International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

ADVANCING DEEPFAKE DETECTION: MOBILE APPLICATION WITH DEEP LEARNING

Arun KS[1*], Juan Shaji Austin[2*], Kiran Paulson[3*], Kevin Paulson[4*], Suzen Saju Kallungal[5*]

[1,2,3,4]BTech.Students,Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala

[5]Assistant Professor,Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam, Kerala

---

Abstract - This paper introduces a pioneering approach to deepfake detection leveraging the power of mobile platforms through Flutter. We combine state-of-the-art ResNeXt and LSTM models for robust deepfake identification and en- capsulate them within a user-friendly mobile interface. By harnessing the versatility of Flutter's InAppWebView, our so- lution seamlessly integrates with mobile devices, empowering users with real-time deepfake detection capabilities on their smartphones. Through rigorous evaluation, we demonstrate the efficacy and usability of our approach, marking a significant advancement in the field of mobile-based deepfake detection.

Index Terms-Flutter, Dart, Inappwebview, LSTM, ResNext

I. Introduction

In recent years, the proliferation of deepfake technology has raised significant concerns regarding the manipulation of digital content and its potential consequences on various aspects of society, including politics, media, and personal privacy. Deepfakes, which refer to synthetic media generated using advanced machine learning techniques, have become increasingly sophisticated, making it challenging to distin- guish between real and manipulated content. As a result, the need for effective deepfake detection mechanisms has become paramount to combat misinformation and protect individuals and organizations from the harmful effects of manipulated media.

Traditional approaches to deepfake detection have primarily relied on complex algorithms and computational resources, often restricting their deployment to high-performance com- puting systems. However, with the widespread adoption of mobile devices, there arises an opportunity to democratize deepfake detection by bringing it to the fingertips of everyday users. Mobile platforms offer unparalleled accessibility and convenience, allowing users to verify the authenticity of media content, irrespective of their location or technical expertise.

In this context, this paper presents a groundbreaking approach to deepfake detection tailored specifically for mobile platforms. Our methodology combines cutting-edge deep learning architectures, namely ResNeXt and LSTM, to effectively identify and classify deepfake content. Leveraging the versatility and efficiency of the Flutter framework, we encapsulate our deep learning models within a user-friendly mobile application, providing users with a seamless and intuitive experience for deepfake detection.

The integration of Flutter's InAppWebView further enhances the functionality of our mobile application, enabling users to analyze online media content directly within the app interface. By harnessing the power of mobile devices, our solution empowers users to verify the authenticity of media content, thereby mitigating the spread of misinformation and preserving the integrity of digital communication channels.

Throughout this paper, we provide a comprehensive overview of our deepfake detection methodology, detailing the architecture, implementation, and evaluation of our mobile application. We present experimental results demonstrating the effectiveness and efficiency of our approach, as well as discuss the potential implications and future directions of mobile- based deepfake detection technology. Ultimately, our research aims to contribute to the ongoing efforts in combating digital manipulation and fostering trust in the authenticity of media content in the mobile era.

II. BACKGROUND

A. Flutter

Flutter is an open-source UI software development kit created by Google, primarily used for building natively compiled applications for mobile, web, and desktop from a single codebase. It offers a fast development cycle, expressive UI components, and native performance across multiple platforms. Flutter utilizes the Dart programming language, developed by Google, known for its efficiency and simplicity. Dart is a statically typed language with features like hot reload,

enabling developers to instantly see changes in their code reflected in the app during development. Together, Flutter and Dart provide a powerful combination for creating cross-platform applications with high performance and a smooth user experience.

B. ResNext

ResNeXt, short for "Residual Next," is a deep neural network architecture developed by Facebook AI Research (FAIR) researchers. It builds upon the ResNet architecture's success in computer vision by introducing a novel parameter called "cardinality." This parameter controls the network's width and enhances parameter efficiency. Unlike traditional methods that increase model capacity by adjusting depth or width, ResNeXt boosts performance by adjusting cardinality, result- ing in superior performance while maintaining computational efficiency. Its effectiveness in image classification tasks has made ResNeXt a widely adopted architecture in both research and practical applications, cementing its position as a pivotal framework in deep learning for computer vision.

C. LSTM

LSTM, or Long Short-Term Memory networks, emerged as a solution to the challenge of gradient disappearance in recurrent neural networks (RNNs). Developed by Hochreiter and Schmidhuber, LSTMs aim to effectively capture long-term dependencies in sequential data. Unlike conventional RNNs, LSTMs feature a memory cell with three gates: input, forget, and output gates. These gates regulate the flow of information, enabling LSTMs to selectively retain or discard data at each step. This capability allows LSTMs to efficiently capture distant relationships within the data. Widely employed in natural language processing, speech recognition, and time series prediction, LSTMs have become a fundamental element in deep learning models designed for sequential data analysis.

D. Python

Python stands out as a multifaceted high-level programming language celebrated for its user-friendly nature, clear syntax, and diverse utility. Crafted by Guido van Rossum, Python prioritizes human-readable code and facilitates multiple programming approaches, including procedural, object-oriented, and functional paradigms. Its rich standard library encompasses a vast array of modules and packages catering to various domains, such as web development, data analysis, artificial intelligence, scientific computing, automation, and beyond. Python's popularity thrives on its accessibility, robust community engagement, and extensive adoption across a spectrum of industries.

III. Literature Survey

[1] This paper discusses the evolution of AI technologies and the emergence of deepfake manipulation, primarily focusing on summarizing existing research on deepfake detection methodologies.This paper discuss the potential risks and challenges associated with the proliferation of deepfake technology, highlighting the importance of developing effective detection methods to mitigate the spread of misinformation and safeguard digital authenticity.

[2] Deepfake videos replace a person's features with those of another, often used for misinformation. While Generative Adversarial Networks (GANs) create deepfakes, they're not commonly used for detection. This study explores using GAN discriminators for detection, training them alongside MesoNet. Various discriminator architectures are tested on different datasets, and ensemble methods are proposed to boost performance. Results indicate GAN discriminators, even with ensemble methods, struggle with videos from unknown sources.

[3] This paper addresses the growing concern over the proliferation of deepfake media, particularly due to their potential misuse in activities like terrorism and blackmail. It highlights the advancements in deep generative networks, which have enabled the creation of realistic face swaps with minimal traces of manipulation. To tackle this issue, the paper proposes a method based on frequency domain analysis to detect subtle unnatural features in deepfake images that are imperceptible to the naked eye.

IV. Proposed Model

The proposed model is a cutting-edge mobile application for deepfake detection, uniquely developed using the versatile Flutter framework. Tailored for seamless user experience, this application harnesses the power of LSTM (Long Short- Term Memory) and ResNeXt architectures to discern deepfake manipulation in uploaded videos. Users can easily navigate the intuitive interface, upload videos, and initiate the deepfake analysis process, all within the familiar environment of their mobile devices. Additionally, users can access information related to deepfake technologies and detection methodolo- gies, as well as watch educational videos to enhance their understanding of the subject. Initially developed as web ap- plication, the model has been seamlessly integrated into a mobile application using Flutter's inAppWebView feature. Through the combination of advanced detection algorithms and user-friendly features, this mobile application offers a robust defense against the proliferation of deepfake content in the digital realm.

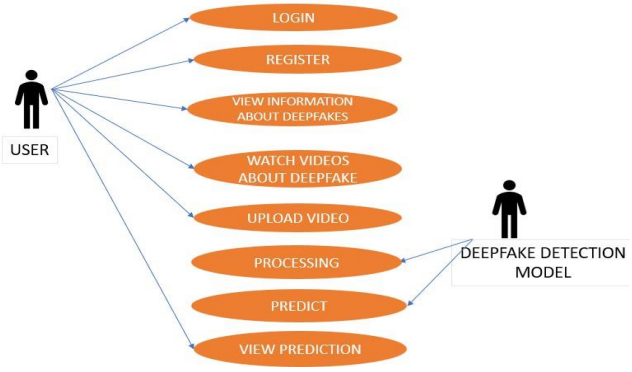

Fig. 1. Use case diagram

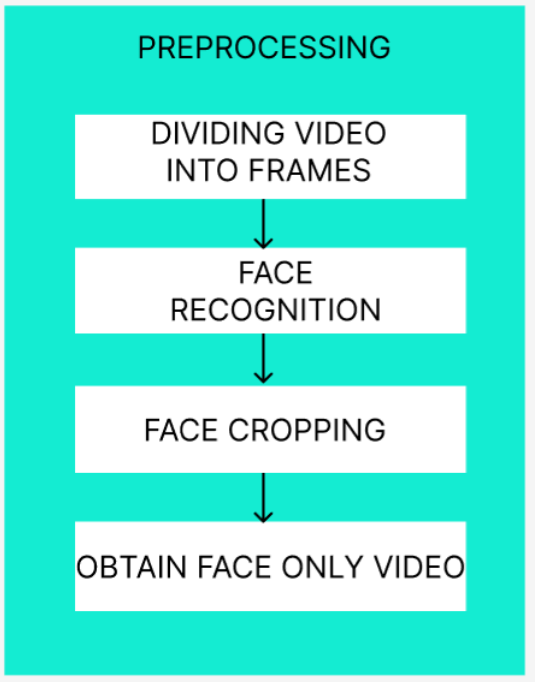

Fig. 2. Preprocessing Diagram

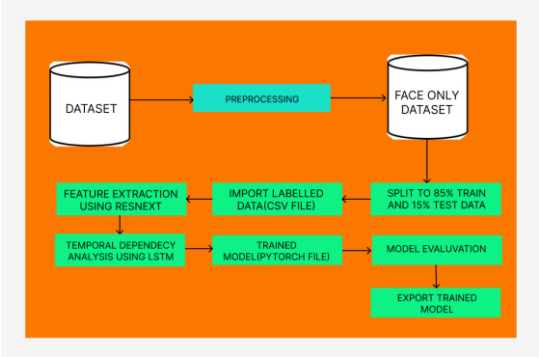

Fig. 3. Training the model Architecture

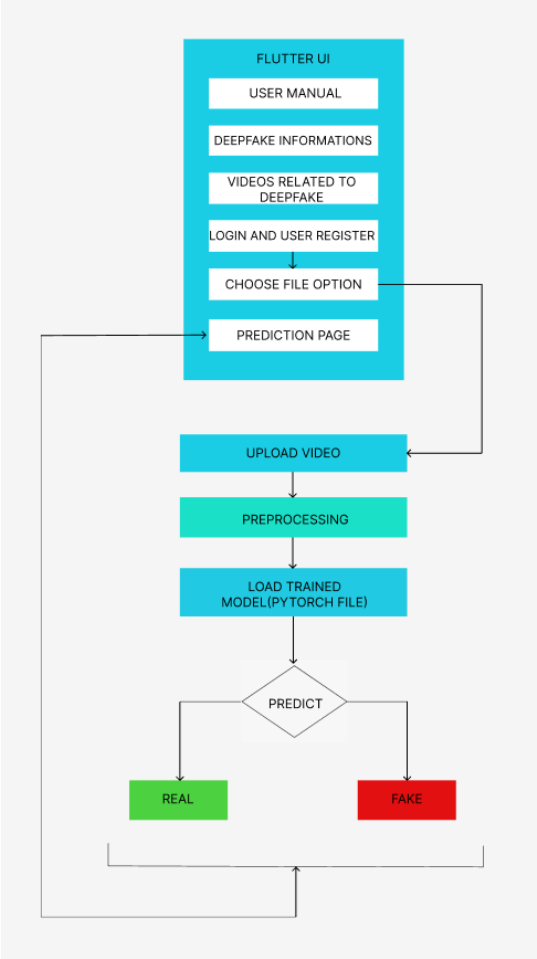

Fig. 4. App Architecture

V. METHODOLOGY

A. *Training the model*

1) Data preprocessing: After gathering datasets like CelebDF Version 1, CelebDF version 2, and DFDC, the initial step involved breaking down each video into individual frames using OpenCV. Following this, a Python face recognition module was employed to detect faces within these frames. Once faces were detected, they were isolated from the frames through a process called face cropping.

This methodology ensures that only the relevant facial re-gions are retained for further analysis, simplifying subsequent processing steps. With this approach, the dataset is primed for subsequent stages like feature extraction and model training, setting a solid foundation for robust deepfake detection sys- tems.

2) Data splitting for training and testing: The dataset was meticulously partitioned into two subsets: a training set comprising 85% of the data and a test set comprising the remaining 15%. This division ensures that the bulk of the data is allocated for training the model, allowing it to learn from a diverse array of examples. Meanwhile, the test set serves as an independent benchmark to evaluate the model's perfor- *mance, providing an unbiased assessment of its effectiveness in detecting deepfakes.*

3) ResNext: Enhancing Feature Extraction: ResNext stands out for its ability to capture rich and diverse feature repre- sentations within images. When applied to feature extraction in deepfake detection, ResNext analyzes facial images at various levels of granularity, discerning intricate details that distinguish between genuine facial expressions and manipu- lated ones. By leveraging its deep architecture and advanced convolutional layers, ResNext enhances the model's ability to detect subtle anomalies indicative of deepfake manipulation, thereby strengthening the overall accuracy and robustness of the deepfake detection system. The ResNeXt model, specifi- cally the ResNeXt50_32x4d architecture code, acts as a feature extractor for images.

4) Modeling temporal dependencies using LSTM: The LSTM layer takes as input a sequence of feature vectors extracted from video frames. Each feature vector represents the output of the ResNeXt model for a single frame.

During the forward pass, the LSTM processes each feature vector in the input sequence one by one, updating its internal state and generating an output hidden state for each time step.

The output hidden state at each time step captures the LSTM's understanding of the input sequence up to that point. The LSTM layer may have multiple layers, and each layer may have multiple hidden units. This allows the LSTM to capture complex patterns and dependencies in the input sequence.

The final output of the LSTM layer is typically the hidden state at the last time step, which summarizes the entire input sequence. This final hidden state is often used for making predictions or further processing downstream in the neural network.

B. Model Integration and Deployment

Developed and integrated a comprehensive deepfake detec- tion system that includes both a web application and a mobile application. The web application, created using HTML, CSS, and Django, allows users to register, log in, and upload videos for analysis.

C. Flutter

1) Registration: Create user accounts securely.
2) Login: Authenticating users securely.
3) Video upload: Users can select and upload a video file.
4) Preprocessing the video
5) Loading trained model: Next, the trained model is loaded into memory, ready to analyze the uploaded video data for any signs of deepfake manipulation.
6) Final result: Gets the final result whether the video is manipulated or not.

VI. Results

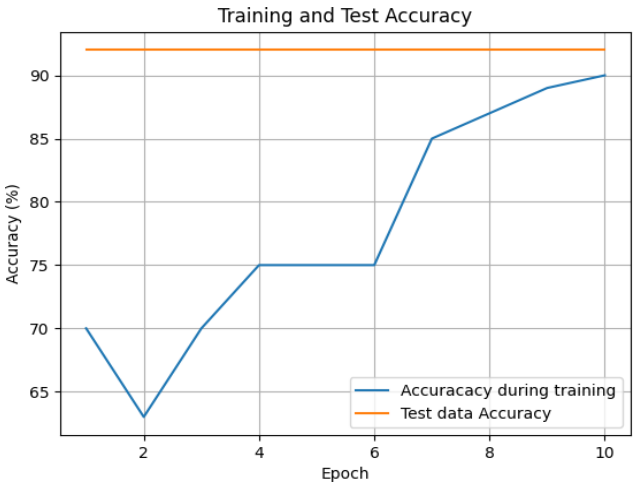

Fig. 5. Model Training and Testing

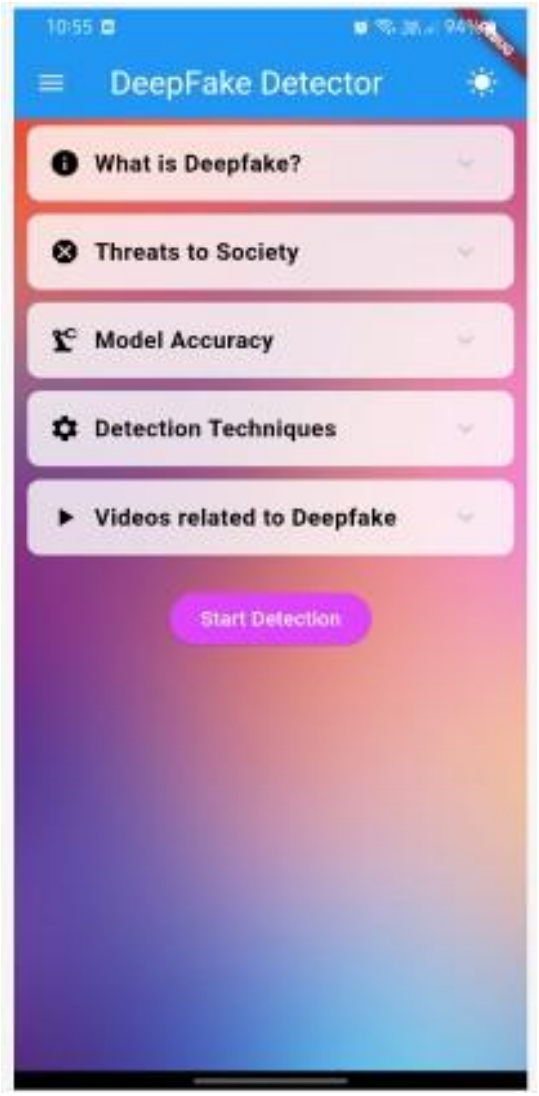

Fig. 6. Mobile App UI (1)

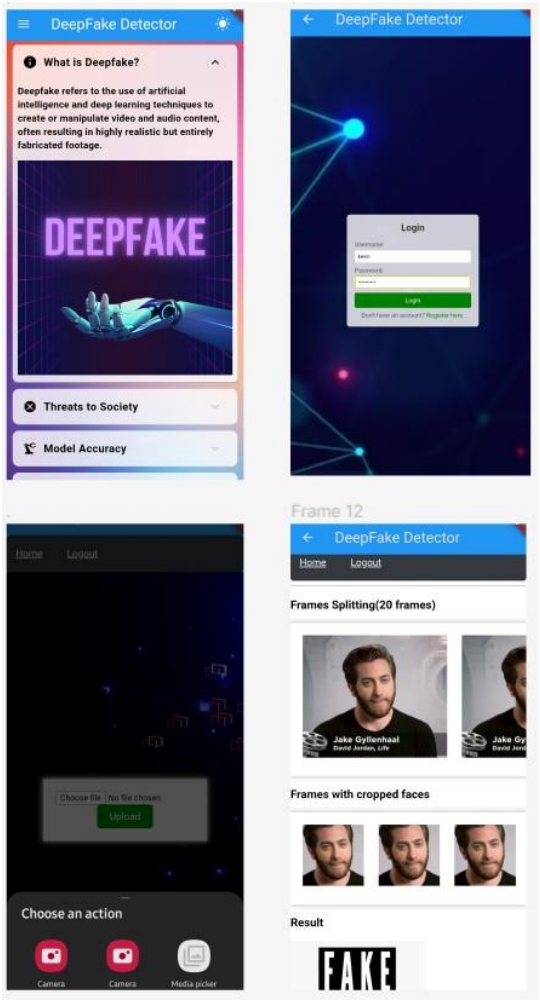

Fig. 7. Mobile App UI (2)

a) Image Descriptions::

1) **Model Training and Testing (Figure 5):** This visual representation presents the outcome of model training and testing. The horizontal axis denotes the training epochs or iterations, while the vertical axis signifies the performance metric, such as accuracy or loss. The graph illustrates the evolution of the model's performance over time, offering insights into its learning dynamics.

2) **Mobile App UI (1) (Figure 6):** This image showcases the user interface of the mobile application, emphasizing its user-friendly design. It provides users with intuitive controls and options for interacting with the deepfake de- tection system. The interface design focuses on simplicity and ease of use, ensuring a smooth experience for users regardless of their level of expertise.

VII. Future Scope

1) **Enhanced Scalability:**

   As computational resources advance, especially with powerful GPUs and better cloud infrastructure, the Deepfake Detection App can handle larger file sizes efficiently. This benefits users dealing with high-resolution or longer videos, expanding its utility.

2) **Improved Frame Extraction and Analysis:** Advancements in GPU technology can enable more efficient frame extraction and analysis, surpassing current limitations. This enhances detection accuracy, especially for videos with subtle manipulation.

3) **Multi-Faced Video Support:**

   Expanding the model's capabilities to detect and analyze videos with multiple faces broadens its applicability, particularly in scenarios like interviews or group settings.

4) **Enhanced Model Training with Larger Datasets:** Training the model on larger and diverse datasets improves accuracy and generalizability. Incorporating extensive datasets helps distinguish between genuine and fake videos, enhancing reliability in combating deepfake content.

3) **Mobile App UI (2) (Figure 7):** This depiction presents additional perspectives of the mobile application interface, highlighting supplementary features and functionalities. Users can seamlessly navigate through various sections of the app to access relevant information and controls, enhancing their overall experience.

VIII. conclusion

In conclusion, the development of a mobile application dedicated to advancing deepfake detection capabilities represents a significant milestone in the ongoing battle against manipulated media in digital environments. Through this project, we've successfully crafted a robust architecture that integrates cutting-edge deep learning algorithms with detection functionalities, facial recognition techniques, and an intuitive user interface. Our project showcases the efficacy of deep learning technology, particularly in its ability to identify and mitigate the spread of deepfake content with precision and ef- ficiency. By harnessing the power of sophisticated algorithms, we've laid the groundwork for a tool that empowers users to combat the threats posed by manipulated media, thereby safeguarding trust and authenticity in online media environ- ments. Furthermore, our work underscores the importance of continual innovation and adaptation in this rapidly evolving field. Moving forward, avenues for further exploration include enhancing offline capabilities, optimizing algorithms for mo- bile performance, and addressing potential limitations such as reliance on consistent internet connectivity. As we look to the future, it's clear that advancements in deepfake detection hold immense promise for bolstering the security and integrity of digital content. By leveraging the insights gleaned from this project and remaining committed to ongoing research and development, we can continue to advance the field of deepfake detection and contribute to the creation of a more secure and trustworthy online ecosystem.

References

[1] Rana, M. S., Nobi, M. N., Murali, B., & Sung, A. H. (2022). Deepfake Detection: A Systematic Literature Review. *IEEE Access*, 10,pp.31578–31593.

[2] Aduwala, S. A., Arigala, M., Desai, S., Quan, H. J., & Eirinaki, M. (2021). Deepfake Detection using GAN Discriminators. In *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)* (pp. 69-77). Oxford, United Kingdom. doi: 10.1109/BigDataService52369.2021.00014.

[3] H. Agarwal, A. Singh, and R. D, "Deepfake Detection Using SVM," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1245- 1249, doi: 10.1109/ICESC51422.2021.9532627.

[4] Khan, S. A., & Dang-Nguyen, D.-T. (2024). Deepfake Detection: An- alyzing Model Generalization Across Architectures, Datasets, and Pre- Training Paradigms. *IEEE Access*, 12,pp.1880–1908.

[5] Ramadhani, K. N., Munir, R., & Utama, N. P. (2024). Improving Video Vision Transformer for Deepfake Video Detection Using Facial Landmark, Depthwise Separable Convolution, and Self Attention. *IEEE Access*, 12,pp.8932–8939.

[6] Patel, Y. et al. (2023). Deepfake Generation and Detection: Case Study and Challenges. *IEEE Access*, 11, 143296–143323.

[7] Malik, A., Kuribayashi, M., Abdullahi, S. M., & Khan, A. N. (2022). DeepFake Detection for Human Face Images and Videos: A Survey. *IEEE Access*, 10,pp.18757–18775.

[8] Mitra, A., Mohanty, S., Corcoran, P., & Kougianos, E. (2021). A Machine Learning Based Approach for Deepfake Detection in Social Media Through Key Video Frame Extraction. *SN Computer Science*, 2.

[9] Pan, D., Sun, L., Wang, R., Zhang, X., & Sinnott, R. O. (2020). Deepfake Detection through Deep Learning. In *2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*.

[10] Gu¨era, D., & Delp, E. J. (2018). Deepfake Video Detection Using Recurrent Neural Networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*.

[11] Theerthagiri, P., & Nagaladinne, G. B. (2023). Deepfake Face Detection Using Deep InceptionNet Learning Algorithm. In *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1-6). Bhopal, India. doi: 10.1109/SCEECS57921.2023.10063128.

[12] Lyu, S. (2020). Deepfake Detection: Current Challenges and Next Steps. In *2020 IEEE International Conference on Multime- dia & Expo Workshops (ICMEW)* (pp. 1-6). London, UK. doi: 10.1109/ICMEW46912.2020.9105991.