANCEMENTS

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511134

Enhancement	Description
Model generalisation	Train and evaluate on diverse data and on manipulation methods not seen before
Advanced audio analysis	Integrate and validate a suitable synthetic-speech detector under recognised protocols
Multimodal fusion	Develop a documented fusion strategy and compare it with single-modality baselines
Temporal consistency	Investigate audio-video synchronisation and mouth-movement consistency where processing allows
Explainability	Add visual or acoustic explanations only if supported by the model and validated
Evidence integrity	Record file hashes, model versions, processing timestamps and relevant metadata
Security	Strengthen access control, upload validation, protected storage and retention policies
Performance	Measure processing time and resource use, then optimise for larger files
Chatbot assistance	Explain the workflow and report fields and advise on safe handling; never certify a file by itself
Independent evaluation	Test on additional datasets and realistic conditions before use in sensitive settings

These steps will help to evolve DFVCISS from being a theoretical concept to a reliable tool for multimedia analysis with verifiable claims.

ACKNOWLEDGMENT

We would like to thank the Department of Computer Science and Engineering at SJB Institute of Technology, Bengaluru, for providing us with the academic environment to complete this project. We would also like to thank all the researchers and data creators who have published their work which has helped us with our research on multimedia forensics and deep fake detection.

REFERENCES

[1] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 1–11, doi: 10.1109/ICCV.2019.00009.

[2] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3207–3216, doi: 10.1109/CVPR42600.2020.00327.

[3] X. Wang et al., "ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Trans. Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021, doi: 10.1109/TBIOM.2021.3059479.

[4] H. Khalid, S. Tariq, M. Kim and S. S. Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," arXiv preprint arXiv:2108.05080, 2021. [Online]. Available: https://arxiv.org/abs/2108.05080

[5] Y. Patel et al., "Deepfake generation and detection: Case study and challenges," *IEEE Access*, vol. 11, pp. 143296–143323, 2023, doi: 10.1109/ACCESS.2023.3342107.

[6] [6] M. Masood, M. Nawaz, K. M. Malik, A. Javed and A. Irtaza, "Deepfakes Generation and Detection: State-of-the-Art, Open Challenges, Countermeasures, and a Way Forward," IEEE Access, vol. 11, pp. 37424–37463, 2023, doi: 10.1109/ACCESS.2023.3266718.

 This work is licensed under a Creative Commons Attribution 4.0 International License