International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

An API-Integrated CNN–RNN Framework for Scalable Deepfake Detection

Rajdeep Paul[1] , Biswajit Dey[2]

[1]Department of Computer Application, Techno India University, West Bengal-700091, India
[2]Department of Computer Application, Techno India University, West Bengal-700091, India

---

Abstract - Deepfake technology has been rapidly advancing, posing significant threats to media authenticity, cybersecurity, and public trust. It's a serious threat to identity verification in the digital medium. To tackle and solve this serious problem a deepfake detection approach is taken. A Convolutional Neural Network (CNN) algorithm named Resnext and a Recurrent Neural Network (RNN) algorithm named Long Term Short Memory (LTSM) is used to train a deepfake detection model. The whole approach and process is discussed. The model accuracy obtained is 91% using Celeb-Df dataset, Then the integration concept is discussed and how we can use this model as an Application Programming Interface (API) service for platforms or users. The model can be accessed through API to detect deepfakes and provide accurate output to validate authenticity of digital content. This work not only proposes a deepfake detection solution but also tries to practically implement the deepfake detection research outcome for real world use cases.

Key Words: Deepfake detection, CNN, RNN, API integration, ResNext, LTSM.

1. INTRODUCTION

The recent rapid growth in Artificial Intelligence(AI) technologies have been used to create hyper-realistic synthetic media. Those altered media are used to tamper or manipulate original multimedia content all over the internet. This altered media content is a serious concern in privacy and known as the term “Deepfakes”. Even AI-enabled software tools like FaceApp [1] have been used to create realistic-looking face swapping in images and videos. Deepfake content is created by combining original existing media and targeted content (mostly images) through Deep Learning techniques [2].

The term “Deepfake” is derived from “Deep Learning (DL)” and “Fake”, and it describes specific photo-realistic video or image contents created with Deep Learning support. To generate such altered media content, deep learning techniques like generative network and discriminative network with FaceSwap techniques are used. The generative network uses an encoder with a decoder to create a fake synthesis image. The discriminative network describes the authenticity of the newly generated content. The combination of those two networks is called Generative Adversarial Networks (GANs), proposed by Ian Goodfellow. [3]

Addressing and detecting the deep learning generated altered media is known as deepfake Detection. Several deepfake detection techniques have been proposed. This paper aims at developing a detection model to detect such deepfake content effectively by responding if an input video or image is real or fake and after that proposed techniques to use them practically in the real world.

The below Fig. 1 depicts real images in the first column and deepfake images in the next two columns.

Fig. 1: Deepfake images [4]

Table of Abbreviation:

Abbreviation	Definition
AI	Artificial intelligence
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LTSM	Long Term Short Memory
API	APPlication programming Interface
DL	Deep Learning
ML	Machine Learning
GANs	Generative Adversarial Network
MERN	MongoDB, Express.js, React and Node,js
ViTs	Vision Transformers
SAM	Self-Attention Mechanism
REST API	Representational State Transfer Application Programming Interface
SVM	Support Vector Machine

CNB	Complement Naive Bayes
AUC Curve	Area Under the Curce

2. RESEARCH MOTIVE

The rapid advancement of Deepfake technology has introduced serious challenges in detecting manipulated or synthesized media, which raise the concern about misinformation, privacy violations and security threats. The increased accessibility of tools to generate highly realistic fake images and videos has grown. So it has become an essential responsibility to develop robust and reliable detection methods. The motivation behind this research is to contribute to the fight against digital and forensic exception by building an accurate and efficient deepfake detection model. By leveraging a hybrid CNN-RNN architecture, combined with pretrained models, this study aims to enhance detection accuracy and address the growing need for authentic trustworthy media authentication in the digital AI era.

3. LITERATURE REVIEW

David Guera, et al. [5] have proposed a temporal aware detection pipeline which combines the Convolutional Neural Network(CNN) and Recurrent Neural Network(RNN) to identify deepfake content. First the CNN model extracts per-frame spatial features and then the result is analyzed to find inconsistency of the data frames sequentially by RNN model. It used large amounts of deepfake data from several platforms. It gives a competitive deepfake detection accuracy. This work contributes to the growing research of deepfake detection.

Hasam Khalid, et al. [6] have demonstrated reframing deepfake detection as a one-class anomaly detection task. One such method is OC-FakeDet, which leverages a one-class Variational Autoencoder(VAE), that is trained exclusively on real face images. The method can bypass the need for fake data by identifying synthetic content as anomalies. It presents that all preliminary tests on the FaceForensics++ benchmark demonstrate promising results with accuracy rate of 97.5% on the NeuralTexture subset without utilising any fake images during their training.

Afchar, et al. [7] proposed a lightweight deep-learning method to detect the synthetic media, which is generated using the deepfake and the Face2Face technique using facial manipulations at a mesoscopic scale. Their approach employs two architectures: a Meso-4 network, a four layer convolutional network with pooling and a ReLU-activated dense layer for generalization and a Mesoinception-4 network, which uses inception modules into Meso-4's layers to enhance feature extraction. This method achieves efficient detection with less computational complexities, demonstrating the viability of shallow networks.

Bestagini, et al. [8] have proposed a local tamper detection method that identifies manipulated spatial temporal regions in video sequences. It analyzes residual traces between neighboring frames. Unlike global detection frameworks, their approach focuses on local anomalies to pinpoint altered areas. The algorithm finds inconsistencies in motion and texture patterns. The method is evaluated on the SULFA database and contains realistic tampered videos.

Bekci et al. [9] proposed a deepfake detection framework that merges metric learning and steganalysis-inspired models for enhanced generalization against unseen manipulation. It provides 5–15% gains in accuracy on benchmarks such as FaceForensics++, DeepFakeTIMIT, and CelebDF and especially for minor or hidden modifications. At the same time, Li et al. [10] investigated behavioral cues for detection that detects inconsistency in eye-blinking patterns between real and fake videos. By learning this said observation, they crafted a specialized approach to identify unnatural blinking patterns in deepfakes that illustrates the discriminability potential of physiological signals. These contributions point to varying strategies from metric-based steganalysis to behavioral biometrics, which tackles burgeoning issues in synthetic media detection.

Ciftci et al. [11] introduced an innovative method to track the source of deepfake content by examining biological signals within residuals. This innovative paper was the first to use biological indicators in detecting the source of deepfakes. Experimental evaluations were conducted on the Face Forensics++ dataset, including several ablation tests to confirm their method's validity. Interestingly, they achieved an impressive accuracy rate of 93.39% in source identification for four different deepfake generators. These findings highlight the effectiveness of their proposed method and its promising capability to trace the origins of deepfake content with accuracy.

Raturi's architecture of 2018 [12] was devoted to detecting fake accounts from different social networks. This study deployed machine learning techniques to identify fake Facebook profiles based on their posts and activity on their walls. The proposed system utilized Support Vector Machine (SVM) and Complement Naïve Bayes (CNB) algorithms for text classification and data analysis, with special regards to the identification of offensive words and their repetition tracking. SVM achieved a 97% accuracy rate while CNB achieved 95%, based on the Bag of Words (BOW) model when detecting fake account cases. This research primarily raised the problem of less data validation before the content gets published, thus leaving social networking sites with a vulnerability of counterfeit accounts.

The findings of Bunk et al. (2017) [13] introduce two systems aimed at detecting and localizing counterfeit photographs. The proposed systems hinge upon resampling properties of images when combined with deep learning techniques. The first system applies radon transform to analyze resampling properties over an overlapping image region. Deep learning classifiers will use a Gaussian conditional domain pattern to generate a heat map of manipulated areas and a Random Walker segmentation method to finally describe all the affected regions. The second system relies on using software resampling properties in overlapping image patches, feeding them into an LSTM based network for long-term memory analysis. They were generally assessed for detection and localization accuracy against effective results in identifying and addressing digital image forgeries.

Aphiwongsophon and Chongstitvatana also used counterfeiting news for studying automated learning techniques [14]. Three common methods were tried in their experimental viewpoint: Naïve Bayes, Neural Network, and Support Vector Machine (SVM). The data was cleaned and normalized before being subjected to model application, thus ensuring accurate classification. Naïve Bayes yielded an accuracy of 96.08% in identifying false news, while the Neural Network and SVM methods gave impressive recognition rates of 99.90% for their combined applications of two in fake news detection.

Kim and Lee [15] elegantly put forward that digital forensics tools help find manipulated and fake images employed for illegal activities. Therefore, this study designed an algorithm with deep learning technology that performs impressively. The first step in the approach applies a convolutional neural network (CNN for the image processing phase. To expose hidden features, a high-pass filter is used rather than stripping away the semantic information. For their experiments, the researchers performed the generation of modified images by the application of intermediate filtering and Gaussian blurring while introducing white Gaussian noise enabling the model to work efficiently in spotting fake image alterations

4. TECHNIQUES

There are various types of deepfake detection techniques available, widely used in many fields.

i. **Machine Learning-Based Methods:** Traditional machine learning (ML) algorithms are very important for interpreting decision logic in human terms, making them effective for deepfake detection. Tree-based models like Decision Tree and Random Forest visualize decision-making, enhancing the explainability. Generative adversarial networks (GANs) generate realistic fake faces, while ML methods identify irregularities in GAN-generated content. Deepfake techniques often alter facial features (e.g., eye shade) to deceive viewers. Advanced methods combine multiple features and use facial landmarks (e.g., eyes, nose) for authenticity checks. Studies show ML methods can detect deepfakes with up to 98% accuracy, though performance varies based on similarity and quality of the dataset.

ii. **Deep Learning Based Methods:** Deep learning methods are widely used to detect Deepfake artifacts by identifying inconsistencies in image generation pipelines. Techniques include GAN simulators that replicate fake-image patterns and classifiers that detect them. Networks using RGB data and physiological measurements, like heartbeat patterns, enhance accuracy. Deep CNNs outperform shallow models, extracting facial features such as eye movement and lip motion. Advanced approaches apply attention mechanisms, capsule networks, and ensemble learning to boost accuracy, sometimes exceeding 99%. Frame-by-frame analysis, RNNs, and optical flow techniques improve detection. Pixel-wise masks and adversarial training further enhance model robustness against facial manipulations.

5. TECHNOLOGIES

- **Python**

Python is an object-oriented, interpreted, dynamically typed, and high-level language with a rich base of applications in software development and automation, data analysis, machine learning, and artificial intelligence. Python is one of the most popular programming languages, thanks to its ease of use, simplicity, and strength, particularly in the area of machine learning and deep learning.
Python has a vast library and framework collection like NumPy, SciPy, and Scikit-learn that form the foundation of scientific computing and data manipulation. It also has deep-learning/machine-learning-specific frameworks such as TensorFlow, PyTorch, and CNTK that enable efficient model development and deployment. These frameworks ease model construction, automate the training process, and enhance the overall efficiency of machine learning pipelines.
The ease of readability of Python and simple syntax make Python a great language for rapid prototyping and iterative development. Python enables rapid testing of complex algorithms, which is useful to developers and researchers and also makes the language very easy to comprehend for non-programmers. The vast community support and large standard library further make Python the first choice language for AI and ML-based applications.

- **PyTorch**

PyTorch is an extremely optimized tensor library for deep learning applications. It is Torch, a scientific computing library for Lua, but with the power and simplicity of Python. PyTorch is backed by Facebook's AI Research team and is an open-source library that has been widely used due to its dynamic computational capabilities, ease of debugging, and Pythonic architecture.
The core of PyTorch is its tensor library, which provides high-performance numerical computation. PyTorch tensors are the same as NumPy's n-dimensional arrays but with extra features such as GPU acceleration, hence making it the library of choice for deep learning applications. PyTorch also provides automatic differentiation with its native module, autograd, which makes gradient computation to be run smoothly, a critical part of training neural networks.
The most critical feature of PyTorch is its dynamic computation graph, which makes the model architecture changeable at runtime, hence making it extremely dynamic compared to TensorFlow's static graph-based system. PyTorch is also renowned for using the imperative paradigm of programming, hence making it natural and less error-prone compared to traditional imperative programming in frameworks such as TensorFlow. PyTorch also provides better memory efficiency, hence making it the tool of choice for humongous-scale deep learning applications.
In comparison to TensorFlow and Keras, PyTorch provides a Pythonic experience and is a notch above compared to writing deep learning models from scratch. It has gained a lot of traction among research groups and industry experts due to ease of experimentation, strong debugging capabilities, and maximum utilization efficiency of the GPU.

6. MODEL TRAINING METHODOLOGY

- **Prepare Dataset:** The model is trained using a dataset like CeleDf v2. The total dataset includes around 990 real videos and 5639 deepfaked videos followed by multiple genders and ages.

Fig. 2: Fake Images from Dataset

Fig. 3: Real Images from Dataset

- **Frame Extraction:** The individual frames are extracted from the video datasets. Specific frame rate can be chosen to balance the data volume.
- **Pre-processing Dataset:** To pre-process the dataset, they need to be processed by some tools like Matplotlib, to crop clearly and center the face. After that the images need to be rescaled for better visibility. Discard the no face detected frames.
- **Data Splitting:** The images were resized to 256 pixels and divided into training, validation, and test sets. The training set contained 100,000 images, while the validation and test sets each included 20,000 images. To maintain balance, both real and fake images were evenly distributed across the sets. Figure 4 displays a bar chart illustrating the distribution of images in the training, validation, and test sets after the split.
- **Pre-Trained Models** The datasets were prepared already. Now there are some pre-trained models available in Deep Learning. For this research, pre-trained model Resnext was used for the analysis. The analysis was monitored closely for the best performance, training and accuracy score with graphs.
- **Model Training:** The training phase for the model begins with the frame by frame data processing, where each extracted and preprocessed sequence frame is processed individually by CNN(Convolutional Neural Network). The CNN or ConvNets is a DL algorithm. It takes an image as input and divides the image into matrices. Then it assigns values to each part of the matrices to find differences from one to another image. It extracts spatial features such as textures, edges and pixel level inconsistencies.The CNN Resnext helps to enhance the efficiency and reduce the training time. The extracted feature maps are then fed into the RNN(Recurrent Neural Network) component.

Long Short Term memory(LSTM) is an RNN based algorithm which analyzes the temporal patterns across consecutive frames. It can learn order dependency in image sequence. RNN captures frame-to-frame inconsistencies, such as unnatural facial movements. The model is trained using a supervised learning approach, where real and fake labels guide the network's learning process. During training, loss functions like binary cross-entropy or mean squared error (MSE) are applied to minimize the gap between predicted and actual labels.

In short the model trained using Resnext and LSTM. First the video frames are loaded into the model as a test dataset for model training. After the training the testing dataset is used to test the model. We can see the accuracy in Fig. 5. A confusion matrix is achieved for the developed model. It is used to calculate the accuracy of the model.

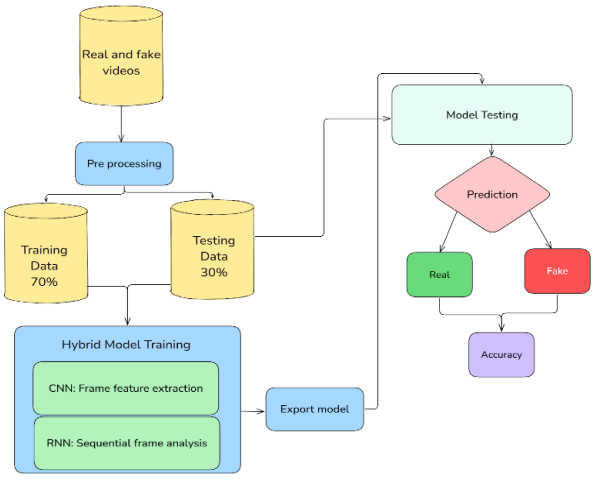

Fig. 4: CNN-RNN Model

- **System Used:** The model is trained using a personal computer with Intel Core i5. NVIDIA RTX 3050 (8GB VRAM), 40GB of DDR4 RAM, 450W of power supply. The used operating system is Windows 10(64-bit).

- **Model Outcome:** The graphs of loss and accuracy for both the training and validation sets shows the performance of the developed model. The loss graph in fig. 4 demonstrates that both the training and validation losses decreased over the epochs, and shows that the model has learned well. The starting training loss was approximately 0.55, and decreased to about 0.15. The validation loss exhibited a little bit of variance, but overall decreased to close to about 0.2. This demonstrates that the model generalized well, without overfitting.In the accuracy graph, the model performance improves steadily throughout the training process. The training accuracy starts at just under 77%. The accuracy for the validation set increased steadily and finished at around 91%, which indicates that the model has robust generalization. The bumps in accuracy for the validation set would be expected since the validation samples all exhibit different levels of complexity.

To summarize, the model accuracy is 91% by AUC score for the obtained confusion matrix. It is comparable with other models trained with the same

datasets as us. It is noticed that the proposed model got higher performance and accuracy.

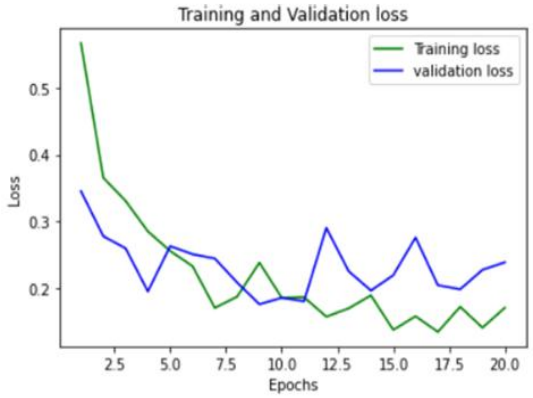

Fig. 5: Loss Graph

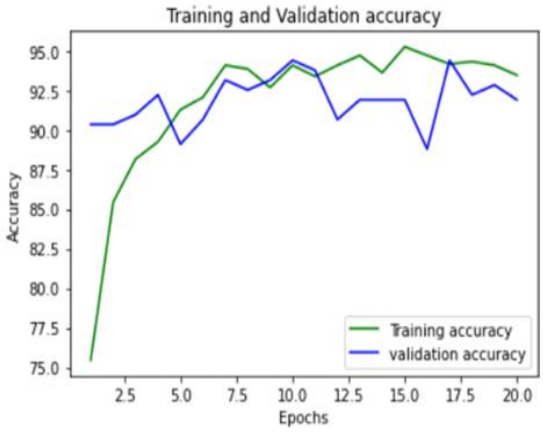

Fig. 6: Accuracy Graph

5. SURVEY ON PREVIOUS STUDIES

Deepfake detection has been an evolving discipline, with various CNN, RNN, and Transformer-based architectures achieving exclusive degrees of accuracy. Several past studies have evaluated detection strategies on datasets like FaceForensics, DeepFake Detection Challenge (DFDC), and Celeb-DF. Below are a few notable benchmarks:

Model Type	Dataset	Reported Accuracy
XceptionNet (CNN)	FaceForensics++	99.2%
EfficientNet (CNN)	Celeb-DF	96.3%
CNN-LSTM Hybrid	DFDC	92.8%
Vision Transformer (ViT)	FaceForensics++	97.5%
Multi-Modal Fusion (Audio + Video)	DFDC	93.6%

Key Takeaways:

- CNNs perform well in identity at the frame level, but struggle with low quality Deepfakes.
- RNNS (LSTMS, GRUs) improves temporary stability detection, causing CNN-RNN hybrids to become more effective.
- Transformer-based approaches gain popularity because of their ability to capture long-distance addiction.
- Multimodal approaches (analysis of both visual and sound objects) improve the detection of sophisticated deepfakes.

From the given above research outcome hybrid architectures integrating CNN for spatial feature extraction and RNN for temporal coherence analysis have demonstrated accuracy starting from 92% to 97%.

7. API INTEGRATION

The whole idea is to introduce a High-level architecture to integrate the developed CNN-RNN deepfake detection model as an Application Programming Interface(API) service using technologies like MERN stack expertise.

MERN stack stands for Mongodb, ExpressJS, React and NodeJS. These are the technologies used to build scalable web services, from Client to server. It follows a REST API protocol to communicate between client and server.

The idea is to expose the deepfake detection model via a REST API so users (services or customers) can send media as input (e.g. video and image files) and get a prediction back.

7.1 API ARCHITECTURE

There are two main architecture options available considering privacy, security and scalability:

- **Monolithic Approach**

  Embedding the API logic into the **NodeJS server backend** and we can call the deepfake detection model from Node using a **child process**.

- **Microservice Approach**

  Wrapping the detection model with its own **python based API (using Flask or Fast API)** and having the **NodeJS backend** forward the **client requests** to that service via **HTTP (HyperText Transfer Protocol)**. This separation can improve the scalability and isolate the heavy machine learning processing.

I am following the microservice approach for scalability, security and reliability.

7.2 HIGH LEVEL ARCHITECTURAL COMPONENTS

Each component here is equally important and dependent on each other. It describes how the input video or image file is processed and gets labeled as an original or a deepfake video or image.

- **Client/Frontend (Optional):**
  It is the part where any video or image file is given as a input to check it's deepfake label. Any third party service or company can set this client/frontend, as our main goal is to provide deepfake detection API to those company for their own use cases and that can be many. I will discuss them on the paper later.
- **Python Deepfake detection Microservices:** In this component we load the detection model , process the file and send the accuracy outcome as a output. It is built using dependencies like Flask or FastAPI. An API endpoint has to be created here so it can accept the file uploads from Node.JS server. Here we Loads the pretrained CNN–RNN model. Then the model processes incoming media and returns a prediction result. Sends the results as a response to the Node.JS server. After an API process is done we containerize the python service API with used dependencies for docker use.
- **Node.JS API Gateway:** This component is responsible for receiving the client request and redirecting it to the python deepfake detection model API. This component acts as the entry point for client requests. It Handles file uploads and routing. Then it Forwards the media file to the Python microservice. After creating the Node.JS API, a docker file is created for docker containerization with the used dependencies.
- **Docker Compose (containerization) for Deployment & Scaling:** Each microservice is containerized with Docker. Docker Compose is used to manage and deploy the containers. First a docker-compose.yml file is created, which will define the two services (python and Node.JS) and ensure that they can communicate with each other via a common network protocol.

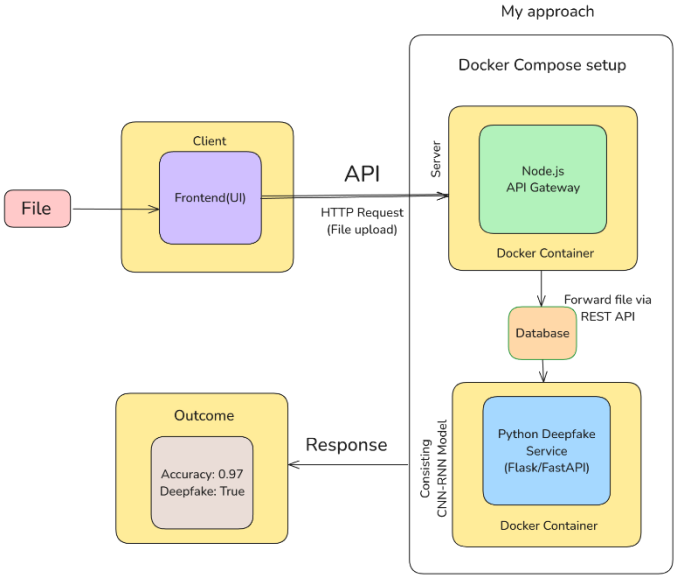

Fig. 7: Process Model of API Architecture

- **Cloud Deployment:**
  In this component we deploy the whole service in cloud services like AWS ECE, Kubernetes for production scalability and accessibility.
- **Logging & Monitoring**

Then we implement a logging mechanism and monitoring solution to track our service health.

8. USE CASES

The proposed deepfake detection system integrates a hybrid CNN-RNN model, designed for distribution as a scalable API service. This API can be integrated into many industries, safety, fraud prevention and material authenticity can be increased. Below are the most important applications where this model can be used effectively:

- **Social media and material moderation**
  With the emergence of AI Media, social media platforms require automatic moderation tools. Deepfake Detection APIs can be built into platforms such as YouTube, Facebook and Instagram to scan the video that is really uploaded. This ensures that the media is manipulated or removed before it reaches a wide audience marked or removed, which reduces the spread of misinformation.
- **Video conferences and certification of real -time**
  Video conference applications such as Zoom, Google Meat and Microsoft Team can integrate this API to confirm real-time participating identity. By analyzing video frames for indications of Deepfake manipulation, the system can increase the safety of virtual meetings, prevent modeling and identity fraud.
- **Financial fraud detection and identity confirmation**
  In the financial sector, Deepfake Technology KYC (knowing your customer) causes risks in verification and online banking. Banks and fintech services can use the API to validate the identity confirmation video, and ensure that fraud actors cannot utilize AI-public media to circumvent security checks.
- **Law enforcement and digital forensic**
  The authenticity of video evidence is important in legal and criminal investigation. Law enforcement agencies and forensic analysts can appoint API to check video events for indications of molesting or manipulation. This will help confirm the reliability of integration digging
- **News and Journalism Facts**
  Media organizations face challenges in verifying the authenticity of viral video. The proposed API can be used by news agencies such as the BBC, CNN to assess video sources before publishing. It helps to prevent the spread of misleading deep-stripped news, and maintains the integrity of journalism.
- **Online dating and prevention of social networks scams**
  The use of Deepfake-related videos for catfishing and deception of identity increases on social platforms. Dating apps like Tinder can confirm the user profile video by integrating Deepfake Detection APIs, and

ensuring that users do not manipulate their appearance using AI-generated material.

- **AI-based employment platforms and verification** Online job application platforms rely on quick video-based CV submissions. Companies such as LinkedIn and actual applicants can integrate API to confirm the authenticity of the video. This feature ensures that decision-making decisions are based on the actual submission instead of manipulating AI-related material.

9. FUTURE WORK AND IMPROVEMENTS

While the proposed hybrid CNN-RNN model acquires high accuracy in deepfake detection, many areas of improvement can be detected to increase its efficiency, scalability and real world's praise. This section emphasizes potential progress that can further refine the performance of the model and expand the projection.

9.1 Transformer-based deepfake detection

Recent progress in computer vision suggests that the Vision Transformers (ViTs) and Self-Attention Mechanisms can improve CNN in some tasks, especially when capturing global relevant information. While CNN is excellent in extracting spatial features from video frames, they can struggle with long -distance addiction.

- **Suggested improvement:** Change or improvement of CNN layer with a visual transformer (VIT) or Swin transformer to increase the extraction function.
- **Expected Benefits:** Improvement in the Generalization of Deepfake Manipulations.

Potential experiments: We can compare a hybrid VIT-RNN model against the current CNN-RNN approach on the same dataset.

9.2 Edge AI Adaptation

Adequate calculation resources are required to detect deepfake on cloud-based APIs. However, many applications in the real world (e.g., social media platforms, mobile authentication, video calls) require low-latency, on-device deepfake detection rather than cloud-based processing.

- **Suggested improvement:** To activate mobile design and built-in equipment, adjust the model using Tensorflow Lite, Onnx Runtime or Nvidia TensorRT.
- **Expected benefits:** Dependency available on cloud services can increase access to low, short time, short time and real-time applications.

Possible experiments: When we distribute the model on a smartphone GPU (e.g Android/iOS), we evaluate the estimation speed and accuracy business.

9.3 Multimodal Deepfake Detection (Video + Audio + Text)

The current deepfake detection models are mainly dependent on visual data, but deepfake videos often show deviations in sound and speech patterns. By incorporating several types, the system can improve accuracy.

- **Suggested improvement:** We can integrate audio deepfake detection (e.g analyzes of voice deviation using a spectrogram-based CNN or WAV2Vec model) to experiment and improve.

  Using lip-Sync analysis (comparison of facial activities with speech to detect mismatches).

  Including text-based relevant identity (false video analyzes transcript using NLP to detect unnatural speech patterns).
- **Expected benefits:** High strength against deepFake techniques that manipulate the video just keeps the sound unchanged.

Possible experiments: We can train a multimodal deepfake Detector using both video and audio features, and then compare the performance with a visual cavalry approach.

9.4 Continuous Education and Auto-Update System

The deepfake technology develops rapidly, which requires a model that detects instead of dynamically stable to detect.

- **Suggested improvement:** Can use a self-learning API that continuously updates the basis of its knowledge by using feedback from details in the real world.

  Distribute Federated Learning to Train Models in distributed data sources without centralized storage.

  Introduce online learning pipes where the system automatically involves new deepfake variations in training kits.
- **Expected benefits:** API is constantly updated and effectively against new deepfake techniques without the need for manual withdrawal.

Possible experiments: Develop a feedback loop where the API flagged uncertain cases and left behind using new deepfake.

10. CONCLUSION

This paper first explores the development of CNN–RNN models for deepfake detection, resulting in a good accuracy, drawing from both foundational deep learning research and recent studies. The research states that by combining the strengths of CNNs and RNNs with the containerize approach we can use the model as an API service for forensics, news, social media and identity validation platforms to validate content authenticity. We have outlined an API architecture that

can be used for practical work. This study provides a clear direction for further research. In the long run, we aim to enhance deepfake detection by incorporating attention mechanisms, multimodal fusion, and self-supervised learning. As more resources become available, we look forward to testing these ideas experimentally and contributing valuable insights to the world for authenticity.

ACKNOWLEDGEMENT

I would like to acknowledge the open-source communities and developers behind the frameworks and tools used in this project, including PyTorch, Flask/FastAPI, Docker, and Node.js, whose contributions were essential in building and deploying the deepfake detection system.

I would like to extend my heartfelt thanks to the authors of the paper, "Deepfake Detection in Digital Media Forensics" by Vurimi Veera Venkata Naga Sai Vamsi, Sukanya S. Shet, Sodum Sai Mohan Reddy, Sharon S. Rose, Sona R. Shetty, S. Sathvika, Supriya M. S., and Sahana P. Shankar from the Department of Computer Science and Engineering, Ramaiah University of Applied Sciences, Bengaluru, India. Their insightful research and contributions significantly influenced and inspired the methodology and direction of this project.

Finally, I am immensely thankful to my family and friends for their unwavering support, motivation, and patience throughout this journey.

This research would not have been possible without the collective support and contributions of all the aforementioned individuals and groups. Thank you.

REFERENCES

1. FaceApp. Accessed: Jan. 4, 2021. [Online]. Available: https://www.faceapp.com/
2. J.P. Nayak, K. Anitha, B.D. Parameshachari, R. Banu, P. Rashmi, PCB Fault detection using Image processing, IOP Conference Series: Materials Science and Engineering, 225, IOP Publishing, 2017, August .
3. I. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS), vol. 2. Cambridge, MA,USA: MIT Press, 2014, pp. 26722680.
4. J. E. Solsman, "Deepfakes' threat to the 2020 US election isn't what you'd think," CNET, 2020. [Online]. Available: **https://www.cnet.com/news/politics/features/deepfakes-threat-2020-us-election-isnt-what-youd-think/**
5. D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6, doi: 10.1109/AVSS.2018.8639163.
6. Khalid, Hasam, and Simon S. Woo. "Oc-fakedect: Classifying deepfakes using one-class variational autoencoder." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.
7. D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 2018, pp. 1-7, doi: 10.1109/WIFS.2018.8630761.
8. P. Bestagini, S. Milani, M. Tagliasacchi and S. Tubaro, "Local tampering detection in video sequences," 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, Italy, 2013, pp. 488-493, doi: 10.1109/MMSP.2013.6659337.
9. Bekci, B.; Akhtar, Z.; Ekenel, H.K. Cross-Dataset Face Manipulation Detection. In Proceedings of the 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 5–7 October 2020; pp. 1–4.
10. Li, Y.; Chang, M.-C.; Lyu, S. In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking. In Proceedings of the 2018 IEEE
11. U. A. Ciftci, İ. Demir and L. Yin, "How Do the Hearts of Deep Fakes Beat? Deep Fake Source Detection via Interpreting Residuals with Biological Signals," 2020 IEEE International Joint Conference on Biometrics (IJCB), Houston, TX, USA, 2020, pp. 1-10, doi: 10.1109/IJCB48548.2020.9304909.
12. R. Raturi, (2018). Machine Learning Implementation for Identifying Fake Accounts in Social Network. International Journal of Pure and Applied Mathematics, 118(20), 4785-4797.
13. J. Bunk, J. Bappy, H. Mohammed, T. M. Nataraj, L., Flenner, A., Manjunath, B., et al. (2017). Detection and Localization of Image Forgeries using Resampling Features and Deep Learning. University of California, Department of Electrical and Computer Engineering, USA.
14. S. Aphiwongsophon, & P. Chongstitvatana, (2017). Detecting Fake News with Machine Learning Method. Chulalongkorn University, Department of Computer Engineering, Bangkok, Thailand.
15. D.-H. Kim, & H.-Y. Lee, (2017). Image Manipulation Detection using Convolutional Neural Network. International Journal of Applied Engineering Research, 12(21), 11640-11646.