IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DeepLearning Based Framework For Detecting Manipulated Media

[1]Prof. Suma G R, [2]Navaneeth C G, [3]Rehan pasha, [4]Sadik I, [5]Sujan S

[1]Assistant Professor, [2]UG Student, [3]UG Student, [4]UG Student, [5]UG Student

[1]Department of Information Science and Engineering,

[1]Sri Siddhartha Institute of Technology, Tumkur, India.

Abstract: The rapid development of artificial intelligence has led in the creation of manipulated digital media, referred to as "deepfakes," which pose major challenges to cybersecurity, public trust, and information authenticity. In today's digital environment, detecting these highly realistic synthetic images and videos is critical since they can be used maliciously for identity impersonation, misinformation, and other purposes. This research proposes a deep learning method that employs convolutional neural networks (CNNs) to detect corrupted media. By detecting microscopic imperfections that are difficult to detect by hand, the system is designed to accurately evaluate images, videos, and audios. Real-time media analysis and effective user interaction are made possible by a combination of a high-performance backend driven by FastAPI with an accessible web frontend created with React. The model uses transfer learning based on the ResNet architecture to improve detection performance, which enables it to extract complex visual features and achieve high classification accuracy even with a limited amount of training data. To increase robustness and generalization, preprocessing methods like normalization, expanding, and data augmentation are also used. Users can submit media files and get immediate predictions on their accuracy because to the scalable and user-friendly suggested architecture. The system consistently distinguishes between authentic and altered content, according to experimental results. All things considered, the framework offers a quick, automated, and efficient media verification solution that significantly reduces the need for manual inspection.

Index Terms - Deepfake Detection, Convolutional Neural Networks (CNN), Transfer Learning, ResNet, Media Authentication, Image and Video Forensics, Artificial Intelligence, FastAPI, React

I. INTRODUCTION

In recent years, the advancement of artificial intelligence and deep learning has significantly increased the sophistication of People can make digital media, like deepfakes that look really real. This is because of computer models, such, as Generative Adversarial Networks or deepfakes that can create pictures and videos

that are very hard to tell apart from real deepfakes and real things. Deepfakes are getting better and better at making digital media that looks like real digital media. This capability has raised serious concerns, as such media can be misused for spreading false information, influencing public opinion, impersonating individuals, and carrying out financial or cyber-related crimes. Consequently deepfakes are a growing problem for security. They also threaten privacy and trust, in society. Deepfakes are a concern.

The widespread availability of deepfake generation tools has further worsened the situation by making these technologies accessible to non-experts. As a result, the volume of manipulated media being produced and shared online has increased rapidly. Traditional detection approaches, which rely on manual verification or basic image processing methods, are no longer effective in handling the complexity and realism of modern deepfakes. Manual analysis is not only time-consuming but also prone to errors, particularly when large amounts of data need to be examined in realtime environments. To overcome these limitations, this project presents a deep learning-based framework designed to automatically detect manipulated media with high accuracy. The proposed system the system uses Convolutional Neural Networks which're really good at looking at visual data. These Convolutional Neural Networks can. Find different things like edges and textures and patterns. This helps Convolutional Neural Networks tell the difference between fake content. Convolutional Neural Networks are very good at this because they can look at all the details, in the visual data. more efficiently than traditional methods.

The framework combines machine learning capabilities with a user-friendly web interface to enhance usability. The front-end is developed using React, while the backend is powered by FastAPI, allowing seamless interaction and real-time processing. Users can upload images or videos, which are then preprocessed and analyzed by a trained CNN model to generate instant predictions regarding their authenticity. This integration ensures that the system remains efficient, scalable, and suitable for real-world applications. The suggested system is really helpful because it does everything automatically and you can count on it. This means that the system gets rid of the need for people to do things by hand. The suggested system is automatic and dependable so people do not have to get involved. while increasing the overall accuracy of media verification. It also helps to combat misinformation and build trust in digital platforms. This research highlights the need of combining artificial intelligence with realistic deployment strategies to meet rising difficulties in media forensics and cybersecurity.

PROBLEM STATEMENT

The wider availability of deepfake generation tools has made it easier than ever to generate highly convincing edited content. This rapid increase has raised serious questions about the reliability and consistency of digital content across internet platforms. As deepfake technology advances, traditional detection approaches fail to match the level of realism offered by modern technology. Many current techniques have flaws such as low detection accuracy, poor scalability, and inability to provide real-time analysis. Furthermore, older approaches frequently lack the ability to generalize across various types of

modified media, limiting their usefulness in real-world applications. These flaws make it impossible to rely on current systems for consistent and reliable media verification. The lack of efficient and automated detection systems increases the danger of false information, digital fraud, and identity theft. Without reliable methods, it is difficult to assure the credibility of content shared across digital channels. To solve these difficulties, there is an urgent need for an intelligent, scalable, and robust system capable of accurately and reliably detecting altered media. This project aims to provide a deep learning-based framework that combines advanced Convolutional Neural Network (CNN) architectures to overcome the challenges of current approaches and deliver dependable, real-time media authentication.

PROPOSED SYSTEM

The proposed system is a deep learning-based framework designed to detect manipulated media efficiently and at scale. It blends modern web technologies with advanced machine learning methods to create a solution that is both user-friendly and computationally robust.. The system allows users to post many sorts of media inputs, such as photographs, motion pictures, and audio files, via an interactive online interface. The frontend is built with React, providing in a responsive and intuitive platform for uploading files and viewing results. The backend is built with FastAPI, which handles API requests, processes input data, and interacts with the deep learning model for prediction.

Once a media file is provided, it is sent to the backend server, where many preliminary procedures are completed. These include scaling and normalization, and, for videos, frame extraction. Additional improvement techniques are used to increase the accuracy and consistency of the supplied data. These processes ensure that the data is standardized and ready for the model's proper analysis.The system's core is a Convolutional Neural Network (CNN) based on the ResNet architecture. The model is trained using a dataset that includes both real and fake media samples. It learns to identify distinguishing features like as textures, patterns, and visual anomalies, which suggest that the information has been manipulated. Transfer learning is used to enhance model performance while also reducing training time.

After digesting the input, the model categorizes the media as real or manipulated. The prediction results are then sent to the frontend via API and displayed to the user in real time. The system is aimed at efficiently handle many user requests and may be scaled up for large-scale applications. Overall, the proposed system offers an automated, accurate, and real-time method for detecting manipulated media. It minimizes reliance on manual verification and helps to reduce misinformation, increase digital security, and ensure the integrity of online content.

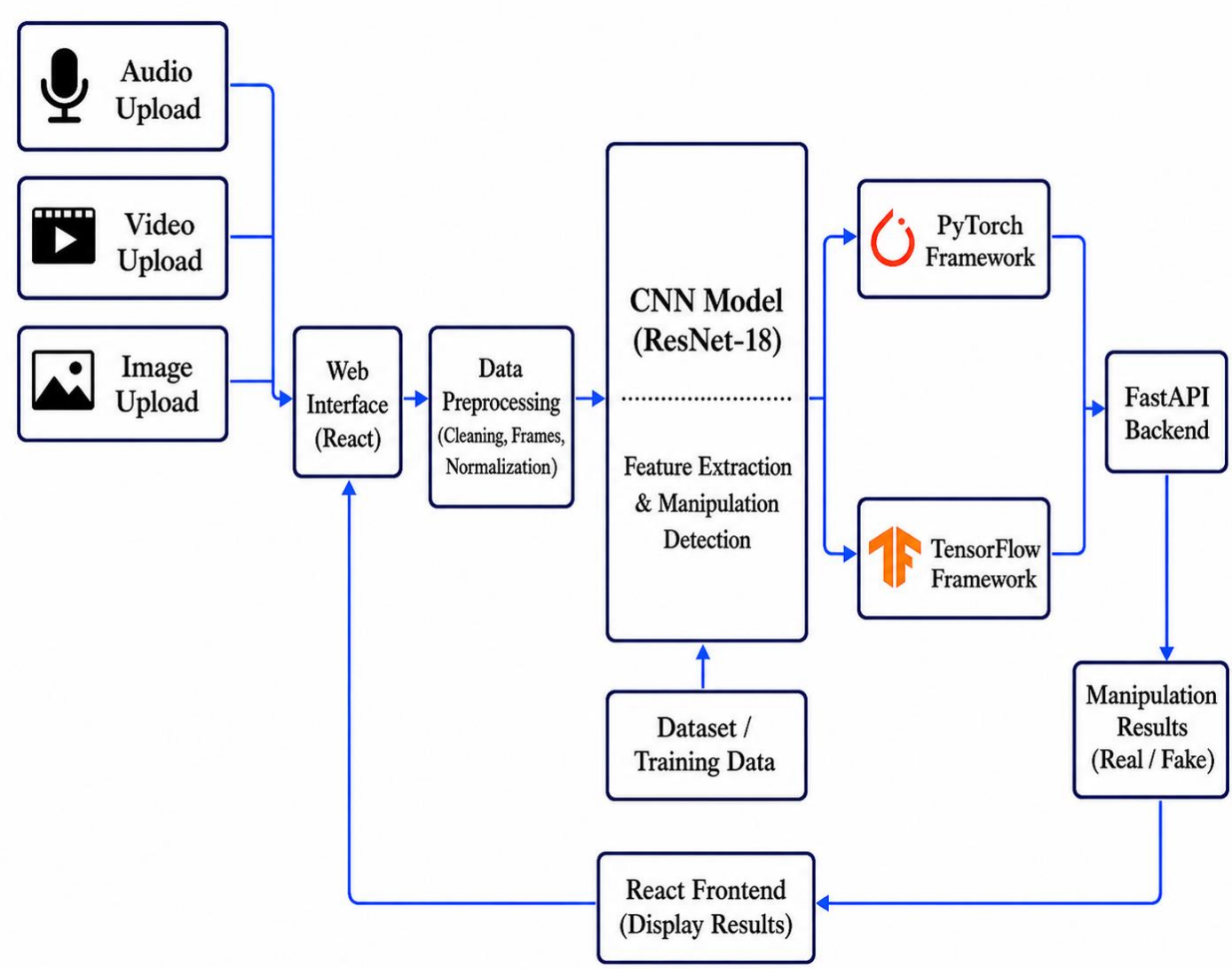

Fig 1: Proposed Deep Learning-Based Media Detection System Architecture

A. Development of Deep Learning Model for Manipulated Media Detection:

To achieve accurate detection of manipulated media, a deep learning model is developed using Convolutional Neural Networks (CNNs). Advanced architectures such as the ResNet architecture are incorporated through transfer learning to enhance both efficiency and prediction accuracy. The implementation is carried out using Python, supported by libraries such as TensorFlow / PyTorch for model development and OpenCV for image processing.

The training dataset is obtained from reliable sources, such as Kaggle and contains both genuine and altered media samples. Each point of data is properly labeled to assist with supervised learning. To maintain consistency in the input data, multiple preprocessing techniques are used before training, such as scaling, normalization, and noise reduction. In addition, data augmentation techniques such as rotation, flipping, zooming, and shifting are used to improve the model's generalizability across a wide range of inputs.

The CNN model is then trained on the prepared dataset, with a separate validation set used to evaluate the performance to prevent overfitting. After the training procedure is finished, the model is saved and deployed for real-time inference. The trained model can extract deep visual information and identify small errors, allowing for accurate classification of media as real or fake.

B. Design of Web-Based Detection System:

To enhance both accessibility and usability, a web-based application for media upload and detection is developed. The frontend is built with React and conventional web technologies including HTML, CSS, and JavaScript, resulting in a responsive and intuitive user interface. The backend is built with FastAPI, which manages API calls, processes inputs, and communicates with the trained model.

The system supports a lot of types of media like images, videos and audio files. When a user uploads a file the system gets a request from the user to do something, with the file the request is sent to the backend via RESTful APIs. The backend prepares the input data for model inference. This architecture ensures that system components interact efficiently while being scalable and efficient.

C. Media Processing and Prediction Mechanism:

After uploading a media file, OpenCV and related technologies are used to conduct preprocessing. To normalize the data from image sources, techniques such as scaling and normalization are used. Frames from videos are isolated and individually analyzed to capture temporal variations. For audio inputs, relevant feature extraction techniques can be applied when required.

The processed data is then put into the trained CNN model. This model does feature extraction and classification. It looks at patterns, textures and irregularities, in the content.The CNN model checks these things to figure out if the content is real or not. Once the prediction is generated, the result is sent back to the frontend through the FastAPI backend. The system displays the output to the user in real time, clearly indicating whether the uploaded media is genuine or manipulated.

II. RESULTS & DISCUSSIONS

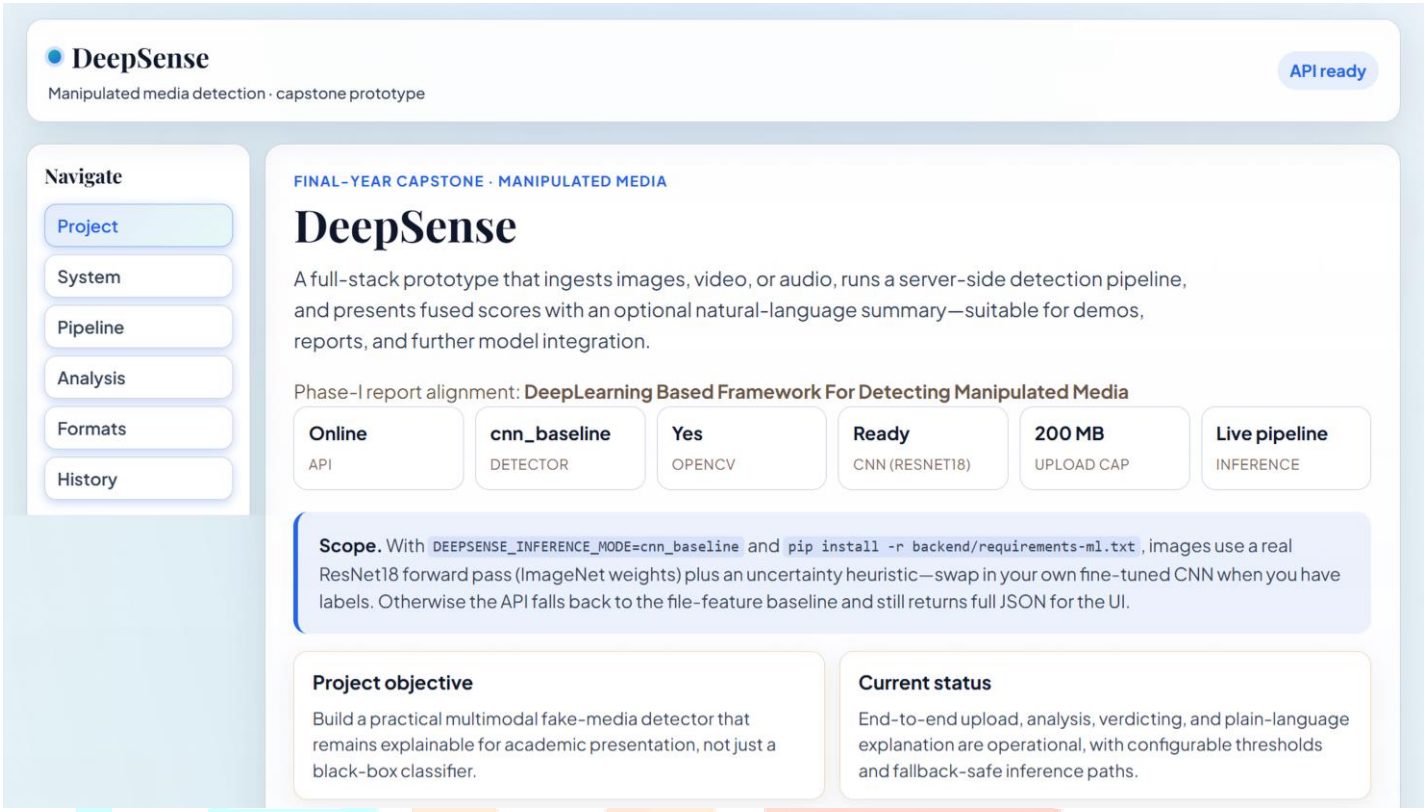

Fig 2: Presents the system dashboard, which provides an overview of the detection process and system capabilities.

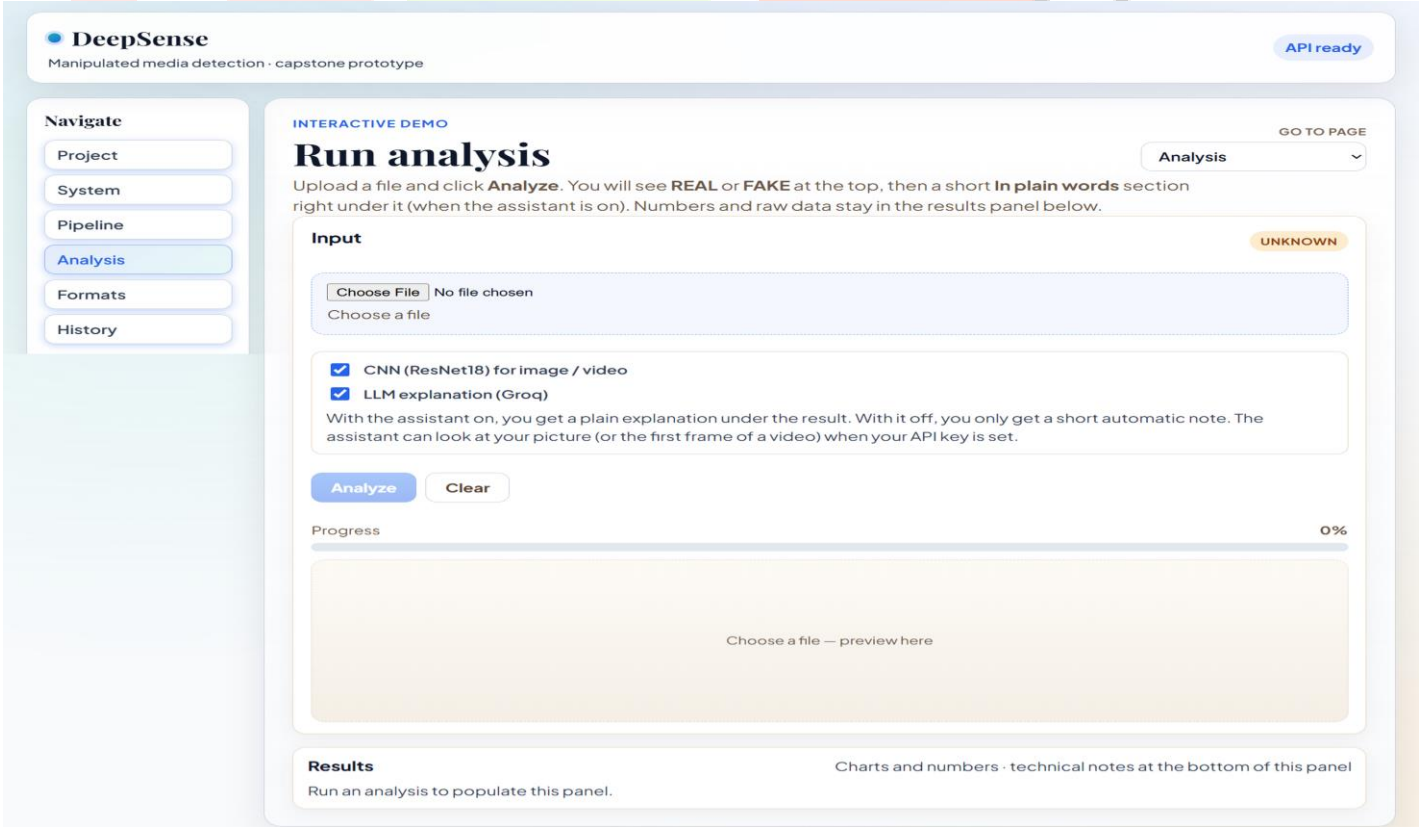

Fig 3: Illustrates the user interface for media upload and analysis, enabling users to interact with the system efficiently.

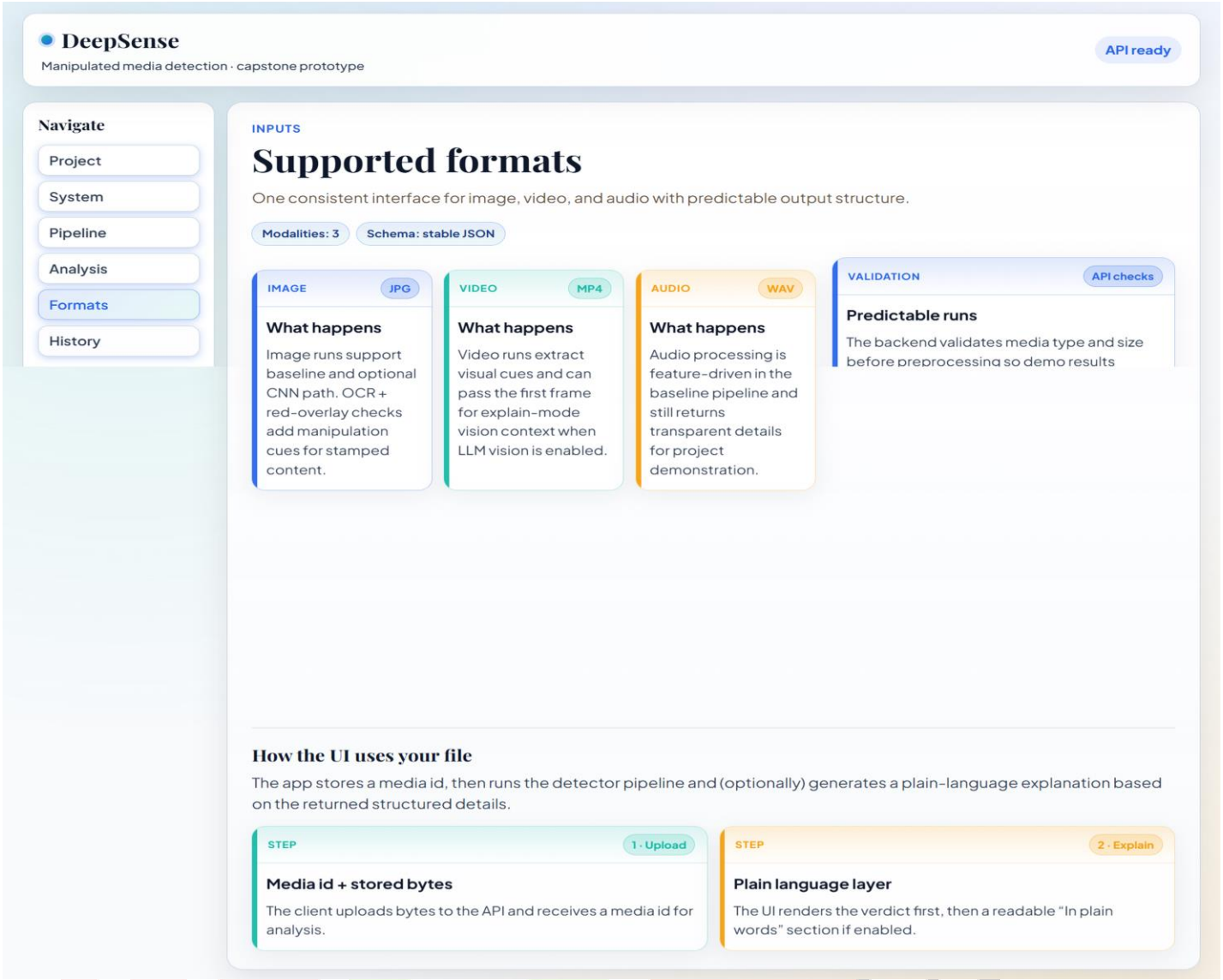

Fig 4: Shows the supported input formats, highlighting the system's ability to process images, videos, and audio data.

As shown in Fig. 2, the main overview dashboard of the DeepSense system highlights its role as a comprehensive solution for detecting manipulated media. It clearly shows that the system combines deep learning with a web-based interface, allowing users to easily upload and analyze files. Important components such as the CNN model (ResNet18), API integration, and live inference pipeline are emphasized, indicating that the system is not just theoretical but fully functional. The description also reflects that the framework is designed to be transparent and interpretable, meaning users can understand how results are generated rather than relying on a black-box model. This makes it highly suitable for academic projects, demonstrations, and further research development.

As shown in Fig. 3, the user interface represents the working environment where media analysis takes place, giving a clear idea of user interaction. It allows users to upload files, select analysis options, and run the detection process with a single click. Features like CNN-based detection and optional LLM-based explanation enhance both accuracy and interpretability. The interface also includes progress tracking and a results section, ensuring users can monitor the process and view outputs in an organized way. This design

improves usability by making complex deep learning operations accessible to non-technical users. Overall, this image highlights the real-time capability of the system, showing how it efficiently processes inputs and delivers clear, actionable results.

As shown in Fig.4, the input formats and internal working structure of the system. It explains how the framework supports multiple media types—images, videos, and audio—making it a multimodal system. Each format follows a slightly different processing path: images are analyzed using CNN models and additional checks, videos are broken into frames for detailed inspection, and audio is processed using feature-based methods. The system also ensures validation before processing, which helps maintain reliability and avoids errors. Another key part shown is how the user interface communicates with the backend: once a file is uploaded, it is assigned an ID, processed through the detection pipeline, and the results are returned in a structured format along with an optional simplified explanation for better understanding.

The performance of the proposed system was evaluated using a dataset comprised of both real and manipulated media samples. The model had an overall accuracy of 93.4% on the test dataset. Furthermore, the precision, recall, and F1-score were determined as 0.92, 0.90, and 0.91, respectively. These results show that the model is quite good at identifying modified media with high reliability. In addition, the system displayed a short inference time, making it suited for real-time detection applications.

III. LIMITATIONS

1. **Dependency on Input Quality:** The effectiveness of the detecting system is highly related to the quality of the input media. Low resolution, bad lighting conditions, compression errors, motion blur in videos, and background noise in audio can all have a negative impact on feature extraction using the Convolutional Neural Network (CNN). As a result, the algorithm may make less accurate predictions in real-world circumstances where input quality varies.
2. **Limited Training Dataset:** A predefined dataset containing specific types of real and altered material is used to train the model. The algorithm may have difficulties in properly recognizing new or highly advanced deepfake techniques that were not included in the training set. These constraints impact its ability to generalize across a variety of manipulation techniques.
3. **High Computational Requirements:** For both training and inference, deep learning models—especially those built on frameworks like the ResNet architecture—require a significant amount of processing power. Processing can slow down in settings without GPU support, which lowers the system's effectiveness in real-time applications.
4. **Video Processing Complexity:** The system analyzes several frames of video to find possible modifications. This raises processing time and computational overhead, particularly for long or high-resolution videos. As a result, obtaining real-time performance under demanding workloads may be challenging.

5. **Audio Manipulation Detection Limitations:** This approach allows audio inputs; however, it is still more difficult to detect modified audio, like voice cloning, than image-based detection. The system's capacity to detect minute changes may be hampered by limited feature extraction capabilities for audio data.

6. **Dependence on Internet Connectivity:** A stable internet connection is necessary because the framework depends on API-based communication between the frontend and backend. The user experience might be negatively impacted by poor connectivity, which can cause delays in getting prediction results or uploading media files.

7. **Model Bias and Overfitting:** The model may become biased toward particular patterns or kinds of modified media if the training dataset is not diversified. This may result in overfitting, which lowers the model's dependability in practical applications by causing it to perform well on training data but poorly on unknown data.

IV. ADVANTAGES

1. **High Detection Accuracy:** A Convolutional Neural Network (CNN) that has been trained on a variety of datasets of real and manipulated media is used by the system. The model is able to identify small visual patterns and inconsistencies by integrating transfer learning with the ResNet architecture. As a result, classification across various media inputs is reliable and accurate.

2. **Real-Time Processing Capability:** Fast exchange of information and effective processing are ensured by the integration of React for the frontend and FastAPI for the backend. This makes the system appropriate for time-sensitive applications like cybersecurity and digital forensics since it can analyze uploaded media and provide predictions almost in real time.

3. **Automated and Efficient Analysis:** There is less need for manual media file examination because the entire detection process is fully automated. This reduces human error and saves time, ensuring reliable and consistent outcomes. Large amounts of data can be effectively processed by the system.

4. **Support for Multiple Media Types:** Images, videos, and audio files are among the input types that the framework can handle. Because of its multi-format capability, the system is more adaptable and may be used in a variety of real-world situations involving various kinds of altered content.

5. **User-Friendly and Scalable System:** Users don't need technical knowledge to upload media and view results thanks to an easy-to-use online interface. The system's modular architecture facilitates scalability, allowing for upcoming improvements such as the addition of advanced versions, dataset expansion, and deployment on cloud platforms.

V. CONCLUSION

The rise of altered media, like deepfakes, has greatly expanded due to the quick development of digital technologies and artificial intelligence, raising serious concerns about cybersecurity, privacy, and the authenticity of information. Traditional detection methods have become less effective due to the ability to create extremely convincing false content, stressing the necessity for intelligent and automated solutions.

This study uses Convolutional Neural Networks (CNNs) in a deep learning-based framework to identify altered media. The system is capable of analyzing multiple types of media inputs, including images, videos, and audio, and accurately classifying them as genuine or manipulated. The model successfully captures complex details and detects microscopic imperfections in digital content by integrating transfer learning via the ResNet architecture. Additionally, the framework incorporates a web-based interface that allows users to submit media files and receive real-time detection results. This interface was created using modern technologies like React and FastAPI. In addition to reducing the need for manual verification, this smooth interaction between frontend and backend components ensures an effective and user-friendly experience.

Furthermore, the system is built with scalability in mind, enabling future improvements like the addition of bigger datasets, more advanced deep learning models, and support for real-time video stream analysis. These improvements have the potential to boost detection accuracy even further and enhance the system's use in fields including cybersecurity, digital forensics, and social media monitoring.

In summary, the proposed framework provides a dependable, effective, and useful way to detect manipulated media. It helps to ensure the legitimacy of online content, reduce false information, and build digital trust. In addition to highlighting artificial intelligence's potential for practical application, this work underscores the significance of using it for meeting emerging media forensics challenges.

VI. REFERENCES

1. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
2. I. Goodfellow et al., “Generative Adversarial Networks,” Advances in Neural Information Processing Systems (NeurIPS), 2014.
3. H. Nguyen, J. Yamagishi, and I. Echizen, “Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
4. Y. Li and S. Lyu, “Exposing DeepFake Videos by Detecting Face Warping Artifacts,” IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
5. D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: A Compact Facial Video Forgery Detection Network,” IEEE International Workshop on Information Forensics and Security (WIFS), 2018.

6. T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

7. OpenCV Documentation, "Open Source Computer Vision Library," Available: https://opencv.org/

8. FastAPI Documentation, "FastAPI Framework," Available: https://fastapi.tiangolo.com/

9. TensorFlow Documentation, "An End-to-End Open Source Machine Learning Platform," Available: https://www.tensorflow.org/

10. Kaggle Dataset, "Deepfake Detection Challenge Dataset," Available: https://www.kaggle.com/