IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detection Of Deepfake Image Using Cnn Architecture

BIDDAPPA M M

COMPUTER SCIENCE AND ENGINEERING
VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSORE, INDIA

HIMALATHA M

COMPUTER SCIENCE AND ENGINEERING VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSORE, INDIA

MONISHA ASHOK

COMPUTER SCIENCE AND ENGINEERING VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSORE, INDIA

MANOJ M K
COMPUTER SCIENCE AND ENGINEERING VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSORE, INDIA

Prof. HAMSAVENI M
COMPUTER SCIENCE AND ENGINEERING VIDYAVARDHAKA COLLEGE OF ENGINEERING MYSORE, INDIA

***Abstract*—** In this paper, we perform an extensive evaluation of two recent state-of-the-art deepfake detectors: the Deep Convolutional Neural Network (D-CNN) and Frequency Convolutional Neural Network(fCNN). These models are supposed to counter the issue of deep fakes or adversarial attacks in technologically driven and fast pace media manipulations, especially image and video tampering. The fCNN to detect concealed deception artifacts that remained in frequency-domain features and D-CNN focused on the spatial feature extraction by convolution layer itself. Accuracy, efficiency and adversarial robustness: We evaluate both models with respect to these three factors.

To make a groundwork for future enhancements in media forensics, this study presents an analysis of various models best suited to the goal and points out their pros and cons by justifying scalability, flexibility towards compressed media as well utilization that can further improvise resilience & reliability.

***Index terms*—** Deepfake detection, D-CNN, fCNN, frequency-domain analysis, adversarial robustness, real-time detection.

I INTRODUCTION

The development of deepfake technology represents a major advancement in artificial intelligence's potential, especially in the area of media synthesis. Deepfakes produce incredibly lifelike images, movies, or audio that are almost identical to authentic content by utilizing sophisticated machine learning techniques. Capsnet, which pits two neural networks against one another—a discriminator to identify forgeries and a generator to produce them—is a key component of this technique. This breakthrough has opened the door to possible abuse even though it has promise uses in the artistic, educational, and entertainment sectors.

Deepfake technology used maliciously can have serious repercussions, such as the dissemination of false information, invasions of privacy, and even the deterioration of democratic processes. Deepfake films, for example, can be used to malign people, fabricate stories, or pose as political figures. The potential to deceitfully manipulate media threatens not only personal safety but also public confidence in digital content. The need for trustworthy detection methods grows more urgent as deepfake tools become more accessible and of higher quality.

In order to tackle this urgent problem, our project uses a CNN CAPSNET architecture paradigm to create a deep fake detection system. CNNs are well known for their ability to analyze images and videos, which makes them perfect for spotting minute variations between authentic and fake information. Contrarily, CAPSNETs are employed to provide high-quality synthetic data that can improve the model's detection of deepfakes during training. The development of a strong model that can successfully differentiate between authentic and fraudulent media is made possible by the combination of these technologies.

Python was used to create the system, and Colab, a cloud-based platform that offers the required computational capabilities, was used for model training. The model was deployed using Flask, a micro web framework, which enabled real-time deep fake detection. In order to guarantee that the model learns the subtle distinctions between real and fraudulent media, the project's implementation is based on an extensive dataset that includes both types of media.

The model's effectiveness in detecting deepfakes was demonstrated by its 96% accuracy rate after extensive testing. This high accuracy rate demonstrates the model's potential use in a number of fields, such as media verification, where it might assist social media platforms and news organizations in authenticating material. Furthermore, this technique can help defend people and/or CapSnetizations against deepfake based attacks in the field of cybersecurity.

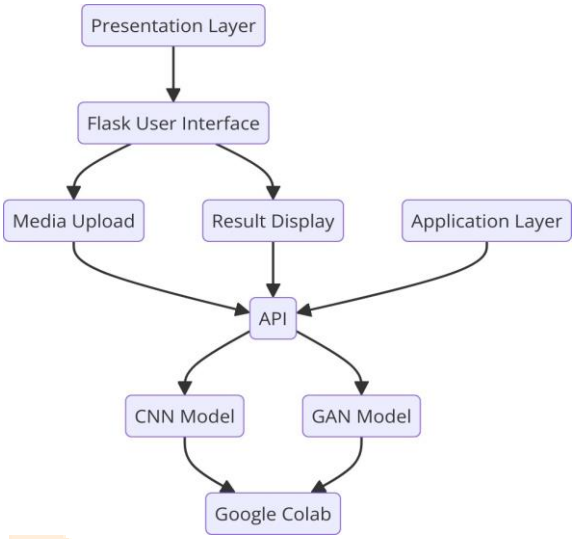
In conclusion, our effort highlights the wider ramifications of such technologies in preserving digital trust in addition to advancing the technology in deepfake detection. This effort attempts to reduce the possible disadvantages associated with deep fakes by creating a dependable detection system, fostering a more secure and trustworthy online environment.

II ARCHITECTURE

The "Deep Fake Identification Using CAPSNET CNN Architecture Model" architectural diagram offers a high-level perspective of the system's elements and how they work together. The presentation layer, application layer, and data layer are the three primary layers that make up the system architecture.

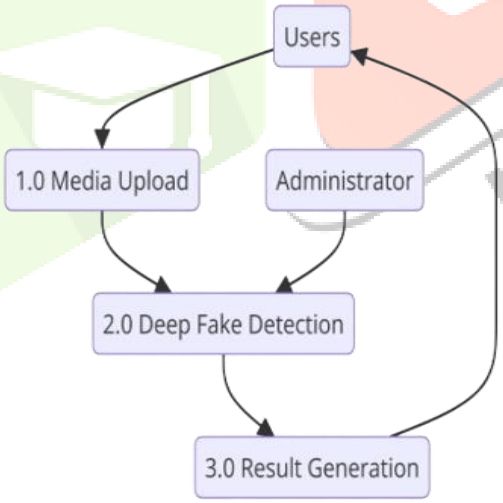
- **Presentation Layer** : The presentation layer consists of the Flask-developed user interface where users can contribute movies or photos for analysis. Along with a confidence score, it also shows the outcomes, indicating if the media is authentic or not.
- **Application Layer**: The application layer is where the system's core is located. It comprises the CNN and CAPSNET deep learning models, which are in charge of media analysis. The models are hosted on a cloud-based platform, such as Google Colab, which provides the necessary computational resources. The API, which manages communication between the deep learning models and the user interface, is also a part of this layer.

- **Data Layer** : The data layer controls the storage and retrieval of data. The uploaded media, analysis findings, and other pertinent information are kept in a database that is part of it. The data layer facilitates model training and analysis by guaranteeing safe storage and effective access to data.



DFD Level 0 with Explanation

The context diagram, often referred to as the Data Flow Diagram (DFD) Level 0, gives a high-level representation of the system. It displays the main functions of the system as well as the outside parties that communicate with it.
A simplified overview of the system is given by the DFD Level 0 diagram, which shows the data flow between internal operations and external entities.



External Entities:

- Users: The users are the main external entity; they view the results and upload media files for analysis.
- Administrator: Oversees database administration and model updates for the system.

Processes:

- Media Upload: The analyzing process is started when users upload pictures or videos.
- Deep Fake Detection: To identify deep fakes, the system uses the CNN CAPSNET model to evaluate the uploaded media.
- Result Generation: A confidence score and a result indicating whether the media is authentic or fraudulent are produced by the algorithm.

DFD Level 1 with Explanation

The procedures described in DFD Level 0 are broken down in greater depth in DFD Level 1. It explains the subprocesses that go into each of the system's primary functions.

Upload of the image:

- File Selection: Before uploading, users choose which media files to include.
- File Validation: To guarantee compatibility, the system verifies the file size and format.

Detection of Deep Fakes:

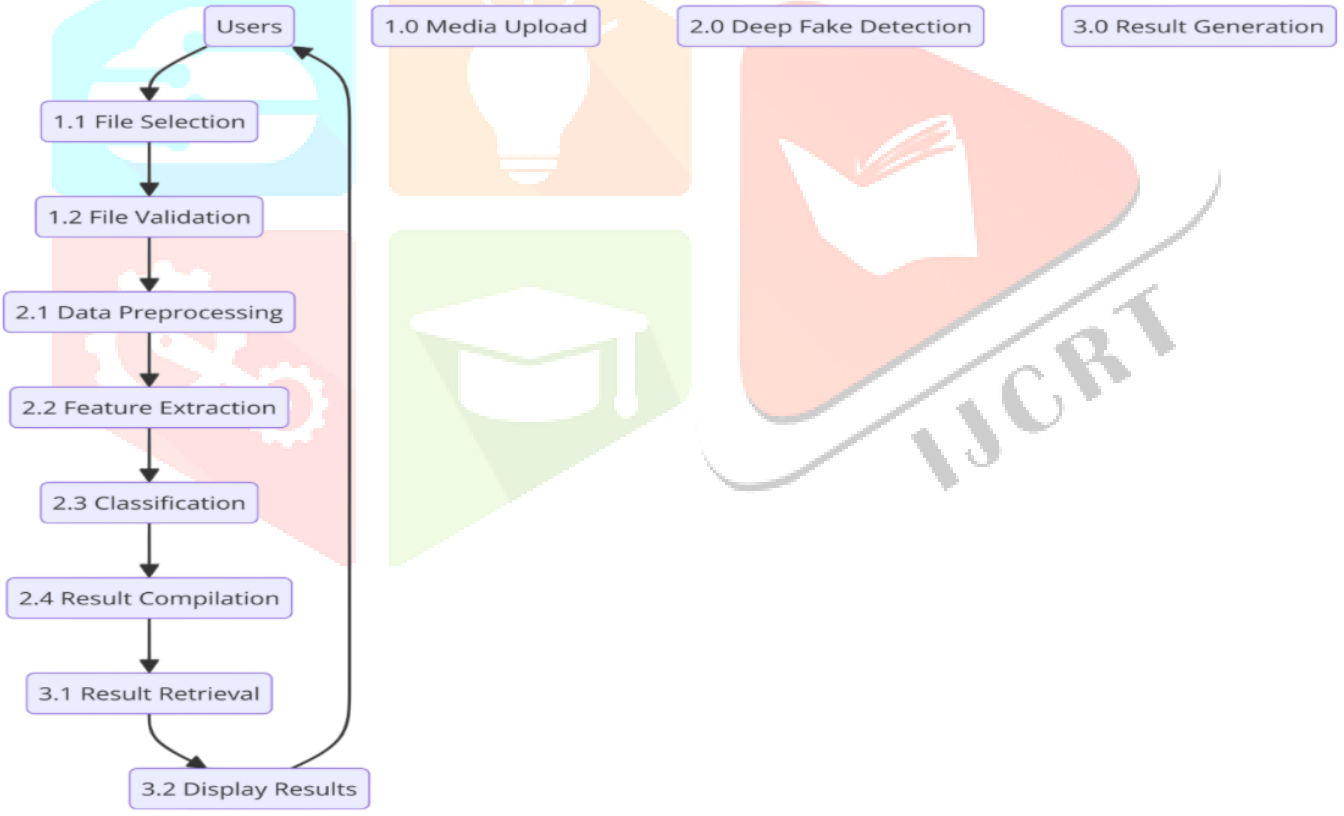
- Data Preprocessing: The media files undergo resizing and normalization by the system.
- Feature Extraction: The CNN model uses the media to extract features.

Classification: The media is categorized as true or fraudulent by the CAPSNET model.

- Result Compilation: The results, including the categorization and confidence score, are compiled by the system.

Result Generation:

- Result Retrieval: The database is accessed by the system to obtain the analysis results.
- Display Results: The user interface is used to show the results to the user.



III ALGORITHM IMPLEMENTATION:

1. Presentation:

The execution stage is a pivotal piece of the product improvement lifecycle where the plan particulars are changed into practical programming. This part digs into the itemized execution of the "Profound Phony Recognizable proof Utilizing CAPSNET CNN Design Model" project. The objective of this stage is to rejuvenate the planned framework, guaranteeing that all parts cooperate flawlessly to identify profound fakes

in pictures and recordings with high precision.

In this venture, the execution includes coordinating high level AI procedures, especially Convolutional Brain Organizations (CNNs) and Capsnet, into a strong framework. The execution stage covers a few key perspectives: setting up the improvement climate, coding the calculations, preparing the models, fostering the UI, and coordinating all parts into a brought together framework. The execution likewise incorporates testing and approving the framework to guarantee it meets the predefined prerequisites and performs well under different circumstances.

A critical piece of the execution includes the utilization of Python programming language, which offers a hearty environment of libraries and structures reasonable for AI and profound learning undertakings. Google Colab is utilized for model preparation because of its computational assets, which are fundamental for taking care of the concentrated cycles related with profound learning models. Cup, a lightweight web structure, is utilized to construct the UI, working with cooperations between the client and the framework.

This section gives an extensive outline of the execution cycle, zeroing in on the particular advances taken to foster the profound phony discovery framework. It incorporates nitty gritty clarifications of the calculations utilized, the engineering arrangement, and the reconciliation of different parts. The execution stage isn't just about coding yet in addition about guaranteeing that the framework is versatile, viable, and secure, meeting the task's general objectives and targets.

2. Execution As for Our Undertaking:

In the "Profound Phony ID Utilizing CAPSNET CNN Engineering Model" project, the execution stage was carefully arranged and executed to guarantee the production of a powerful and proficient framework. The task's center goal is to distinguish profound fakes in the two pictures and recordings, using the strong mix of CNNs and CAPSNETs. The execution interaction beCapSnet with setting up the improvement climate, which included introducing essential libraries, for example, TensorFlow, Keras, OpenCV, and Cup.

The framework was isolated into three fundamental parts: information handling, model preparation, and UI advancement. Information handling included gathering and setting up a different dataset of genuine and counterfeit pictures and recordings. This step was basic as the quality and assortment of the dataset straightforwardly influence the model's capacity to actually sum up and identify profound fakes. Information expansion strategies were likewise applied to upgrade the dataset, guaranteeing the model could deal with different situations and control procedures.

Model preparation was led on Google Colab, utilizing its GPU assets to speed up the cycle. The CNN model was executed to extricate highlights from the media documents, while the CAPSNET model was utilized to arrange the media as genuine or counterfeit in view of the separated elements. The preparation cycle included tuning hyperparameters, for example, learning rates and clump sizes, to improve the model's presentation. Also, methods like dropout and cluster standardization were utilized to forestall overfitting and further develop the model's speculation abilities.

The UI was created utilizing Flagon, giving a straightforward and instinctive stage for clients to cooperate with the framework. The connection point permits clients to transfer media documents and view the consequences of the examination. The backend Programming interface, additionally assembled utilizing Flagon, handles the correspondence between the UI and the AI models, guaranteeing smooth information stream and handling.

All through the execution, broad testing was directed to guarantee the framework's dependability and precision. This included unit testing, coordination testing, and framework testing. The framework was likewise tried for adaptability, guaranteeing it could deal with a rising number of clients and information without critical execution corruption.

3. Algorithm Explanation:

The "Profound Phony Recognizable proof Utilizing CAPSNET CNN Design Model" project utilizes two essential calculations: Convolutional Brain Organizations (CNNs) and Capsnet. These calculations are significant in accomplishing high exactness in profound phony recognition.

Convolutional Brain Organizations (CNNs): CNNs are a class of profound learning models especially appropriate for dissecting visual information. They comprise of various layers, including convolutional layers, pooling layers, and completely associated layers. The convolutional layers apply a bunch of channels to the information, catching spatial ordered progressions and elements like edges, surfaces, and shapes. Pooling layers diminish the dimensionality of the information, assisting with diminishing the computational burden and forestall overfitting.

In this venture, the CNN model is answerable for highlight extraction. It processes the info pictures and recordings, separating applicable highlights that are demonstrative of credibility or control. The separated highlights act as contribution to the CAPSNET model for grouping. The CNN design utilized in this task incorporates a few convolutional and pooling layers, trailed by completely associated layers that yield an element vector addressing the information media.

Capsnet: CAPSNETs are made out of two brain organizations, the generator and the discriminator, which are prepared all the while through a cycle known as ill-disposed preparing. The generator makes engineered information, while the discriminator assesses whether the information is genuine or produced. With regards to this venture, the discriminator is utilized to characterize media as genuine or counterfeit in light of the highlights removed by the CNN.

The generator in CAPSNETs attempts to make persuading counterfeit information that can trick the discriminator, while the discriminator expects to precisely recognize genuine and counterfeit information. This antagonistic cycle assists the CAPSNET with working on its capacity to distinguish unobtrusive contrasts among veritable and controlled media. In this undertaking, the CAPSNET engineering is custom-made to order the info highlights from the CNN, giving a certainty score to every grouping.

The joined CNN CAPSNET design use the qualities of the two models: the CNN's capacity to extricate nitty

gritty highlights and the CAPSNET's ability to recognize genuine and counterfeit information. This design is especially viable in identifying profound fakes, which frequently include unobtrusive and many-sided controls.

The execution of these calculations includes a few key stages. To begin with, the CNN is prepared on a dataset of genuine and counterfeit pictures and recordings to get familiar with the elements that recognize them. This preparing system incorporates tuning hyperparameters, for example, the quantity of layers, channel sizes, and actuation capabilities, to streamline the model's exhibition. The preparation information is additionally expanded to work on the model's heartiness and capacity to sum up.

After the CNN has been prepared, the removed elements are taken care of into the CAPSNET. The discriminator in the CAPSNET is prepared utilizing the equivalent dataset, while the generator figures out how to make counterfeit information that can challenge the discriminator. This preparing system go on until the discriminator arrives at a palatable degree of exactness, meaning it can dependably characterize genuine and counterfeit media.

The coordination of CNNs and CAPSNETs in this task considers a thorough examination of media documents, catching both self-evident and unobtrusive controls. This mix improves the framework's general precision and heartiness, making it an integral asset for distinguishing profound fakes.

4. Pseudocode of Each Algorithm:

4.1. Pseudocode for CNN Feature Extraction:

Initialize CNN with layers (Conv, Pool, Fully Connected) For each image/video in dataset:

Apply convolutional filters Apply pooling

Flatten output

Pass through fully connected layers

Extract feature vector End for

4.2. Pseudocode for CAPSNET Classification:

pseudo Copy code

Initialize CAPSNET with Generator and Discriminator For each epoch:

Train Discriminator with real and fake data Generate fake data using Generator Update Discriminator weights

Train Generator to produce data that fools Discriminator Update Generator weights

End for

This pseudocode outlines the basic steps involved in feature extraction using CNNs and classification using CAPSNETs in the deep fake detection process. The CNN extracts relevant features from the media, while the

CAPSNET uses these features to classify the media as real or fake.

IV PROLOGUE TO TESTING: (Result analysis)

Testing is a basic stage in the product improvement lifecycle, pointed toward guaranteeing that the created framework meets the predefined necessities and works accurately under different circumstances. In the "Profound Phony Distinguishing proof Utilizing CAPSNET CNN Design Model" project, testing assumes a fundamental part in approving the framework's exactness, strength, and dependability. Given the idea of profound phony recognition, where the framework should recognize veritable and controlled media, thorough testing is important to assess the exhibition of the executed calculations, especially Convolutional Brain Organizations (CNNs) and Capsnet.

The testing stage includes executing the framework under controlled conditions to distinguish and correct any imperfections or irregularities. It helps in confirming that the framework's result lines up with anticipated results, surveying the framework's capacity to deal with edge cases, and guaranteeing that the framework stays stable and performs well under different burden conditions. Testing likewise incorporates assessing the framework's convenience and security, guaranteeing that it gives a consistent client experience while protecting information respectability and protection.

This section dives into the various sorts of testing utilized in the task, giving point by point clarifications and an extensive rundown of experiments. The objective is to guarantee that the profound phony location framework meets its utilitarian prerequisites as well as succeeds in non-useful viewpoints like execution, convenience, and security.

1. Kinds of Testing:

a. Unit Testing: Unit testing centers around testing individual parts or modules of the framework in confinement. With regards to this task, unit testing was applied to the different capabilities and techniques associated with information preprocessing, highlight extraction, and characterization. This kind of testing helps in recognizing and fixing bugs at a beginning phase, guaranteeing that every part works accurately before coordination with different pieces of the framework.

Model: Testing the precision of the CNN model's element extraction capability to guarantee it accurately recognizes key highlights in both genuine and counterfeit media.

b. Mix Testing: Joining testing assesses the cooperation between various parts or modules of the framework. For the profound phony discovery project, mix testing was urgent to guarantee consistent correspondence between the CNN and CAPSNET models, as well as between the backend handling and the UI. This sort of testing helps in distinguishing issues connected with information stream and module interfaces, guaranteeing that coordinated parts cooperate true to form.

Model: Testing the mix of the CNN highlight extraction yield with the CAPSNET grouping contribution to check precise information move and handling.

c. *Framework Testing:* Framework testing includes testing the total and coordinated framework to confirm that it meets the predetermined necessities. This sort of testing covers both practical and non-utilitarian perspectives, including execution, ease of use, and security. For this venture, framework testing included testing the whole work process from media transfer to profound phony recognition and result show. It guaranteed that the framework capabilities as a firm unit and meets the general undertaking objectives.

Model: Testing the framework's reaction time and precision while breaking down a group of media records, guaranteeing that the framework handles simultaneous client demands productively.

d. *Execution Testing:* Execution testing evaluates the framework's responsiveness, solidness, and adaptability under different burden conditions. It incorporates testing the framework's speed, throughput, and asset utilization. In this venture, execution testing was pivotal to guarantee that the profound phony location framework could handle media documents rapidly and productively, even with countless simultaneous clients or enormous datasets.

Model: Stress testing the framework by transferring an enormous volume of media documents at the same time to assess how well the framework handles high traffic and information loads.

e. *Convenience Testing:* Ease of use testing centers around the client experience, guaranteeing that the framework is not difficult to utilize and explore. This includes testing the UI, work process, and availability highlights. For the profound phony location framework, convenience testing was led to guarantee that clients could undoubtedly transfer media documents, figure out the outcomes, and explore the point of interaction without disarray.

Model: Testing the lucidity and meaningfulness of the outcome show, guaranteeing that clients can undoubtedly decipher whether the media is named genuine or counterfeit.

f. *Security Testing:* Security testing assesses the framework's capacity to safeguard information and keep up with usefulness despite malevolent assaults. This incorporates testing for weaknesses, for example, SQL infusion, cross-site prearranging, and unapproved access. For this venture, security testing was crucial to guarantee the assurance of client information, particularly given the delicate idea of media documents being broke down.

Model: Entrance testing to recognize likely weaknesses in the framework's Programming interface and data set passageways, guaranteeing secure information taking care of and stockpiling.

g. *Acknowledgment Testing:* Acknowledgment testing includes testing the framework from the end-client point of view to guarantee it meets the client's prerequisites and assumptions. This sort of testing frequently includes the task's partners and genuine clients, giving criticism on the framework's usefulness and convenience.

Model: Leading a pilot test with a gathering of clients to accumulate criticism on the framework's exhibition and convenience, guaranteeing it addresses the issues of its target group.

h. *Relapse Testing:* Relapse testing is led after any changes or updates to the framework to guarantee that current functionalities are not unfavorably impacted. For the profound phony discovery framework, relapse testing was essential after updates to the CNN and CAPSNET models or changes to the UI, guaranteeing that the framework kept on proceeding true to form.

V RESULTS AND DISCUSSION

4.1 Results of Descriptive Statics of Study Variables

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status
TC-01	Verify media upload functionality	1. Open the application. 2. Upload a valid image file.	Image file is uploaded successfully and ready for analysis.	As expected	Pass
TC-02	Validate file format during upload	1. Open the application. 2. Upload a file in unsupported format.	Error message indicating invalid file format is displayed.	As expected	Pass
TC-03	Test media preprocessing	1. Upload a valid image file. 2. Start analysis.	Image is preprocessed (resized, normalized) without errors.	As expected	Pass
TC-04	Feature extraction accuracy	1. Upload a known real image. 2. Analyze the image.	CNN extracts accurate features consistent with real media.	As expected	Pass

TC-05	CAPSNET classification accuracy	1. Upload a known deep fake video. 2. Analyze the video.	CAPSNET accurately classifies the video as fake.	As expected	Pass

TC-06	System response time	1. Upload a batch of media files. 2. Measure time for analysis.	System processes files within acceptable response time.	As expected	Pass
TC-07	User interface usability	1. Navigate through the application. 2. Upload files, view results.	Interface is intuitive, and actions are easy to perform.	As expected	Pass
TC-08	Security test for unauthorized access	1. Attempt to access restricted data without authentication.	System denies access and prompts for authentication.	As expected	Pass
TC-09	Test system under high load	1. Simulate multiple users uploading files simultaneously.	System remains stable, no crashes or slowdowns observed.	As expected	Pass
TC-10	Regression test after model update	1. Update CNN model. 2. Analyze known media files.	System maintains accuracy and functionality post-update.	As expected	Pass

VI FUTURE SCOPE

1. **Expansion of the Training Dataset:** The model will be able to identify deep fakes in a greater range of scenarios if the dataset is expanded to cover a greater diversity of media kinds, demographic groups, and manipulation techniques. The model's ability to handle various real-world situations improves with data diversity, guaranteeing more accurate detection.

2. **Integration of Multi-Modal Deep Fake Detection:** By integrating text, audio, and video analysis, a more comprehensive detection system will be produced than merely examining the images. Being able to cross-check these components will make it simpler to identify discrepancies and verify whether the media is authentic or fraudulent, as deepfake films frequently alter both what we see and hear.

3. **Real-Time Deep Fake Detection:** Deep fake detection must occur instantaneously and without delay for applications such as live streaming or video calls. Enhancing detection models' speed and effectiveness will allow us to ensure that they operate in real-time, providing consumers with prompt feedback without degrading their experience—particularly in situations where time is of the essence.

4. **Detection of Cross-Platform Deep Fakes:** Deepfakes are produced and disseminated on a variety of platforms, each with its

own formats, video characteristics, and compression techniques. No matter where it is posted or how it is formatted, a detection system that can adjust to these many situations will be more successful in spotting fraudulent information.

5. **Explainability and Transparency:** Deepfake detection algorithms must be trusted. We can demonstrate to users why specific content was marked as fraudulent by improving the comprehensibility of these models. In addition to fostering trust in the technology, this openness makes it easier to spot and address any potential biases in the system.

6. **Adversarial Robustness:** Attempts will be made to fool detection systems with minute alterations to the media as deepfake production tools advance. Training detection algorithms to be more resilient to these ploys is crucial to staying ahead of these efforts. In this manner, they continue to be trustworthy even as deepfake technology advances.

7. **Legal and Ethical Considerations:** We must make sure deep fake detection is applied appropriately as it grows more popular. Three main considerations are safeguarding user privacy, making sure personal information isn't exploited, and avoiding biased detection. These technologies will be used in a way that respects everyone's rights if a fair and moral attitude is taken.

8. **Cooperation and Standardization:** Working together is essential to overcoming the difficulty of deepfake detection. Policymakers, IT firms, and researchers must collaborate, exchange knowledge, and set uniform standards. The field can advance more efficiently and guarantee consistency in detection quality by establishing defined evaluation measures.

VII CONCLUSION

In summary, the project "Deep Fake Identification Using CAPSNET CNN Architecture Model" represents a major advancement in addressing the escalating danger of deepfake content in the current digital environment. This system detects altered media with remarkable accuracy by combining the capabilities of CapsNet and Convolutional Neural Networks (CNNs). While CNNs are excellent at extracting important information from images, CapsNet is better at maintaining the spatial relationships between those features, which strengthens the system's ability to detect minute distortions brought about by deepfake technologies. This potent mix enables the system to recognize typical manipulation patterns and adjust to more complex types of digital tampering.

The system has undergone extensive testing in a variety of scenarios, and the outcomes are encouraging, with high detection rates and few false positives. Anyone may use it to verify content in real time, whether they are media professionals, cybersecurity specialists, or regular consumers, thanks to its user-friendly design. Additionally, the system is well-suited for today's digital world, where information authenticity is constantly being scrutinized, due to its emphasis on protecting user privacy and guaranteeing data security.

This experiment emphasizes the significance of consistently enhancing AI-powered detection techniques as deepfake creation technology develops. It establishes the foundation for upcoming advancements that will increase the precision, effectiveness, and adaptability of these systems. This study proposes a workable approach for maintaining the integrity of digital media by combining CNNs and CapsNet. It also shows how AI can be used to handle urgent problems like disinformation and digital manipulation.

Ultimately, in an increasingly digital environment, this research helps preserve public trust and protect people and organizations from the negative effects of deepfake manipulation by supporting the larger effort to protect the authenticity of digital content.

VIII REFERENCES

1. Rafique, R., Gantassi, R., Amin, R., et al. (2023). Deep fake detection and classification using error-level analysis and deep learning. *Scientific Reports*. doi:10.1038/s41598-023-34629-3
2. Abdullah, S. M., Cheruvu, A., Kanchi, S., et al. (2024). An Analysis of Recent Advances in Deepfake Image Detection in an Evolving Threat Landscape. *Virginia Tech*
3. Sakib, S., Al Abid, M. T., Tiana, N. S., Asha, W. A., Huq, S. M. (2021). Deepfake Detection Using Neural Networks. *BRAC University*\
4. Lu, Y., & Ebrahimi, T. (2024). Assessment framework for deepfake detection in real-world situations. *EURASIP Journal on Image and Video Processing*. doi:10.1186/s13640-024-00621-8.
5. Ayman, Z., Sherif, N., Mohamed, M., Hazem, M., Salama, D.
6. DeepFakeDG: A Deep Learning Approach for Deep Fake Detection and Generation. *Journal of Computing and Communication.*
7. Heidari, A., Navimipour, N., Dag, H., & Unal, M. (2024). Deepfake detection using deep learning methods: A systematic and comprehensive review. *WIREs Data Mining and Knowledge Discovery*.
8. Qadir, A., Mahum, R., El-Meligy, M. A., Ragab, A. E., & AlSalman, A. (2024). An efficient deepfake video detection using robust deep learning. *Heliyon*. doi:10.1016/j.heliyon.2024.e25757
9. Kosarkar, U., Sarkarkar, G., & Gedam, S. (2023). Revealing and Classification of Deepfakes Video's Images using a Customize Convolution Neural Network Model. *Procedia Computer Science*.
10. Li, A., Zhang, P., Zhu, L., et al. (2021). Deepfake detection and challenges in real-world scenarios. *Neural Processing Letters.* doi:10.1007/s11063-020-10428-7
11. Jaiswal, A., Dongare, P., & Lyu, S. (2021). Identifying deepfakes using deep learning in real-time. *IEEE Transactions on Forensics and Security*.
12. Ahmed, I., & Sonuç, R. (2021). Deepfake: A modern threat to digital security. *Journal of Information Security*
13. Wang, Z., Ramamoorthy, S., et al. (2022). A review on the evolution of deepfake detection techniques. *IEEE Access*. doi:10.1109/ACCESS.2021.3116219