Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Deepfake Verifier: Detection of AI-Generated and Manipulated Images and Videos

Prof. Meghana A. Deshmukh , Sahil R. Pagde , Sakshi T. Surjuse , Purva N. Adsod, Prasad S. Pranjale, Arjun S. Deshmukh

Prof. Ram Meghe Institute of Technology and Research, Amravati, Maharashtra, India.

Abstract : - The growth of the technology and development in the ai models are making it easy to generate deepfake videos, images and audios. To give an example in 2023 a worker in Hongkong was fooled using video call which had the deepfake video and was looted for 39 million us Dollars. Studies and reports shows that 77% to 80% of the work to develop and to find these deepfake videos is done using deep learning. Mostly technologies used are FaceForensics+++, CELL-DF and DFDC. Also two largely used Ai models are CNN and LSTM. Currently technologies and models used to detect these Deepfakes , do not work well on newly generated and unseen data. Peoples also test them in multiple ways, so the results are hard to compare. These models just compare them as real and fake. In this paper , we propose a deepfake image detection model named as *Deepshield*. This model detects fake photos and also explain the reason for it's choice. The simple working of the model is as follows : the user uploads an image to the model. This image is taken as an input data. The model performs different steps like validation, preprocessing on this input image and then pass it to the trained model. The trained model makes prediction such as real or fake and also explain why it made this particular choice. It compare the features of the face in the image such as skin tone, textures, alignment and light etc. with the training data. The output generated by the model is explained by the Grad-CAM and the result is displayed to the user.

Keywords: Deepfake detection, explainable AI, Grad-CAM, CNN-LSTM, confidence scoring, media forensics.

INTRODUCTION :

The new developments in Artificial Intelligence has changed how people make and share digital content. In today's advanced world, it just requires few seconds to generate a new image and video. This has given rise to deepfakes. This are the fake images and videos that look real this are generated with the help of Deep-Learning technologies. The word deepfake stands for deep learning + fake.

Deepfakes can cause big harms because cybercriminals use them in several ways. They can spread false news, pretend to be someone else, commit financial frauds, threaten people to do something illegal. For example, in 2023 in Hongkong, cybercriminals used a deepfake video of the higher rank official on the video call to fool the worker of the company and worker sent them 39 million US Dollars. Another example, in 2022 a deep fake video of the President Zelensky was circulated which told his soldiers to stop fighting. The main problem is that it is very difficult and time consuming to separate deepfakes from the real facts. Another problem is that these deepfakes are spread with rapid speed and true facts. Because of this people has stopped trusting digital media, politics, privacy and money world.

Deepfake content is made using special ai tools that can create a new content such as pictures, videos and audios etc. most widely used tools are Generative Adversarial Network (GAN's). Another common and important tool is Autoencoders. These tools are used to put one person's face on another person's face, to make face of people who does not exist etc.

Multiple tools and models for detecting deepfakes have been developed by scientist and engineers. the number of research paper published on this topic grew from 3 in 2018 to 46 in 2020. One paper studied another 112 paper's and found that about 77% to 80% of them used Deep Learning. Rossler and other developed a test data called as FaceForensics++. It become popular test data set. Guera and Delp used to RNN's to study change in one image to next image in a video. Chintha and others combined CNN with RNN to study audio and video. Guarnera and other found digital marks left

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

behind by AI tools when they make fakes. Agarwal and Farid looked at movements such as blinking of eyes and head movements. Different research papers have also been studied . Sunil and others studied the techniques from 2018 to 2024 for images, audios and videos. Sunkari and Nagesh studied many tools for finding deepfakes and their social impact on the society.

However, some problems still exists. Firstly, these models cannot deal with new Techniques of making fakes or with unseen data. Whenever these models from perform on the unseen data their accuracy decreases significantly. Secondly, different studies and models uses different techniques to measure results and also use different amount of data. So, it becomes very difficult to compare their results. Lastly, most models are "black boxes", they just say these content is fake and these is real. In jobs like news reporting or law cases, people cannot make decisions without knowing the proper reasons.

To address these gaps, this paper presents Deepshield : A Platform for detecting deepfake faces with proper explanation . the main contributions are :

1. AI tool that analyse face in the input image and give results as "likely real" and " likely fake" along with a confidence score.
2. Grad-CAM Heatmap that show which parts of the face led to each answer, for better understanding of the results.
3. An Identity-aware splitting method that stores all pictures from the same video in the same group. Therefore, text data do not mix with the learning data.
4. The Safe and Secure web application built with FastAPI Backend, React Interface, and PostgreSQL data stores prediction history for records.

OBJECTIVES:

1. To study how Deepfake Images and Videos are generated and misused to spread false information, cheat people, commit financial frauds, harm the reputations therefore the problem solved by the model is clear.
2. To review and study current Deepfake detection models and test data sets such as FaceForensics++, CelebDF , DFDC and to find their limitations. They do not work on unseen data and do not explain the results and also mixes the test and training data.
3. To train an EfficientNet-B2 model on FaceForensics++, CelebDF and DFDC that recognizes a face as "likely fake" and "likely real" along with the confidence score.
4. To measure the performance of the model using different performance measures such Accuracy, Precision, Recall, F1-Score and ROC-AVC and compare with EfficientNet-B0 ResNet-50 and Xception, including speed.
5. To show which part of the face led to each result using grad-cam heatmaps for better understandings of results.
6. To develop a safe and secure web applications using React, FastAPI and PostgreSQL.

METHODOLOGY :

This part explains how DeepShield was built. DeepShield is a web app that checks whether an uploaded face image is real or a deepfake. The user uploads an image, and the result is either “Likely Real” or “Likely Deepfake.” It also shows a confidence score and a heatmap that points to the parts of the face the result is based on.

3.1 Overview and Workflow

1. An input image is uploaded to the model.
2. Server performs validation of the input image
3. System identifies the face in the image and crops it.
4. Face is resized to 260x260 pixels.
5. EfficientNet-B2 model identify whether the face is real of fake.
6. Heatmap created by grad-cam shows which part of image has model focused.
7. Result and images are saved
8. The result is displayed to the user on the dashboard.

If system do not find the face, system do not make any prediction. instead, it informs that input image is not valid for analysis. Each step is explained in the following sections.

3.2 Datasets

Deepshield is trained on the three most popular datasets use for deepfake verification. these datasets consists of videos of real faces and videos of fake face made from AI.

FaceForensics++ : It contains 1000 , videos from Youtube of real faces. These videos are modified using deepfake

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

methods : Deepfakes, Face2Face, Faceswap, and neutral textures. The quality of video's is of three levels : Raw (c0), Lightly compressed (c23), and Highly compressed (c40).

Celeb-DF (v2): It has 590 real videos of celebrity faces and 5639 fake videos made from AI. The faces in this dataset seems very real and have very less flaws.

DFDC (Deepfake Detection Challenge) : 100,000 video clips of different faces with different lighting and background is present in this dataset.

The Deepshield model mainly works on the images, so frames are used from each video, and face is cropped from each frame. Frames from fake videos are labelled as fake while those from real videos are labelled as real. Afterwards, this labelled set of faces is divided to training data, validation data and test dataset as mentioned in section 3.3.

Dataset	Real videos	Fake videos	Main feature
FaceForensics++	1,000	4 methods per real video	Three compression levels (c0, c23, c40)
Celeb-DF (v2)	590	5,639	High-quality, realistic fakes
DFDC	Not stated in the files	Not stated in the files	100,000+ clips with varied people and lighting

Table 3.1: Summary of datasets used in DeepShield

3.3 Dataset Organization and Leakage Prevention

Once the frames are obtained from the videos, the images containing faces are split into the training data, validation data and test dataset. The training dataset is used to train the model, validation data is used for tuning the model and test dataset is used to check the performance and accuracy of the model.

Frames from the same video looks similar. They have the same person with common features like background and lighting. Therefore, if some of these frames go into training dataset and some go into the test dataset, model will remember the common features instead of learning the sign of the fake data. This process is called as data leakage. Because of this, test accuracy of the model is very high but it performs very poorly on the unseen data.

To solve this problem, Deepshield uses an Identity-Aware split. The script reads the video or identity ID from each filename and groups together the frames that come from the same source. The dataset contains each group, using a 70%, 15%, and 15% split for training data, validation data and test data as described in table 3.2 below. one dataset contains only one video or identity.

Before the training, the dataset is checked by verification script. It counts the real and fake images in each dataset, finds broken files, and works out the class imbalance ratio.

et	Ratio	Purpose
Training	70%	Teach the model
Validation	15%	Tune the model and stop training early
Test	15%	Final evaluation on unseen videos

TABLE I : DATASET SPLIT RATIOS

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

3.4 Input Validation and Preprocessing

An image undergoes two steps in Deepshield before it reaches the model.

1. **Input Validation :** Firstly, the file uploaded is checked whether it is safe and it is safe to use. File formats such as JPG, JPEG, PNG and WEBP are only accepted. And size of the file should be only up-to 10MB. Server opens the header of the uploaded file to confirm that it is not broken and uploaded image is of size between 32 x 32 and 8192 x 8192 pixels. A random id is given as a file name. the validation step are shown in Table III.

2. **Face Detections :** To find the main face in the image Deepshield uses OpenCV. Then the size of the bounding box is increased by 20% in width and height. This extra space is covered but facial features such as jawline, chin and hair edges, where the fakes are often detected. If invalid face is found system stops processing and display error message :

"Unable to analyse this image reliably due to no suitable face detected."

3. **Resizing and normalization:** The face is resized to 260 x 260 pixels, because EfficientNet-B2 need this input size. Mean and Standard Deviation are used to adjust the pixel values of ImageNet, because previously model was trained on the ImageNet images.

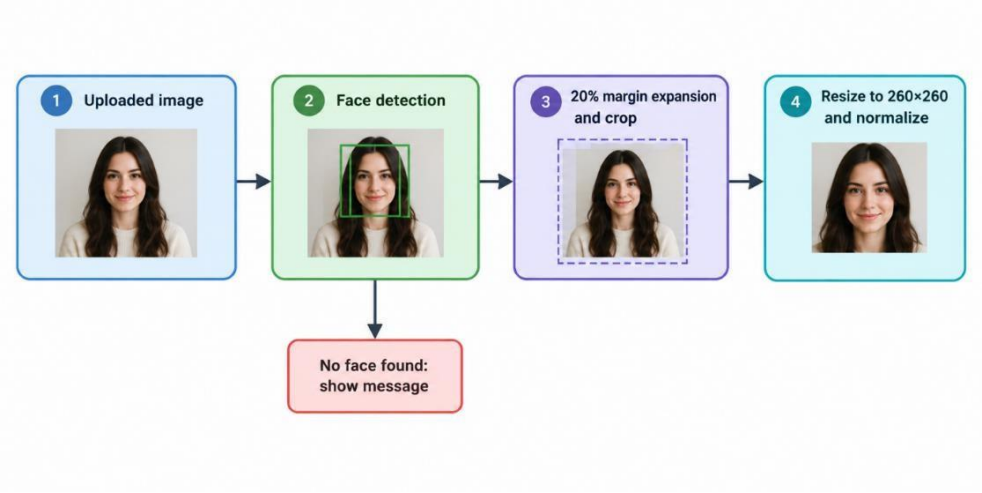


Fig. 3. Preprocessing pipeline of DeepShield.

Check	Rule
File type	JPG, JPEG, PNG, WEBP only
File size	Up to 10 MB
File integrity	Image header must be readable
Image dimensions	32×32 to 8192×8192 pixels
File name	Replaced with a random ID

TABLE III : INPUT VALIDATION CHECKS

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

3.5 Model Architecture

EfficientNet-B2 network is used by Deepshield as its main model. Image classification model of EfficientNet-B2 model grow three things together: width, depth and resolution. Because, of this balanced growth, network is able to reach good accuracy with low computing cost.

Transfer learning : Huge amount of data is required to train a model from scratch. To manage this need, in EfficientNet-B2 model, early layer of weights in convolutional layers are trained and are able to identify edges, textures and are able to identify edges, textures and shapes in images. Then the fine-tuning of model on deepfake face image is performed.

Classification Head : The original ImageNet classifier is replaced with two layers : a dropout layer (p value = 0.3) and fully connected layer 1408 input values and
2 output values. Some connection are turned off by dropout during training, which protects model from overfitting.

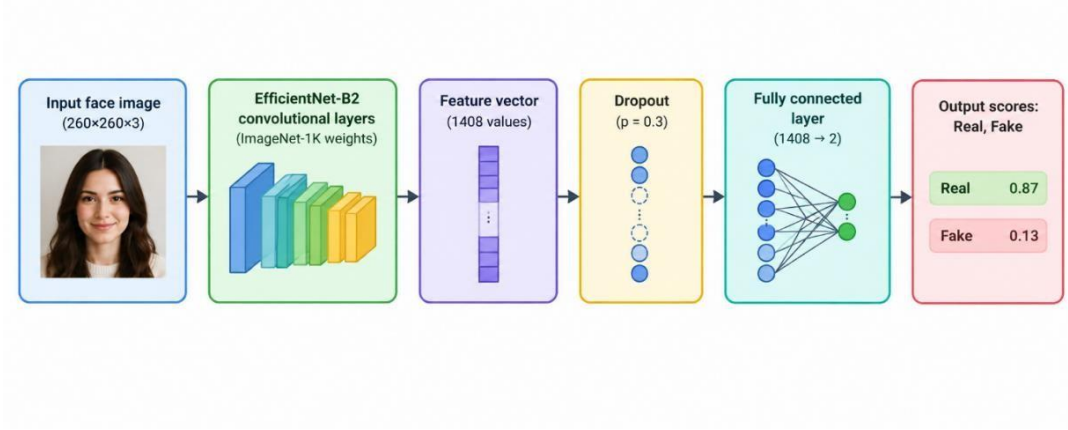


Fig. 4. EfficientNet-B2 architecture used in DeepShield with the replaced classification head.

Automatic feature learning : Features like textures or lighting are chosen by hand. Important patterns such as edge mixing and difference in textures are learned by convolutional layers on their own with the help of training image.

Other Architectures : Other similar algorithms can also be build such EfficientNet-B0, Restnet-50 and Xception. This are used for comparison .

Item	Value
Backbone	EfficientNet-B2
Initial weights	ImageNet-1K
Input size	260×260 pixels
Dropout	0.3
Final layer	Linear (1408 → 2)
Output classes	Real, Fake

TABLE IV : MODEL CONFIGURATION

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

3.6 Training Procedure

Transfer learning is used to train the model on training dataset, and validation dataset is used to test the model at the end of each epoch. An epoch is the one full pass through the training data.

Optimizer and Settings : AdamW optimizer with a learning rate of 0.0001 and weight decay of 10^{-4} is used by Deepshield. the batch size is of 16 and maximum number of epoch for training is 20. the main parameters are shown in table V below

Item	Value
Optimizer	AdamW
Learning rate	0.0001
Weight decay	10^{-4}
Loss function	Weighted cross-entropy
Maximum epochs	20
Batch size	16
Learning rate scheduler	ReduceLROnPlateau (factor 0.5, patience 2)
Early stopping	Patience of 5 epochs
Precision	Mixed precision (when GPU is available)

TABLE V : TRAINING SETTINGS

Class Weighting : The number of fake images are more than real images in a deepfake dataset. To fix this problem, a weight is given to each class based on how often it appears

W_c = N_total / (2 × N_c) (1)

where N_total is the total number of training images and N_c is the number of images in class c. heigher weight is given to classes that appear less often.

Data Augmentation : The images in the training dataset are randomly changed. so that model should not learn the small pixed patterns. These include steps like : flipping the image horizontally (50% probability), rotating it by up to ±15°, and changing its brightness, contrast, saturation (±15%), and hue (±5%).

Learning rate and Early stopping : The learning rate is cut in half, if the validation loss does not improve for two epoch in a row. If there is no improvement for five epochs then training is stopped.

Hardware and Saved models : Mixed precision training is used to save memory and speed up training if a GPU is available. If GPU is not available, then training is done on CPU. the best models are with least validation loss given by weights.

3.7 Classification and Confidence Scoring

The two raw numbers are given by the network namely, one for the real class and one for the fake class. This process take place after the face image is passed through the network. These raw numbers are turned into probability values with softmax function :

P(c) = e^(z_c) / (e^(z_real) + e^(z_fake)) (2)

where z_c is the raw score of class c. The two probabilities always add up to 1.

The predicted class is the class with the highest probability. this probability values are called as confidence score. For example, if the fake probability is 0.96, the predicted class is “Likely Deepfake” with 96% confidence.

In deepshield, classes are named as "likely real" and "likely fake" and not as "real" and "fake". This naming is because of probablistic nature of deepshield model. the confidence score only shows, how much the model believes, that the image is real or fake. The model is loaded only one , at the moment when the server starts.

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

Output	Meaning
Real probability	Model's score for the real class
Fake probability	Model's score for the fake class
Prediction	Class with the higher probability
Label	Likely Real or Likely Deepfake
Confidence	Probability of the predicted class

TABLE VI : PREDICTION OUTPUT FIELDS

3.8 Explainability with Grad-CAM

Usually, a deep learning does not explain its results. to increase the transparancy of deepshield, Grad-CAM is used. Grad-CAM creates a heatmap which highlights important parts of face used for prediction.

How it works : During the prediction, feature maps from the final convolutional block are taken by the Grad-CAM along with their gradients. These gradients displays the importance of feature map for the predicted class. Importance of feature map is given by the average of gradients :

$$\alpha_k^c = (1/Z) \Sigma_i \Sigma_j \partial Y^c / \partial A^k_{ij} \quad (3)$$

where A^k is the k-th feature map, Y^c is the score for class c, and Z is the number of pixels in the map. The feature maps are then weighted using this value, and all negative numbers are removed:

$$L^c = ReLU(\Sigma_k \alpha_k^c A^k) \quad (4)$$

Output images : Jetcolormaps shows the colors of the map which is scaled between 0 & 1. Red and yellow show high influence, and blue shows low influence. about 50 % , the heatmap is blended with face image to make an overlay. therefore, model displays these three images.

How to read results : The highlighted area shows where the model looked for something suspicious. This view shows that these area was edited. These heatmaps supports the human reviewer.

3.9 System Implementation

Deepshield is designed and built as a web application with four parts : user interface, server, database and file storage are

Component	Technology	Role
Frontend	React 18, Vite	Upload, results, history
Backend	FastAPI, Uvicorn	Validation, prediction, API
Machine learning	PyTorch, OpenCV	Face detection, classification, Grad-CAM
Database	PostgreSQL, SQLAlchemy	Analysis records
Storage	Local folders	Original images, heatmaps, overlays
Web server	Nginx	Reverse proxy and static files

TABLE VII: SYSTEM COMPONENTS

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

User Interface : Interface is built using Reat18 and Vite. It include different page which shows different activities such as to upload an image , to view the prediction result, to past analyses, to see the working of the model. The prediction output, confidence meter and three Grad-CAM pictures are displayed on the result page.

Server : The backend of the project is built using FastAPI and Uvicorn. It has three main functions : a health check to make sure that the model is ready, prediction function that takes an image and produce a result, history functions to list, show and remove or delete the past analyses.

Database and Storage : PostgreSQL database saves every analyses . Different entries are recorded such as unique ID, file name, prediction, probabilities, confidence, model name and version, processing time, image size, and the paths of the saved images. A separate folder on server stores original image, heatmap and overlay.

3.10 Security and Ethical Safeguards

Since, Deepshield accepts files from users , it has safeguards against misuse. It also states the limitations of the system, that what is can do and what cannot do.

Security Measures : Before the model receives file, the uploaded file undergoes checks for file format, size and integrity as **Traceability :** SHA-256 hash of the uploaded file is shown on the result page. This links a report to exact file that was checked. Full results as a JSON report could be downloaded by the user.

shown in Table III. No uploaded file can be run as a program. The uploaded file is saved under the name of randomly generated, which it makes difficult for the user to choose file name and to reach the other folders on servers. Use of safe parameterized queries for all database action is made, which prevents SQL injections attacks. Websites allowed in CORS settings are only allowed to access the API.

Threat	**Defence**
Harmful file upload (for example .exe or .php)	Only JPG, JPEG, PNG, and WEBP accepted; image header verified
Path traversal (for example ../../etc/passwd)	File names cleaned and replaced with random IDs
Very large or oversized images	10 MB limit and dimension limits
Image decompression bomb	Early failure on corrupted headers and image safety limits
Information leakage through errors	Generic error messages; details only in server logs
Unauthorized cross-site use	CORS whitelist
SQL injection	Parameterized queries through SQLAlchemy

TABLE VIII : THREATS AND DEFENSES

Ethical Safeguards : Deepshield is designed as Probablistic tool, therefore it's results are called "likely real" and "likely fake" and they are not displayed as th concrete proof. If no face is detected, system does not make a prediction. The

Grad-CAM heatmaps are shown to support human review and it should not be used alone as legal evidence.

3.11 Evaluation Methodology

Following training, the best-performing model is evaluated using the test set defined in Section 3.3. The images in this

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

set have not been seen by the model during training or validation, and none of the videos or people in this set appear in any of the other sets.

Metrics : The performance of the model is evaluated using Accuracy, Precision, Recall, F1 score, and ROC-AUC. In the formulas below, TP and TN are true and false positives and negatives, respectively. Fake images have been considered as a positive class.

Accuracy = (TP + TN) / (TP + TN + FP + FN) (5)

Precision = TP / (TP + FP) (6) Recall = TP / (TP + FN) (7)

F1 = 2 × Precision × Recall / (Precision + Recall) (8)

Precision represents the proportion of false images marked as false. The recall measures the ability to detect fake images. ROC-AUC is used to determine the performance of the model in classifying real and fake images for any cut-off. This should be close to 1.

Visual results : The confusion matrix, the ROC curve, the precision-recall curve, as well as the loss and accuracy curves are also generated.

Model comparison : The model EfficientNet-B2 will be compared against EfficientNet-B0, ResNet-50, and Xception with the same dataset and evaluation metrics. Time for inference on each image is also estimated. The results are provided in the Results section.

RELATED WORKS :

Deepfake detection has advanced from manipulation-specific visual artifacts to the development of more generalizable, temporally aware, and interpretable detection frameworks. Thies et al. [1] Introduced the Face2Face technique, which is real-time facial reenactment and showed the possibilities of manipulating facial expressions in RGB videos. Subsequently, Güera and Delp [2] conducted research on the application of recurrent neural networks in detecting deepfake videos using convolutional features and temporal features. Rössler et al. [3] developed the benchmark called FaceForensics++, allowing the detection of facial manipulations using different compression rates. Nevertheless, detectors designed for particular methods of manipulation can become less effective in recognizing unknown data. To address this issue, Guarnera et al. [4] studied convolutional fingerprints and frequency-domain footprints generated by generative models, while Agarwal et al. [5] considered facial appearance and behavior, such as blinks and head turns, to differentiate between fake and real videos. Hu et al. [6] Further investigated disentangled representation learning in order to enhance generalization to novel manipulations. Rana et al. [7] Reviewed literature on deepfake detection and identified persistent issues that include the diversity of datasets and consistency in model evaluations. Multimodal analysis, generalization, and explainability have gained increasing attention in recent researches. Valente et al. [8] explored convolutional neural networks in detecting speech deepfakes, thereby establishing the relevance of audio analysis as an additional means of detection.

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

LITERATURE REVIEW :

Sr.No.	Author(s) & Year	Research Work and System	Major Features	Key Contributions	Limitations
1.	J. Thies et al. (2016)	Face2Face: real-time facial reenactment	Tracks the face and transfers expressions in RGB video	Showed live expression transfer with ordinary cameras	Struggles with extreme head poses; not a detector
2.	D. Güera and E. J. Delp (2018)	Deepfake video detection using RNNs	CNN frame features with an LSTM for frame sequences	Early use of frame-to-frame changes for detection	Gives no explanation of its result
3.	A. Rössler et al. (2019)	FaceForensics++ benchmark dataset	Real videos manipulated by four methods at different compression levels	Created the most used benchmark for detectors	Detectors trained on it generalize poorly to new fakes
4.	L. Guarnera et al. (2020)	Detection using convolutional traces	Finds GAN fingerprints through frequency-domain analysis	Showed generation models leave detectable traces	High false positives on non-GAN processed images
5.	S. Agarwal et al. (2020)	Detection from appearance and behavior	Combines face appearance with blinking and head movement	Showed natural behaviour helps expose fakes	Less useful on short clips and single images
6.	J. Hu et al. (2021)	Detection using disentangled representation learning	Separates forgery cues from other image content	Aimed to improve detection of unseen fakes	Generalization still limited; no visual explanation
7.	M. S. Rana et al. (2022)	Systematic review of deepfake detection	Reviewed 112 studies in four method groups	Found deep learning in about 77% of studies, with best results	No new experiments; inconsistent metrics across studies
8.	L. P. Valente et al. (2024)	Speech deepfake detection using CNNs	CNN on audio features	Showed audio features help detect fake speech	Audio only; no image or video support

RESEARCH GAP :

However, advances in artificial intelligence technologies have enabled the creation of convincing fake photos, videos, and audio files. These are referred to as deepfakes. There exist many methods of deepfake detection, but they are all associated with certain limitations. First, most existing models cannot recognize new types of deepfakes since they are trained on specific datasets and manipulation techniques. Thus, there is a need to create a better Deepfake Verifier that would be able to detect different types of fakes and operate efficiently on any dataset. Second, most existing

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

deepfake detection systems do not explain their decisions and simply classify data as fake or authentic, which hinders the understanding of the reason for such classifications. Hence, the best Deepfake Verifier should provide some explanation and highlight suspicious spots if possible. Third, researchers work with different sets of data and use different criteria for measuring performance of deepfake detection models. Applying common criteria for evaluation such as accuracy, precision, recall, F1-score, and AUC-ROC will help to make the comparison of these models more fair. Fourth, most of existing models take into account only one type of features instead of combining several types of them. The integration of spatial features, which analyze image details, temporal features, which track changes between frames in a video, and frequency features, which uncover patterns, can boost the accuracy of the detection process. Finally, most detection models need high computational power and increased processing time; therefore, they are hard to deploy on mobile devices and cannot be used in real-time settings. Their accuracy may also drop in case of poor-quality media or compressed videos. In light of the above, there is a necessity of developing a Deepfake Verifier capable of producing reliable outcomes without spending too much processing time and computational power. The outlined research issues serve as a foundation for developing a Deepfake Verifier. The key issue is that there is no cohesive method that takes into account all of these aspects at once. The solution to such problems may make deepfake detection more reliable and useful. It could also assist in verifying the veracity of digital media.

FUTURE SCOPE :

1. **Detection of New Types of Deepfakes :** For the future, there is an opportunity to develop the system further in order to recognize deepfakes generated by newly created AI methods. This could be achieved by training the model on various data sets, which would allow recognizing deepfakes unseen before.
2. **Improved Explanation of Results :** The model can be enhanced to enable more transparent explanations of the model's prediction. Methods like Grad-CAM can point out the
   suspicious regions within an image or frame of the video.
3. **Combining More Detection Features :** The extension of the proposed system is possible through the combination of spatial, temporal, and frequency-based features. This can assist in detecting various indicators of manipulations and performing better than using only one kind of feature.
4. **Audio and Video Deepfake Detection :** The system can be further enhanced in the future to analyze both audio and video. It
   could detect any abnormalities in the voice tone as well as discrepancies between speech and lip movements.
5. **Real-Time Deepfake Detection :** The system can be enhanced to enable video analysis during realtime calls, online conferences, and live streaming. Increased processing speed and efficient models can help deliver quick results.
6. **Mobile and Edge-Device Support :** The model can be optimized to work on phones and
   devices with low computational power. This will make deepfake detection more easily accessible without the need for powerful computers all the time.
7. **Improved Testing and Reliability :** It is possible to test the system against other data sets, video quality, and compression ratios through metrics like accuracy, precision, recall, F1-score, and AUC-ROC. This may help uncover weaknesses and make the results more reliable.

LIMITATIONS :

1. **Limited Detection of New Deepfakes:** This detection system might fail to recognize different categories of deepfake imagery and videos since new methods can be developed for making fake imagery and videos.
2. **Dependence on Dataset Quality**: Performance of the system depends on the quality and diversity of the datasets that are being used for training and testing. Bad quality or poor data may affect the detection efficiency.
3. **High Computational Requirements:** Use of Deep Learning in analysis of images and videos could be resource-intensive, leading to increased time taken for processing due to high computation requirements.
4. **Limited Audio Detection:** The current system mainly focuses on detecting fake images and videos. It does not fully detect fake audio or verify whether the audio matches the person shown in the video.
5. **Possibility of Incorrect Results:** This classifier may mislabel true information as false information or false information as true information. Factors such as changes in video quality, lighting, compression, and facial characteristics might impact this prediction.

CONCLUSION :

Deepfake Verifier is an effective tool which is designed for detection of the AI-generated images and videos to minimize misinformation and increase authenticity of digital content. The system was created with the help of Python, PyTorch, and OpenCV and evaluates facial characteristics,

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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 11 , November - 2026

skin texture, lighting, shadows, picture quality, face movements, and differences between video frames to detect any signs of manipulation. Moreover, it uses confidence score and Grad-CAM heatmap for increasing clarity of its estimations. But the performance of this tool strongly depends on the quality and variety of training data and it might have problems with detecting advanced deepfake images, low-quality pictures, strong compression, and unusual lighting. In addition, it is difficult to process videos because of their computation-intensive nature and this tool does not support audio deepfake detection. For the future, there is potential for improvement of the system through the use of bigger and more diverse data, sophisticated detection techniques,

deep fake analysis on the audio spectrum, and checks for mismatch between lip movements and speech. Real-time detection while making video calls and during live broadcasts could be beneficial in order to add even more value to the system. All in all, Deepfake Verifier proves to be an excellent starting point for development of more advanced solutions.

REFERENCES :

[1] L. Guarnera, O. Giudice, and S. Battiato, “Fighting deepfake by exposing the convolutional traces on images,” IEEE Access, vol. 8, pp. 165085–165098, 2020.

[2] L. P. Valente, M. M. S. de Souza, and A. M. Da Rocha, “Speech audio deepfake detection via convolutional neural networks,” in Proc. IEEE Int. Conf. Evolving Adapt. Intell. Syst. (EAIS), May 2024, pp. 1–6.

[3] S. Agarwal, H. Farid, T. El-Gaaly, and S.-N. Lim, “Detecting deep-fake videos from appearance and behavior,” in Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS), Dec. 2020, pp. 1–6.

[4] Güera, D., Delp, E. J. (2018, November). Deepfake-video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1–6). IEEE.

[5] J. Hu, S. Wang, and X. Li, “Improving the generalization ability of deepfake detection via disentangled representation learning,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2021, pp. 3577–3581.

[6] Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to detect manipulated facial images,” 2019, arXiv:1901.08971.

[7] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, “Deepfake detection: A systematic literature review,” IEEE Access, vol. 10, pp. 25494–25513, 2022.

[8] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, “Face2Face: Real-time face capture and reenactment of RGB videos,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2387–2395.

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