“AI POWERED DETECTION OF DEEPFAKE MEDIA WITH REAL TIME INSIGHTS”

A Project Report Submitted in the partial fulfilment of requirement of the Degree

of

Bachelor of Technology in Artificial Intelligence & Data Science

to

RashtrasantTukdoji Maharaj Nagpur University, Nagpur

Under the guidance of

Guide

Prof. Mrs. U. A. S. Gani

Submitted by

Siddhi Chindhalore - 116 **Shreya Walde - 115**

Sanskruti Pote - 114 **Suchita Pawar - 225**

Shreya Ghoradkar -138

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PRIYADARSHINI COLLEGE OF ENGINEERING, NAGPUR - 440019

2024-2025

VISION OF INSTITUTE

To become one of the India's leading engineering institutes in both education & research. We are committed to provide quality & state of the art technical education to our student, so that they become technologically superior & in turn contribute for creating a great society.

MISSION OF INSTITUTE

i. Fostering a dynamic learning environment that equips students with Technical expertise, problem-solving skills, and a deep commitment to ethical practices.
ii. To cultivate a culture of innovation, incubation, research, and entrepreneurship that drives technological advancements.
iii. To uphold the spirit of mutual excellence while interacting with stake holders of our Institutional ecosystem.
iv. Promoting lifelong learning, professional growth and ensuring holistic development of students and the well-being of society.

VISION OF DEPARTMENT

Achieve academic excellence in the field of Artificial Intelligence and Data Science through innovative research ideas to produce competent professionals for the betterment of Society through most appropriate and Ethical practices.

MISSION OF DEPARTMENT

i. To impart quality education to students with foundation of Artificial Intelligence and Data Science Engineering through Outcome Based Education
ii. To provide a learning ambience to enhance problem solving skills, leadership qualities, team spirits and ethical responsibilities with a commitment to lifelong learning
iii. To bring out the competent and industry ready students by practicing theoretical aspects with experiential Learning.
iv. To collaborate with industry and premier institutes in terms of research and academics for upgrading knowledge and achieving go.

PRIYADARSHINI COLLEGE OF ENGINEERING, NAGPUR-440019

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that project report entitled - "**AI Powered Detection of Deepfake Media with RealTime Insights**" is a Bonafide work done by the students **Siddhi Chindhalore, Shreya Walde, Sanskruti Pote, Suchita Pawar, Shreya Ghoradkar**. The project report is submitted to **Rashtrasant Tukadoji Maharaj Nagpur University**, Nagpur in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Artificial Intelligence and Data Science** in Session 2024-2025.

Guide

Mrs. Umme Ayeman Gani
Assistant Professor

Dr. Snehal Golait
Head of Department

Dr. S. A. Dhale
Principal, PCE

PRIYADARSHINI COLLEGE OF ENGINEERING, NAGPUR-440019

Department of Artificial Intelligence and Data Science

DECLARATION

We, the undersigned, declare that the project entitled **"AI Powered Detection of Deepfake Media With Real Time Insights ",** being submitted in partial fulfilment for the award of Degree in Artificial Intelligence & Data Science, affiliated to Rashtrasant Tukadoji Maharaj Nagpur University, is the work carried out by us.

Place: Nagpur

Date:

PROJECTEES: -

Siddhi Chindhalore (116)
Shreya Walde (115)
Sanskruti Pote (114)
Suchita Pawar (225)
Shreya Ghoradkar (138)

ACKNOWLEDGEMENT

It gives us great pleasure to express our profound gratitude and sincere thanks to our guide, **"Mrs. U. A. S. Gani"** Professor of the Artificial Intelligence and Data Science Department, for their ongoing expertise and assistance in carrying out this dissertation work. They had supported us in this attempt from the beginning to the end.

We are grateful that the institute's facilities are provided by **Dr. S. A. Dhale**, Principal of Priyadarshini College of Engineering, Nagpur.

We express our gratitude to **Dr. Snehal Golait,** the department head of Artificial Intelligence and Data Science Department at Priyadarshini College of Engineering Nagpur.

We also take the opportunity to thank all, who have directly or indirectly extended help and encouragement in executing this project. Our sincere and healthy thanks.

PROJECTEES:

Siddhi Chindhalore (116)

Shreya Walde (115)

Sanskruti Pote (114)

Suchita Pawar (225)

Shreya Ghoradkar (138)

ABSTRACT

A number of methods for altering faces in films have been effectively created and made publicly accessible in recent years (e.g., Face Swap, deepfake, etc.). Using these technologies, it is possible to facilitate face video modifications with inaccurate results. It is employable in almost all fields. However, the overemphasis of all technologies is fatal prone to have a certain effect in society which may be negative (e.g., fake news, and cyberbullying revenge porn).

It is therefore important to be able to tell whether a person's face in a video has been modified, subjectively. To be able to address the problem of deepfake videos, we focus on the problem of face alteration detection in video sequences. In particular, we focus on the ensembles of several Convolutional Neural Network (CNN) models that have been developed.

The proposed methodology attains these objectives through the use of attention layers and data training powerful models derived from a base network, EfficientNetB4. By using two publicly available datasets and combining over 119,000 videos, we show how to be able to address these bezier curves, Detecting face alteration is a crucial field of computer vision remains a challenging task in most scenarios, but we demonstrate that in our case, the combined networks approach highly improves the results.

Keywords- Deepfake, Video Forensics, Deep Learning, Attention.

INDEX

Sr. No.	Content	Page No.
CHAPTER 1	INTRODUCTION	1 - 4
CHAPTER 2	LITERATURE REVIEW	5 – 6
CHAPTER 3	OBJECTIVE	7 – 8
CHAPTER 4	ARCHITECTURE	9 – 11
CHAPTER 5	ALGORITHMS	12 – 19
CHAPTER 6	DATA FLOW DIAGRAM	20 – 21
CHAPTER 7	SOFTWARE REQUIREMENTS	22 – 24
CHAPTER 8	IMPLEMENTATION	25 – 38
CHAPTER 9	CODING	39 - 41
CHAPTER 10	SCREENSHOTS	42 - 48
CHAPTER 11	PERFORMANCE EVALUATION	49 - 52
CHAPTER 12	APPLICATION AREA	53 - 54
CHAPTER 13	RESULTS & DISCUSSION	55 - 57
CHAPTER 14	FUTURE SCOPE	58
CHAPTER 15	CONCLUSION	59
CHAPTER 16	REFERENCES	60 - 63
	PAPER ACCEPTANCE MAILSCREENSHOT	

LIST OF FIGURES

Sr. No.	Figure Name	Page No.
1	1.1 Sample faces extracted from FF++ and DFDC datasets.	1
2	4.1 Architecture of Convolutional Neural Network	10
3	4.2 Architecture of Recurrent Neural Network	11
4	5.1 General Workflow of Machine Learning Algorithms	12
5	5.2 Classification of Machine Learning Algorithm	14
6	5.3 Multiple hyperplanes separate the data from two classes	15
7	5.4 Decision Tree Structure	16
8	5.5 Logistic Regressions	17
9	5.6 Naïve Bayes Classifier	19
10	6.1 Workflow Diagram	20
11	8.1 Dataset collection and Forwards it to preprocessing Step	26
12	8.2 Preprocessing of Datasets	27
13	8.3 Data Augmentation	29
14	8.4 Training and Prediction Phase	36
15	8.5 Threshold Phase	38

CHAPTER 1:
INTRODUCTION

CHAPTER 1: INTRODUCTION

This means that a speaker's identity can be changed with a moderate amount of effort. Digital face editing tools are now easy to use making them accessible to everyone regardless of art or picture retouching experience. Users can now start accessing artificial tools that effectively handle tasks by themselves. New artistic developments help people create better art with their technological tools. Advanced technology enables criminals to produce false videos with relative ease. Face-altering technology poses dangers because attackers can spread fake videos and create illegal revenge pornography. Establishing true identities in video sequences stands as today's major concern because spreading fake content creates serious problems for society.

Research around checking if filmmakers change their content has existed for a long time. Experts in multimedia forensics began studying this field long ago with their research about different solution methods. These authors examine film coding details to discover information about movie processing. Research institutions study copy-move detection modifications using dense data blocks. Many experts have created ways to spot when video frames repeat or get removed. All the above methods rely on the same principle: Each permanent change makes a unique detection mark to help find exactly where the editing took place. The traces forensic scientists look for tend to be hard to see and pick up. Hard-to-detect video edits occur during extreme down sampling or simultaneous complex edits plus strong compression steps. Realistic manipulation techniques create effective obstacles for forensic modelling systems.

Current facial transformation techniques prove difficult for forensic experts to identify accurately in modern times. Several different techniques modify face images with no single explanation working for all cases Their technology operates on limited areas within video frames-usually just the face or parts of it. Reference taken from and the Facebook DFDC dataset declare on Kaggle in December 2019 we study how different manipulation tools like deepfakes, Face2Face, Footage Swap and Neural Textures can be identified. We create a new variant of EfficientNetB4 through our work by adding attention elements from. Researchers find it harder to detect manipulated films because these videos spread on social media platforms apply data compression and coding.

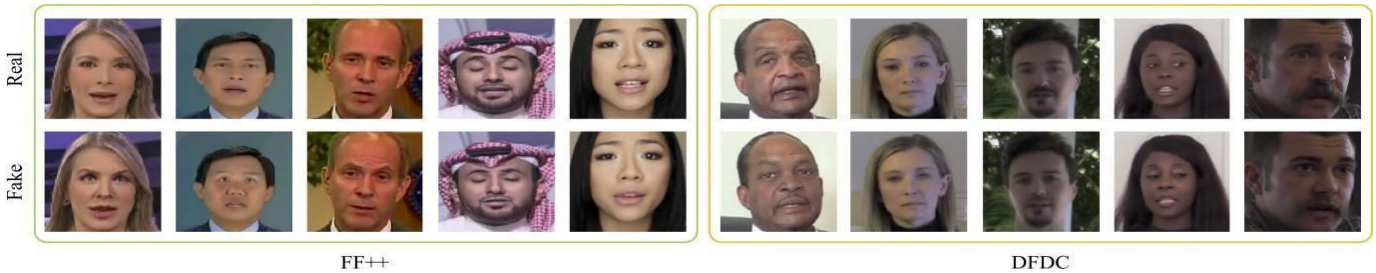


Fig.1.1 Samples Faces extracted by FF++ and DFDC datasets.

1.1 Problem statement

The rise of deepfake technology, powered by advancements in artificial intelligence, has led to an increase in manipulated media that is difficult to distinguish from real content. Deepfake videos, images, and audio clips are being used in misinformation campaigns, identity fraud, and malicious activities, posing a significant threat to individuals, organizations, and even national security.

Traditional detection methods struggle to keep up with the rapid evolution of deepfake algorithms, making it increasingly difficult for media consumers and security agencies to identify fake content in real time. Existing detection solutions often lack scalability, speed, and adaptability to new deepfake generation techniques. Additionally, there is a gap in providing actionable real-time insights that can help users make informed decisions regarding the authenticity of digital media.

To address this growing challenge, an AI-powered deepfake detection system with real-time insights is required. This system must be capable of efficiently analysing media content, identifying deepfake patterns, and providing users with reliable results within seconds. The need for an intelligent, fast, and transparent detection mechanism is critical to ensuring trust in digital media and preventing the spread of misinformation.

1.2 Need for Automation: Deepfake media is evolving at an alarming rate, making it increasingly difficult for human reviewers and conventional detection systems to keep up. Manual detection methods are time-consuming, error-prone, and ineffective against rapidly generated deepfake content. Automated AI-powered detection is essential for the following reasons:

- **Speed & Efficiency** – Real-time detection ensures timely identification of deepfakes before they spread.
- **Scalability** – Automated systems can analyse large volumes of media content across multiple platforms
- **Adaptability** – AI models can continuously learn and adapt to new deepfake generation techniques.
- **Accuracy & Reliability** – AI-driven detection reduces human bias and improves precision in identifying manipulations.
- **Security & Misinformation Control** – Prevents the spread of malicious deepfake content in sensitive domains such as politics, finance, and personal security.

1.3 Objectives: The primary objective of this research is to develop a robust AI-powered deepfake detection system that provides real-time insights for combating synthetic media threats. The specific objectives include:

- **Develop a Deep Learning Model** – Train a neural network to detect deepfake artifacts such as facial inconsistencies, unnatural expressions, and mismatched lip synchronization.
- **Enhance Real-Time Processing** – Optimize the model to ensure fast and efficient deepfake detection suitable for real-world applications.
- **Improve Accuracy & Robustness** – Utilize advanced AI techniques to minimize false positives and false negatives in detection.
- **Adapt to Evolving Deepfake Techniques** – Implement continuous learning mechanisms to keep up with the latest deepfake generation methods.
- **Provide Actionable Insights** – Develop a system that not only detects deepfakes but also offers interpretability and confidence scores for better decision-making.

- **Integrate into Various Platforms** – Design the solution to be adaptable for integration with social media, law enforcement, and cybersecurity frameworks.

1.4 PROJECT PURPOSE

1. Enhancing Deepfake Detection Accuracy: The primary purpose of this project is to develop an AI-driven system that improves the accuracy of deepfake detection. By leveraging deep learning models, computer vision, and audio analysis techniques, the system aims to detect subtle inconsistencies in deepfake media. It will utilize neural networks trained on large datasets of real and fake media to achieve high precision in distinguishing authentic content from manipulated ones.

2. Real-Time Analysis and Detection: One of the key purposes of this project is to enable real-time detection of deepfake content. Many existing detection tools require significant processing time, making them impractical for live media verification. By optimizing AI models and leveraging advanced processing techniques, the system will be capable of providing instant results, ensuring that users can assess the credibility of content within seconds.

3. Providing Actionable Insights: Beyond just identifying deepfake content, this project aims to provide users with actionable insights. The system will generate detailed reports highlighting key indicators of manipulation, such as facial inconsistencies, unnatural lip-syncing, or audio distortions. These insights will help individuals, media organizations, and cybersecurity teams make informed decisions regarding content authenticity.

4. Combating Misinformation and Digital Fraud: The project seeks to contribute to the fight against misinformation and digital fraud. Deepfakes are increasingly used to spread false narratives, impersonate individuals, and manipulate public opinion. By deploying an AI-powered detection system, this project aims to mitigate the risks associated with manipulated media, ensuring a more secure and trustworthy digital environment.

5. Scalability and Adaptability to Emerging Deepfake Techniques: Deepfake technology is constantly evolving, with new methods emerging to bypass existing detection techniques. The purpose of this project is to build a scalable and adaptable AI model that can be continuously updated with new datasets and detection algorithms. By implementing self-learning mechanisms and model retraining, the system will remain effective against advanced deepfake generation techniques.

6. Ensuring Transparency and Explainability: Many AI-based deepfake detection systems function as "black boxes," providing results without explaining the reasoning behind their conclusions. This project aims to enhance transparency by incorporating explainable AI (XAI) techniques. Users will be able to understand why a piece of media was classified as a deepfake, increasing trust in the system and improving its adoption in critical sectors such as journalism, cybersecurity, and law enforcement.

7. Enhancing Cybersecurity and Protecting Identities: With the rise of AI-generated impersonation attacks, this project will play a crucial role in cybersecurity. Many individuals, celebrities, and politicians have fallen victim to deepfake-based identity theft and fraudulent activities. By implementing a robust detection system, this project will help protect individuals from digital identity manipulation and cyber threats.

8. Supporting Organizations and Government Agencies: This project aims to serve not just individuals but also organizations and government agencies that need reliable deepfake detection tools. Media companies, social media platforms, and law enforcement agencies can integrate this system into their workflows to prevent the spread of misleading content and ensure public safety.

9. Raising Awareness and Educating the Public: A significant aspect of this project is to raise awareness about the dangers of deepfake media. By providing real-time insights and educational resources, the system will help users develop media literacy skills, enabling them to critically analyse digital content and avoid falling victim to misinformation.

CHAPTER 2:
LITERATURE REVIEW

CHAPTER 2: LITERATURE REVIEW

[1] M. Zoller, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Prez, M. Stamm Inger, M. Niner, and C. Theobald, "State of the art on monocular 3d face reconstruction, tracking, and applications," Computer Graphics Forum, vol. 37, pp. 523–550, 2018, This paper provides a comprehensive review of monocular 3D face reconstruction, tracking, and their applications. It discusses various state-of-the-art techniques, including geometric and learning-based methods, emphasizing their accuracy and efficiency. The authors highlight challenges such as occlusion, illumination changes, and real-time processing. The study also explores applications in virtual reality, animation, and identity verification.

[2] J. Thies, M. Zoll Hofer, M. Stamm Inger, C. Theobald, and M. Neuner, "Face2face: Real-time face capture and e-enactment of grub videos," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2387–2395, In this paper, the authors introduce Face2Face, a real-time face capture and e-enactment system for manipulating facial expressions in videos. Their approach utilizes a monocular RGB camera and a dense 3D model to transfer facial expressions from one person to another in real-time.

[3] J. Thies, M. Solnhofen, and M. Neuner, "Deferred neural rendering: Image synthesis using neural textures," ACM Transactions on Graphics (TOG), vol. 38, no. 4, pp. 1–12, 2019, This paper presents Deferred Neural Rendering, a novel method for image synthesis using neural textures. The authors propose a hybrid approach combining traditional rendering pipelines with deep neural networks to achieve high-fidelity image generation. The study demonstrates its effectiveness in real-time rendering, achieving photorealistic results. It has applications in gaming, virtual reality, and content creation.

[4] "Deepfakes GitHub, https://github.com/deepfakes/faceswap, In this paper, open-source project provides tools for face swapping using deep learning techniques. It utilizes generative adversarial networks (GANs) and autoencoders to create highly realistic deepfake videos. The repository serves as a research platform for advancements in deepfake technology and detection. The project has raised concerns about ethical implications and misinformation.

[5] "Faceswap" https://github.com/MarekKowalski/FaceSwap/, In this method, Similar to Deepfakes GitHub, this repository provides a deep learning-based face-swapping framework. It allows researchers and developers to experiment with face manipulation techniques. The tool is widely used for studying deepfake generation and forensic countermeasures. Its open-source nature contributes to both the advancement and detection of deepfake technologies.

[6] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, In this paper, the authors address the growing threat of deepfake videos targeting world leaders and propose countermeasures. They present a deep learning-based detection framework that analyses facial arti facts and inconsistencies to identify synthetic videos. Their research highlights the potential dangers of deepfakes in political misinformation and digital security. The study emphasizes the need for robust forensic tools to combat malicious deepfake content.

[7] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, “Vision of the unseen: Current trends and challenges in digital image and video forensics,” ACM Computing Surveys, vol. 43, no. 26, pp. 1–42, 2011, This paper reviews trends and challenges in digital image and video forensics. It categorizes forensic methods into source identification, content integrity analysis, and counterfeit detection.

[8] S. Milani, M. Fontana, P. Betaine, M. Barni, A. Piva, M. Pagliacci, and S. Tufaro, “An overview on video forensics,” APSIPA Transactions on Signal and Information Processing, vol. 1, p. E2, 2012, In this method, the authors present an overview of video forensics techniques used to detect tampering and ensure content authenticity. They discuss various approaches, including compression analysis, noise pattern detection, and metadata verification. The study highlights the growing need for automated forensic tools in the digital era. The findings are relevant to forensic analysts and cybersecurity experts.

[9] M. C. Stamm, Min Wu, and K. J. R. Liu, “Information forensics: An overview of the first decade,” IEEE Access, vol. 1, pp. 167–200, 2013, This paper provides a decade-long review of information forensics, focusing on multimedia security. The authors discuss forensic methods for detecting digital tampering, watermarking techniques, and anti-forensic strategies. Their study highlights the ongoing challenges in ensuring data integrity. The paper serves as a foundational reference for researchers in digital forensics and cybersecurity.

10] P. Betaine, S. Milani, M. Tallahatchie, and S. Tufaro, “Codec and Gop identification in double compressed videos,” IEEE Transactions on Image Processing (TIP), vol. 25, pp. 2298–2310, 2016, In this paper, the authors propose a method for identifying codec and Group of Pictures (GOP) structures in double-compressed videos. Their approach helps in detecting video forgery by analysing compression inconsistencies.

SUMMARY

This literature review, explores state-of-the-art techniques for deepfake image and video detection, drawing insights from various research studies. Zoller et al. (2018) provide a comprehensive review of monocular 3D face reconstruction and tracking, highlighting challenges like occlusion and illumination. Thies et al. (2016) introduce Face2Face, a real-time facial e-enactment system that demonstrates vulnerabilities in video manipulation. Deferred Neural Rendering (Thies et al., 2019) presents a hybrid rendering approach, advancing photorealistic image synthesis. Open-source projects like Deepfakes GitHub and Face Swap contribute to deepfake generation and forensic countermeasures. Agarwal et al. (2019) address deepfake threats against world leaders, proposing detection frameworks analysing facial artifacts. Rocha et al. (2011) and Milani et al. (2012) review digital forensics trends, emphasizing forensic techniques for media authentication. Stamm et al.

CHAPTER 3:
OBJECTIVE

CHAPTER 3: OBJECTIVE

1. **Develop an Efficient AI Model for Deepfake Detection:**

    - Design and implement a deep learning-based model capable of identifying deepfake content with high accuracy.
    - Utilize state-of-the-art architectures such as Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), or Generative Adversarial Networks (GANs) to enhance detection capabilities.
    - Fine-tune and optimize the model using benchmark datasets to improve precision and recall.

2. **Real-Time Processing and Analysis:**

    - Implement a system capable of detecting deepfake content in real-time with minimal latency.
    - Optimize model inference using techniques like quantization and pruning to ensure efficient deployment on edge devices or cloud platforms.
    - Develop a pipeline for continuous learning, where the model updates itself based on new deepfake patterns.

3. **Explainability and Interpretability of AI Decisions:**

    - Integrate explainable AI (XAI) techniques to provide users with insights into why a particular media file is classified as a deepfake.
    - Use heatmaps, attention maps, and feature visualization techniques to highlight manipulated regions in deepfake media.
    - Provide transparency in decision-making to enhance trust and adoption in real-world applications.

4. **Dataset Curation and Augmentation:**

    - Collect and preprocess large-scale datasets of real and synthetic media from various sources.
    - Utilize data augmentation techniques to improve model robustness against adversarial deepfake manipulations.
    - Ensure dataset diversity to enhance model generalization across different demographics, lighting conditions, and media formats.

5. **Integration with Streaming and Social Media Platforms:**

    - Develop APIs and software solutions to integrate deepfake detection capabilities with existing social media and video streaming platforms.
    - Create browser extensions or mobile applications for real-time verification of media authenticity.
    - Provide alerts and detailed reports on detected deepfake content to relevant authorities and users.

6. Robustness Against Adversarial Attacks:

- Evaluate the resilience of the deepfake detection model against adversarial attacks designed to bypass detection mechanisms.
- Implement adversarial training techniques to strengthen the model's ability to detect sophisticated deepfake techniques.
- Conduct extensive testing against novel and unseen deepfake manipulation techniques.

7. Ethical Considerations and Policy Recommendations:

- Address ethical implications related to deepfake detection, privacy concerns, and potential misuse of the technology.
- Propose guidelines and recommendations for regulatory bodies on the responsible use of AI-powered deepfake detection systems.
- Foster public awareness regarding deepfake threats and the importance of media verification.

8. User Interface and Experience Design:

- Develop an intuitive and user-friendly interface for non-technical users to easily verify the authenticity of media files.
- Incorporate interactive visualization features that allow users to explore detected deepfake artifacts.
- Ensure accessibility features to make the system inclusive for a wide range of users.

9. Scalability and Deployment:

- Design a scalable infrastructure that can handle high-volume media processing across various platforms.
- Utilize cloud-based and edge computing techniques to balance performance and resource utilization.
- Develop a deployment strategy for making the system available as an open-source tool or commercial product.

10. Continuous Improvement and Future Enhancements:

- Establish a framework for continuous improvement by incorporating feedback from real-world applications.
- Explore integration with blockchain for immutable authentication and verification of media authenticity.
- Investigate future deepfake generation techniques to stay ahead of emerging threats and enhance detection methods.

CHAPTER 4:
ARCHITECTURE

CHAPTER 4: ARCHITECTURE

4.1. ARCHITECTURE

1. Convolutional Neural Networks:

Convolutional Neural Networks (CNNs) are deep learning models that are highly effective for image-related tasks, including deepfake detection. They specialize in learning hierarchical features directly from raw pixel data, making them well-suited for image analysis and interpretation.

4.2 Key Components and Layers of CNNs:

Convolutional Layers:
Filters/Kernels: Small matrices (e.g., 3x3, 5x5) that slide over the input image to detect local patterns such as edges and textures. Each filter generates a feature map by applying convolution operations across the image.

Stride and Padding: Stride determines how much the filter moves at each step, and padding adds extra pixels around the border of the image to control the output size.

Activation Functions:
ReLU (Rectified Linear Unit): Introduces non-linearity by setting negative values to zero and keeping positive values unchanged, helping the network learn complex patterns.

Pooling Layers:

- **Max Pooling:** Reduces the spatial dimensions of feature maps by taking the maximum value in each sub-region (e.g., 2x2), which helps reduce computation and overfitting.
- **Average Pooling:** Similar to max pooling but takes the average value of each sub-region.

Fully Connected Layers (Dense Layers):

These layers connect every neuron in one layer to every neuron in the next, typically used towards the end of the network to perform high-level reasoning and classification based on the extracted features.

Dropout Layers:
A regularization technique that randomly sets a fraction of neurons to zero during training to prevent overfitting and ensure the model generalizes well.

For more complex and accurate fruit detection models, more advanced architectures like MobileNet V2, ResNet, or EfficientNet are often used.

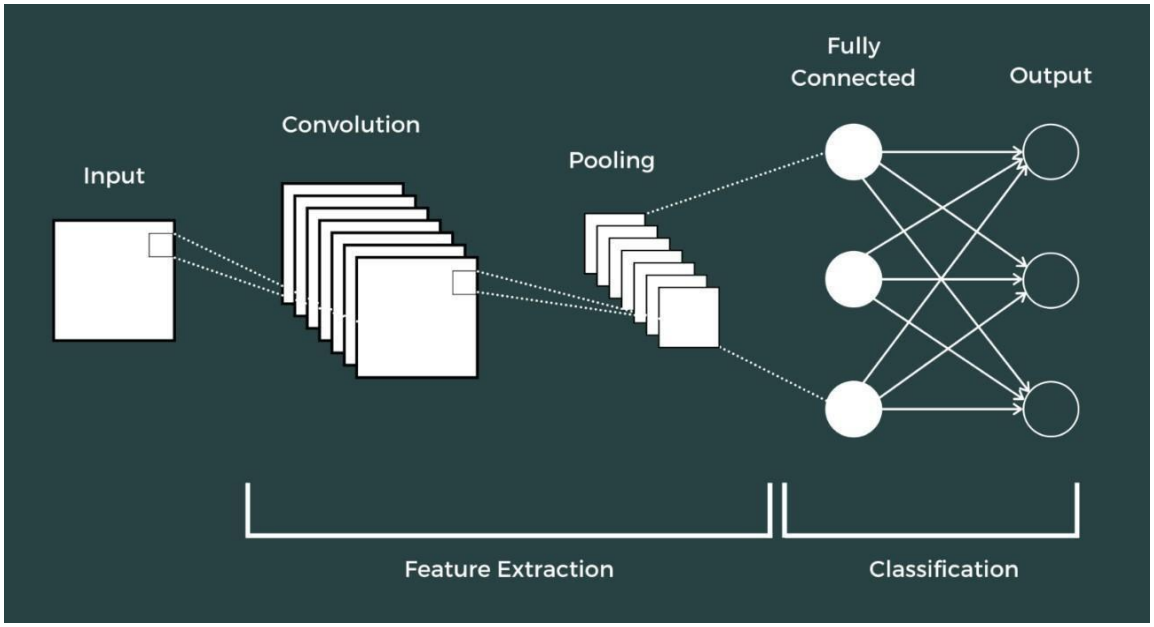


Fig. 4.1 Architecture of Convolutional Neural Network

2. Recurrent Neural Networks:

Recurrent Neural Networks (RNNs) are a class of deep learning models designed for sequential data processing, making them well-suited for tasks like speech recognition, natural language processing, and video analysis. Unlike traditional neural networks, RNNs have memory cells that retain information from previous inputs, enabling them to capture temporal dependencies. Variants like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) address issues like vanishing gradients, improving performance in deepfake video and audio detection.

4.3 Key Components and Layers of RNNs:

- **Convolutional Layer** – Extracts spatial features from input images using learnable filters (kernels) that detect edges, textures, and patterns.
- **Activation Function (ReLU)** – Introduces non-linearity by applying the Rectified Linear Unit (ReLU), which helps the network learn complex patterns efficiently.
- **Pooling Layer (Max/Average Pooling)** – Reduces spatial dimensions by selecting the most important features, improving computational efficiency and reducing overfitting.
- **Fully Connected (FC) Layer** – Flattens the extracted features and connects them to output neurons for final classification or regression tasks.
- **Dropout Layer** – Regularization technique that randomly drops connections during training to prevent overfitting and enhance model generalization.
- **Batch Normalization** – Normalizes inputs across mini-batches to stabilize learning, improve convergence speed, and mitigate internal covariate shifts.
- **Softmax/Sigmoid Output Layer** – Converts final feature maps into probabilities for classification tasks, with Softmax used for multi-class and Sigmoid for binary classification.

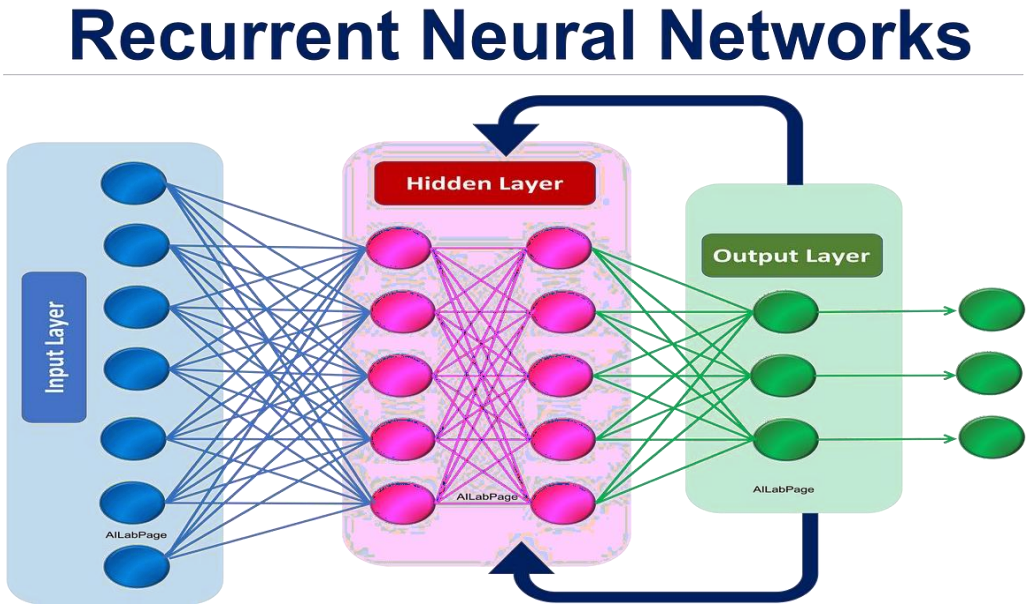


Fig. 4.2 Architecture of Recurrent Neural Networks

CHAPTER 5:
ALGORITHMS

CHAPTER 5: ALGORITHMS

5.1. Machine Learning Approaches

Deepfake image and video detection is a critical area of research, addressing the growing threat of AI-generated synthetic media used for misinformation, identity theft, and fraud. The detection techniques fall into three major categories: feature-based, machine learning-based, and deep learning-based methods. Computer systems may learn from experience and become more intelligent without explicit programming thanks to a technique called machine learning. When a computer algorithm is trained, it can solve related issues by using the relationship it learned during training. Training data is used to categorize online suggestions to individual users based on information gathered from previous purchases or searches.

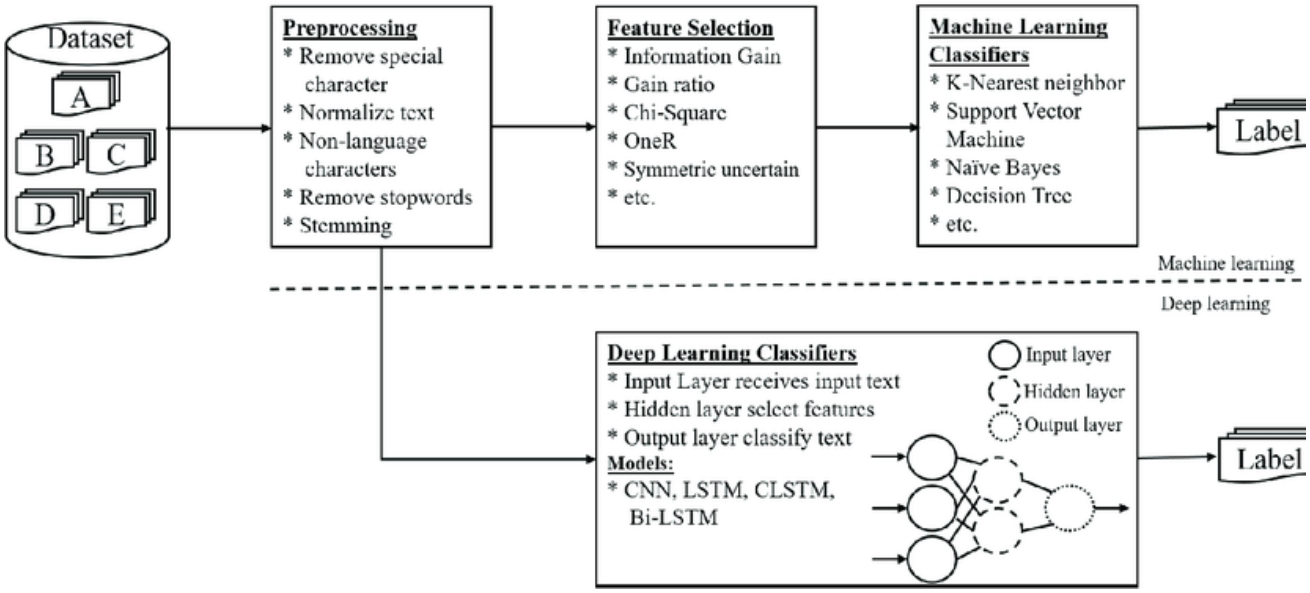


Fig.5.1 General Workflow of Machine Learning Algorithms

5.2. Types Of Machine Learning Algorithms

Supervised machine learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset. Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc. After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height, shape, colour, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Cat category. This is the process of how the machine identifies the objects in Supervised Learning.

5.2.1. Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabelled dataset, and the machine predicts the output without any supervision. In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision. The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences. Machines are instructed to find the hidden patterns from the input dataset. Let's take an example to understand it more preciously; suppose there is a basket of fruit images, and we input it into the machine learning model. The images are totally unknown to the model, and the task of the machine is to find the patterns and categories of the objects.

5.2.2. Semi-Supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabelled datasets during the training period. Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabelled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels. To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. The main aim of semi-supervised learning is to effectively use all the available data, rather than only labelled data like in supervised learning. Initially, similar data is clustered along with an 18 unsupervised learning algorithm, and further, it helps to label the unlabelled data into labelled data. It is because labelled data is a comparatively more expensive acquisition than unlabelled data. We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college. Further, if that student is selfanalysing the same concept without any help from the instructor, it comes under unsupervised learning. Under semi-supervised learning, the student has to revise himself after analysing the same concept under the guidance of an instructor at college.

5.2.3. Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards. In reinforcement learning, there is no labelled data like supervised learning, and agents learn from their experiences only. The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his daytoday life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards. Due to its way of working, reinforcement learning is employed in

different fields such as Game theory, Operation Research, Information theory, multi-agent systems A reinforcement learning problem can be formalized using Markov Decision Process (MDP). In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

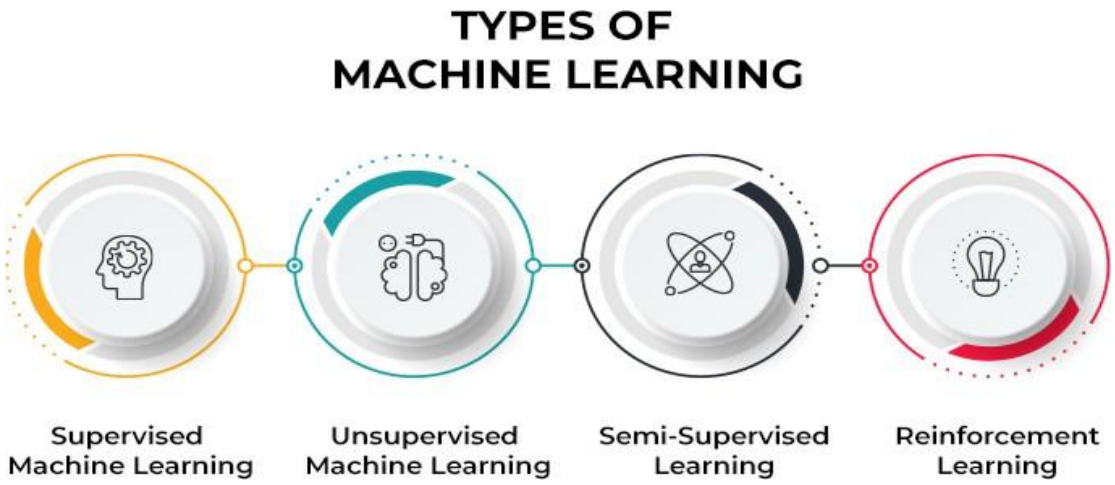


Fig.5.2 Classification of Machine Learning Algorithm

5.3. Support Vector Machines

SVM, also known as Support Vector Machine, is one of the most widely used Supervised Learning methods for both Regression and Classification problems. On the other hand, machine learning classification challenges are its main application. SVM assesses the input data and finds patterns for regression analysis and classification. The hyperplane that optimizes the margin between data point clusters pertaining to distinct classes is found by SVM in order to do classification.

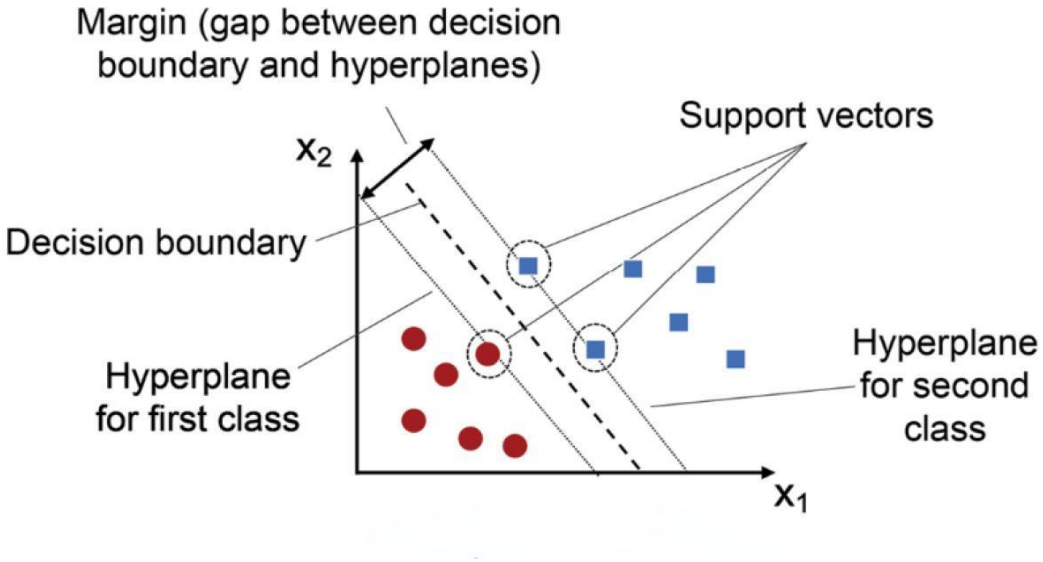


Fig .5.3 Multiple hyperplanes separate the data from two classes

A soft margin allows for some misclassifications or violations of the margin to improve generalization. The SVM optimizes the following equation to balance margin maximization and penalty minimization:

Objective Function = (1/margin) + λ∑penalty

The penalty used for violations is often **hinge loss**, which has the following behavior:

- If a data point is correctly classified and within the margin, there is no penalty (loss = 0).
- If a point is incorrectly classified or violates the margin, the hinge loss increases proportionally to the distance of the violation.

5.4. Decision Tree

Based on input data, decision trees—also referred to as regression trees and classification trees—predict output responses. The output response to that specific input data is obtained by following the decisions made by the tree's root node to the leaf node. As seen in Figure 17, the decision tree method uses a series of statistical tests to classify data. These tests divide the path of the tree by comparing the value that is input to a node with a threshold value. Multiple test outcomes are possible, as are various tree pathways that lead to the same output class label. The amount of branch splits in a tree determines its complexity. Depending on the complexity, these trees have minimal computational memory requirements, fast training and prediction speeds, and moderate predictive accuracy.

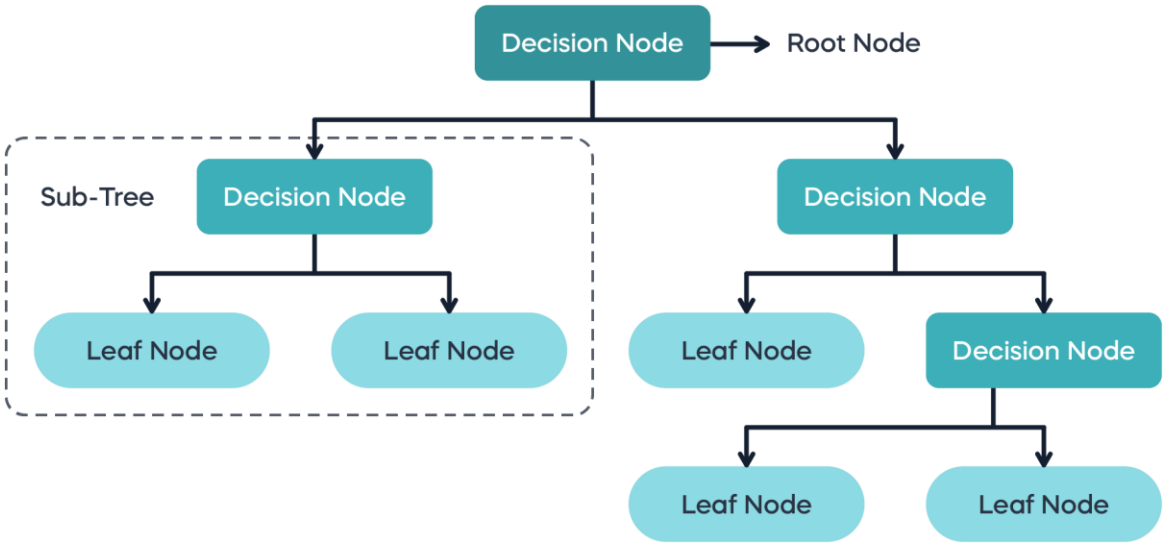


Fig.5.4 Decision Tree Structure

5.4.1. Decision Tree Terminologies

There are specialized terms associated with decision trees that denote various components and facets of the tree structure and decision-making procedure.

- **Root Node:** A decision tree's root node, which represents the original choice or feature from which the tree branches, is the highest node.
- **Internal Nodes (Decision Nodes):** Nodes in the tree whose choices are determined by the values of particular attributes. There are branches on these nodes that go to other nodes.
- **Leaf Nodes (Terminal Nodes):** The branches' termini, when choices or forecasts are decided upon. There are no more branches on leaf nodes.
- **Branches (Edges):** Links between nodes that show how decisions are made in response to particular circumstances.
- **Splitting:** The process of dividing a node into two or more sub-nodes based on a decision criterion. It involves selecting a feature and a threshold to create subsets of data.
- **Parent Node:** A node that is split into child nodes. The original node from which a split originates.
- **Child Node:** Nodes created as a result of a split from a parent node.
- **Decision Criterion:** The rule or condition used to determine how the data should be split at a decision node. It involves comparing feature values against a threshold.

- **Pruning:** The process of removing branches or nodes from a decision tree to improve its generalization and prevent overfitting.

5.5. Logistic Regression

Logistic regression is a statistical model used for binary classification. It predicts the probability that a given input belongs to one of two classes. Instead of fitting a straight line (as in linear regression), logistic regression applies the sigmoid function to map predictions to a probability range between 0 and 1.
Logistic regression is a fundamental machine learning model, but its application to deepfake detection is somewhat limited. While it can be used as a **baseline model**, deepfake detection often relies on more complex deep learning architectures. However, logistic regression can be useful in certain cases, such as:

- **Feature-Based Classification** – If deepfake detection models extract handcrafted features (e.g., pixel inconsistencies, frequency-domain artifacts in audio), logistic regression can be used for classification.
- **Lightweight Detection Systems** – For quick, low-resource inference, logistic regression can classify deepfakes based on predefined statistical features.
- **Preprocessing Step** – Before applying deep learning, logistic regression can serve as an initial filter for detecting anomalies.

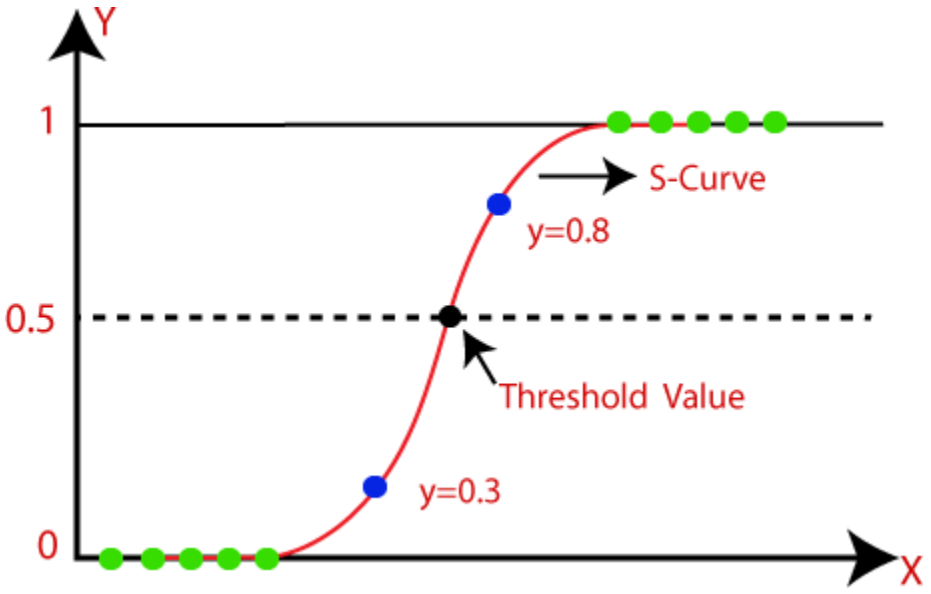


Fig.5.5 Logistic Regression

Logistic regression can be used as a **baseline model** for deepfake image and video detection by classifying extracted features as real or fake. Since logistic regression is a simple linear classifier, it requires meaningful handcrafted features rather than raw images or videos.
In **deepfake image detection**, features like pixel intensity inconsistencies, frequency domain artifacts, and metadata analysis (compression patterns and noise distribution) can be extracted. Logistic regression then assigns weights to these features and predicts whether an image is real or fake using the sigmoid function.

For **deepfake video detection**, time-series-based features such as **facial movement irregularities, unnatural eye blinks, lip synchronization errors, and frame-level inconsistencies** can be analyzed. By extracting statistical descriptors (e.g., average blink rate, head pose variations) and using logistic regression, a decision boundary is created to classify videos.

Although logistic regression is interpretable and computationally efficient, it struggles with high-dimensional raw data. Hence, it is often combined with **feature extraction techniques like PCA, Fourier transforms, or pre-trained CNN embeddings** to improve performance. For state-of-the-art deepfake detection, deep learning models such as **CNNs, LSTMs, and Transformers** are preferred due to their ability to learn complex patterns automatically.

5.6 Naive Bayes Classifier Algorithm

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

5.6.1. Bayes´ Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where,

- **P(A|B) is Posterior probability:** Probability of hypothesis A on the observed event B.
- **P(B|A) isLikelihood probability:** Probability of the evidence given that the probability of a hypothesis is true.
- **P(A) is Prior Probability:** Probability of hypothesis before observing the evidence.
- **P(B) is Marginal Probability:** Probability of Evidence.

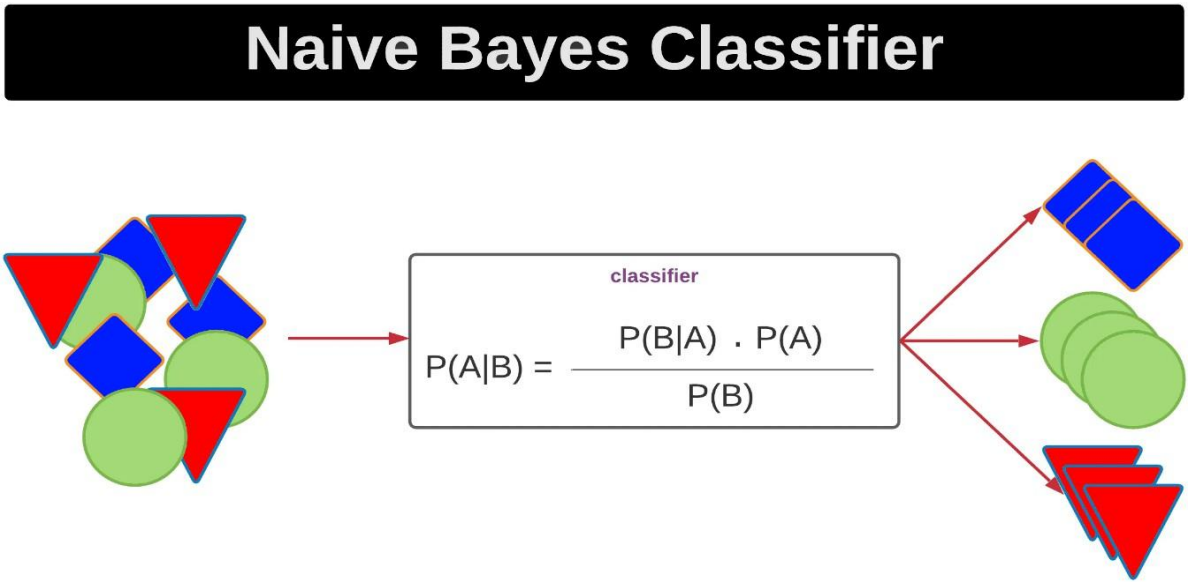


Fig. 5.6 Naïve Bayes Classifer

CHAPTER 6:

DATA FLOW DIAGRAM

CHAPTER 6: DATA FLOW DIAGRAM

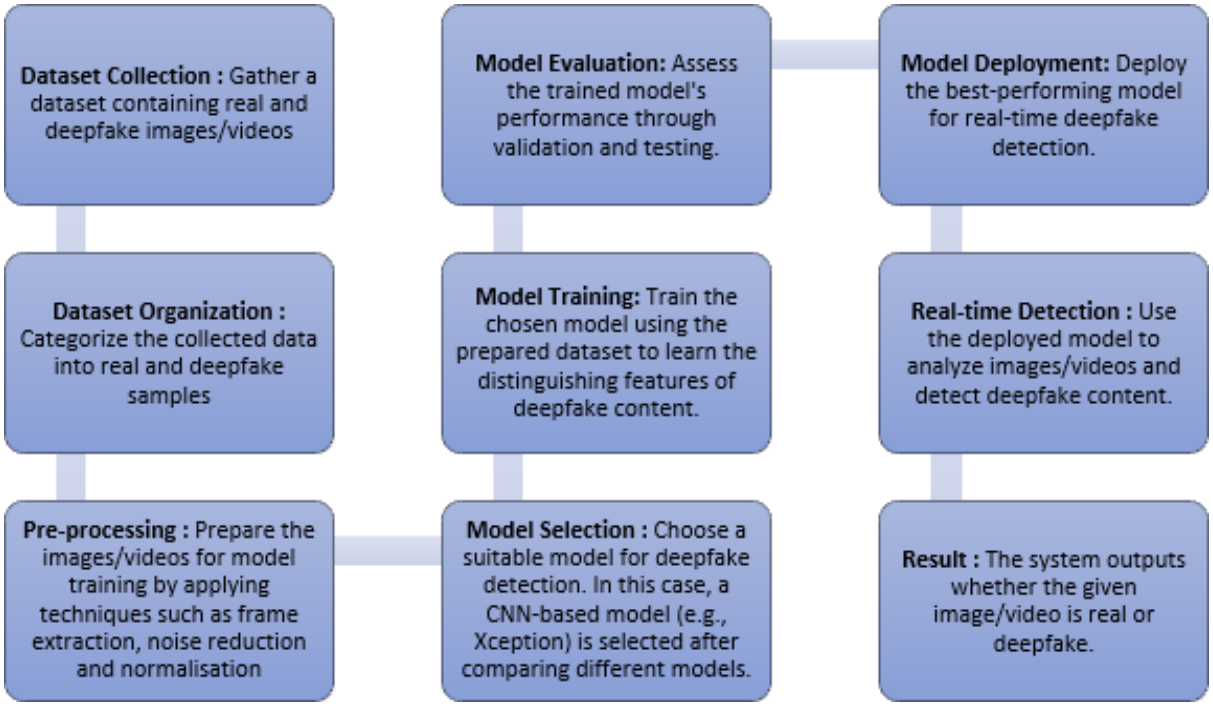


Fig.6.1 Workflow Diagram

A **Data Flow Diagram (DFD)** is a graphical representation that depicts the flow of data within a system. It illustrates how data is input, processed, and output in a system, highlighting the interactions between different components. DFDs use various symbols to represent processes, data stores, data flows, and external entities, providing a visual and intuitive way to understand the flow of information within a system. The DFD for this system can be outlined as follows:

1. **Deepfake Data Collection**: Gather a dataset of real and deepfake images/videos. The dataset should include different types of deepfake manipulations, lighting conditions, and resolutions. Ensuring diversity in the dataset is crucial for improving model robustness.

2. **Dataset Organization (Real vs. Deepfake Categories)**: Store images/videos in separate directories based on their labels (e.g., "Real" and "Deepfake"). This categorization aids in structured training and evaluation.

3. **Pre-processing**: **Frame Extraction (for videos)**: Convert deepfake videos into image frames for frame-wise analysis.
    - **Normalization**: Standardize pixel values to improve model convergence and performance.
    - **Feature Extraction**: Convert images into feature maps using different transformations such as frequency-domain analysis or attention-based mechanisms.

4. **Model Selection (Compare Different Models)**: Evaluate various deep learning models such as Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and LSTM-based models for sequential analysis. Performance is assessed using metrics like accuracy, precision, recall, and F1-score.

5. **Model Training (Using Xception or CNN-based Approach)**: Train a deep learning model using labeled deep fake and real images/videos. Data augmentation (flipping, cropping, noise addition) may be applied to improve generalization.

6. **Model Evaluation (Validation, Testing)**: Split the dataset into training and validation sets. Fine-tune the model using validation data and assess generalization on a separate test set.

7. **Model Deployment (Best Model Chosen)**: Deploy the trained model for real-time deepfake detection in applications such as social media monitoring, forensic analysis, and content authentication.

8. **Real-time Deepfake Detection (Processing Live Data)**: The deployed model analyzes incoming images/videos and identifies deepfake content in real-time. Pre-processing steps ensure compatibility with the model.

9. **Result**: The system outputs a classification result indicating whether the input image or video is real or deepfake, along with a confidence score.

CHAPTER 7:
SOFTWARE REQUIREMENTS

CHAPTER 7: SOFTWARE REQUIREMENTS

1. Programming Languages:

- **Python:** The primary language used for developing machine learning and deep learning models, due to its extensive library support and ease of use.

2. Development Environments:

- **Jupyter Notebook:** An interactive web application that allows for writing and running code, visualizing data, and interactively documenting the machine learning workflow.
- **Integrated Development Environment (IDE):** Tools such as PyCharm or Visual Studio Code offer robust environments for writing, debugging, and managing code efficiently.

3. Machine Learning and Deep Learning Libraries:

- **TensorFlow:** An open-source deep learning framework developed by Google, used for building and training neural networks.
- **PyTorch:** Another powerful open-source deep learning framework, known for its flexibility and dynamic computation graph.
- **Keras:** A high-level neural networks API, that runs on top of TensorFlow, simplifying the process of building and training models.

4. Data Processing Libraries:

- **NumPy:** A fundamental library for numerical computations, providing support for arrays and matrices, along with a collection of mathematical functions.
- **Pandas:** A library offering data structures and data analysis tools, essential for data manipulation and preprocessing.
- **OpenCV:** An open-source computer vision and image processing library, useful for tasks like reading images, image transformations, and edge detection.

5. Visualization Libraries:

- **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualizations in Python.
- **Seaborn:** A data visualization library based on Matplotlib, providing a high-level interface for drawing attractive and informative statistical graphics.

6. Machine Learning Utilities:

- **Scikit-learn:** A machine learning library that includes simple and efficient tools for data mining, data analysis, model selection, and evaluation metrics.

7. Web Frameworks:

- **Flask:** A lightweight web framework for Python, ideal for building web applications and APIs to deploy the machine learning model.
- **Django:** A more extensive web framework for Python, providing more features for large-scale applications but also suitable for machine learning model deployment.

8. User Interface Tools:

- **Streamlit:** An open-source app framework that allows for the quick creation and sharing of beautiful, custom web apps for machine learning and data science projects.

9. Version Control:

- **Git:** A version control system for tracking changes in the source code during the software development process, essential for collaboration and maintaining code history.

10. Package Management:

- **pip:** The package installer for Python, used to install and manage additional libraries and dependencies.
- **Anaconda:** A distribution of Python and R for scientific computing, providing package management and deployment, often used for managing data science environments.

11. Operating Systems:

- **Windows, macOS, or Linux:** The project can be developed on any of these operating systems. Linux is often preferred for its compatibility with various development and deployment tools.

12. Cloud Services (optional):

- **Google Collab:** A free cloud service for machine learning research, offering GPU acceleration to speed up training processes.

HARDWARE REQUIREMENTS

1. High-Performance CPU:

- powerful CPU is essential for managing various computing tasks, such as data loading and preprocessing. Multi-core CPUs are particularly advantageous as they can handle multiple processes simultaneously, increasing efficiency.

2. GPU (Graphics Processing Unit):

- GPUs are critical for training deep learning models due to their ability to perform parallel processing. This significantly speeds up the training of neural networks by handling numerous computations simultaneously.

3. RAM (Memory):

- Sufficient RAM is necessary for handling large datasets and the training process of machine learning models. More RAM allows for smoother and faster data processing.

4. Storage:

- Fast and ample storage is crucial for storing datasets, trained models, and intermediate data. SSDs are preferred over HDDs due to their faster read/write speeds and reliability.

5. **High-Resolution Monitor:**
   - High-resolution monitor is useful for visualizing data, monitoring training progress, and debugging. It helps in better-analyzing outputs and making necessary adjustments.

6. **Cooling System:**
   - Robust cooling system is essential to prevent overheating, especially during prolonged training sessions with high-performance GPUs and CPUs. Overheating can lead to hardware damage and performance throttling.

7. **Power Supply:**
   - Reliable PSU is necessary to ensure stable power delivery to all components. Highperformance GPUs and CPUs require more power, so the PSU must be capable of meeting these demands.

8. **Networking Equipment:**
   - High-speed internet is essential for downloading datasets, accessing cloud services, and collaborating with team members. Stable and fast internet connectivity ensures smooth and efficient workflow.
     (1) **Ethernet connection:** Provides the most reliable and fastest internet connection.
     (2) **High-speed Wi-Fi:** Suitable for flexibility and convenience.

9. **Backup Solution:**
   - A robust backup solution is important to prevent data loss. Regular backups ensure that data can be recovered in case of hardware failure or other issues.
   - **External SSDs:** Provide fast and reliable backup storage.
   - **Cloud storage:** Services like Google Drive, Dropbox, or AWS S3 offer scalable and accessible backup solutions.

10. **Cloud Computing Resources (Optional):**
    - For projects requiring more computational power than available locally, cloud computing resources can be used. These services offer scalable solutions for training and deploying models, with access to powerful hardware without the need for physical setup.
    - **Google Colab:** A free cloud service that offers GPU acceleration, allowing you to run Jupyter Notebooks in the cloud.

CHAPTER 8:
IMPLEMENTATION

CHAPTER 8: IMPLEMENTATION

8.1. Dataset Collection:

Dataset collection for deepfake image and video detection involves gathering real and manipulated media from publicly available datasets to train and evaluate detection models. Popular datasets include **DeepFake Detection Challenge (DFDC)**, which provides a large-scale collection of deepfake videos created using various generation techniques, and **FaceForensics++**, which contains real and tampered videos generated using DeepFake, FaceSwap, and NeuralTextures methods. **Celeb-DF** offers high-quality deepfake videos of celebrities, helping to improve model generalization. Other datasets like **DeeperForensics-1.0** and **Google/Jigsaw DeepFake Dataset** provide additional diversity in deepfake generation methods. For custom dataset creation, tools such as **DeepFaceLab** and **Faceswap** can be used to generate deepfake videos. Collected datasets should be preprocessed by extracting frames, resizing, normalizing pixel values, and applying face detection techniques to focus on relevant features. Proper dataset balancing between real and fake samples ensures unbiased model training, while augmentation techniques can help enhance robustness. By using a combination of these datasets, deepfake detection models can be trained effectively to identify manipulations in real-world scenarios.

Collected datasets should be preprocessed by extracting frames, resizing, normalizing pixel values, and applying face detection techniques to focus on relevant features. Proper dataset balancing between real and fake samples ensures unbiased model training, while augmentation techniques can help enhance robustness. By using a combination of these datasets, deepfake detection models can be trained effectively to identify manipulations in real-world scenarios.

Collecting a diverse and high-quality dataset is essential for training robust deepfake detection models. Several publicly available datasets provide deepfake videos and images generated using different AI techniques, such as **Generative Adversarial Networks (GANs)** and **Autoencoders**. Among the most widely used datasets, the **DeepFake Detection Challenge (DFDC)** dataset, developed by Facebook and AWS, offers thousands of real and manipulated videos with various deepfake generation methods. **FaceForensics++** is another commonly used dataset containing both real and fake videos manipulated using methods like **DeepFake, Face2Face, and FaceSwap**, making it useful for training deep learning models.

Another notable dataset is **Celeb-DF**, which consists of high-quality deepfake videos of celebrities with minimal visual artifacts, making it more challenging for detection models. The **DeeperForensics-1.0** dataset provides a large collection of face-manipulated videos created under different lighting conditions and compression levels to enhance model generalization. Additionally, the **Google/Jigsaw DeepFake Dataset** contains thousands of deepfake videos generated using various techniques, serving as an essential resource for training AI-based detection models.

Beyond public datasets, researchers can generate custom deepfake data using tools such as **DeepFaceLab, Faceswap, and First Order Motion Model**, which allow controlled manipulation of facial features in videos. Preprocessing collected datasets is crucial for effective model training. This includes **frame extraction from videos, face detection using OpenCV or Dlib, image**

resizing, and normalization to standardize input features. Balancing real and fake samples is essential to prevent model bias, and **data augmentation techniques** such as random cropping, flipping, and Gaussian noise addition can improve robustness.

For real-world deepfake detection, datasets should include diverse subjects, ethnicities, lighting conditions, and compression artifacts to enhance model adaptability. Many deepfake videos found online suffer from **compression loss and artifacts**, making it essential to include low-quality samples in the dataset to train models capable of handling real-world scenarios. By integrating multiple datasets and ensuring high-quality preprocessing, deepfake detection models can be trained effectively to identify even the most sophisticated AI-generated manipulations.

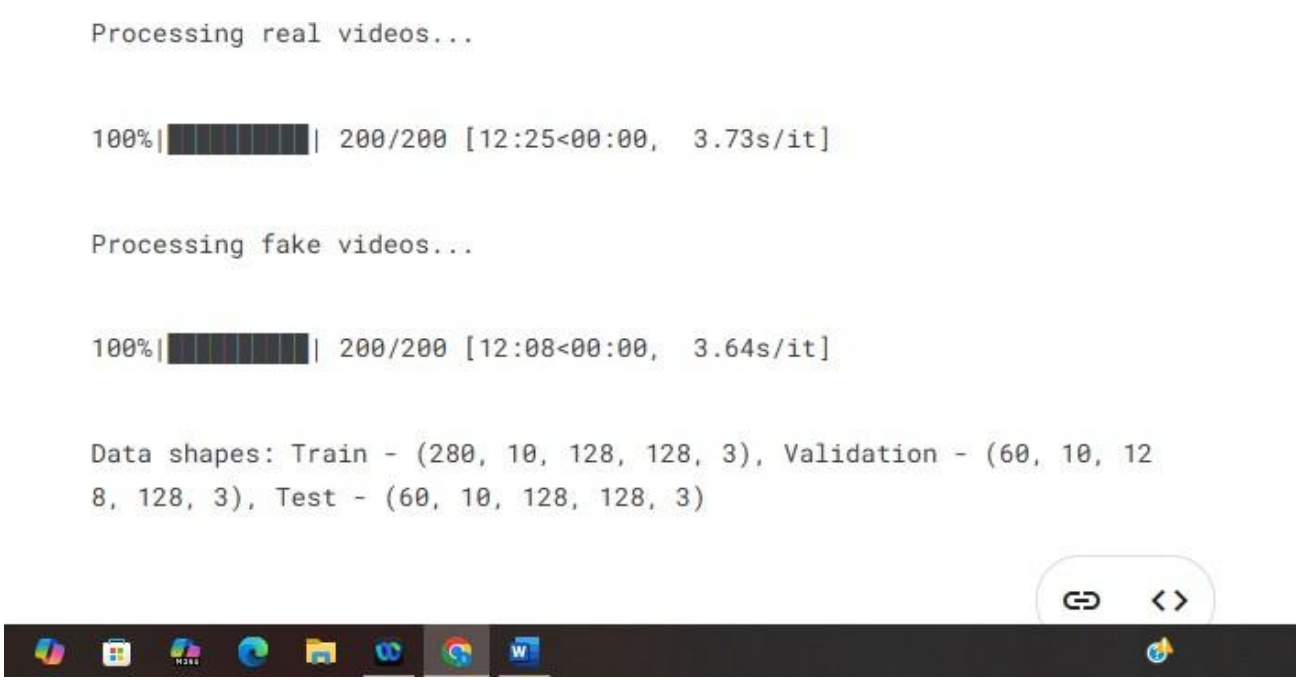


Fig. 8.1 Dataset collection and Forwards it to preprocessing step

Kaggle plays a significant role in deepfake image and video detection research by providing access to high-quality datasets, powerful computational resources, and a collaborative platform for data science and machine learning projects. Researchers can leverage Kaggle's datasets, such as the **DeepFake Detection Challenge (DFDC)** and **Celeb-DF**, to train and evaluate deepfake detection models. Kaggle also offers GPU and TPU support, allowing researchers to train deep learning models efficiently without requiring expensive hardware.

In addition to datasets, Kaggle provides pre-trained deepfake detection models and open-source notebooks that can accelerate research by offering baseline implementations and feature extraction techniques. The platform also fosters a community-driven approach, enabling researchers to collaborate, share insights, and improve detection methodologies through discussions and competitions. By using Kaggle, researchers can experiment with different machine learning architectures, apply transfer learning techniques, and benchmark their models against state-of-the-art deepfake detection methods. The availability of diverse datasets, powerful computing resources,

and a collaborative research environment makes Kaggle an essential tool for deepfake detection studies.

8.2. Pre-Processing

Preprocessing is a fundamental step in deepfake image and video detection as it enhances data quality and extracts meaningful features for model training. The process begins with **frame extraction**, where videos are converted into individual frames using tools like OpenCV or FFmpeg, allowing models to analyze deepfake manipulations at a granular level. Next, **face detection and alignment** are performed using algorithms such as MTCNN, Dlib, or FaceNetto focus on manipulated facial regions. Since deep learning models require consistent input sizes, **image resizing and normalization** are applied, typically scaling images to 224×224 or 299×299 pixels and normalizing pixel values between 0 and 1. To improve model robustness, **data augmentation** techniques such as flipping, rotation, brightness adjustment, and Gaussian noise addition are employed to prevent overfitting.

Additionally, **frequency domain analysis using Fourier Transform** helps reveal inconsistencies in the spatial patterns of deepfake images, as synthetic media often exhibit artifacts that are not visible in the time domain. Another important aspect is **compression artifact analysis**, where deepfake videos are examined for quantization noise and blockings, as compression often introduces detectable distortions. In cases where traditional machine learning methods like logistic regression or SVM are used, **feature extraction techniques** such as pixel intensity histograms, edge detection, and colour analysis help in differentiating real and fake images. By incorporating these preprocessing techniques, deepfake detection models become more effective at identifying AI-generated manipulations, improving overall accuracy and reliability in real-world applications.

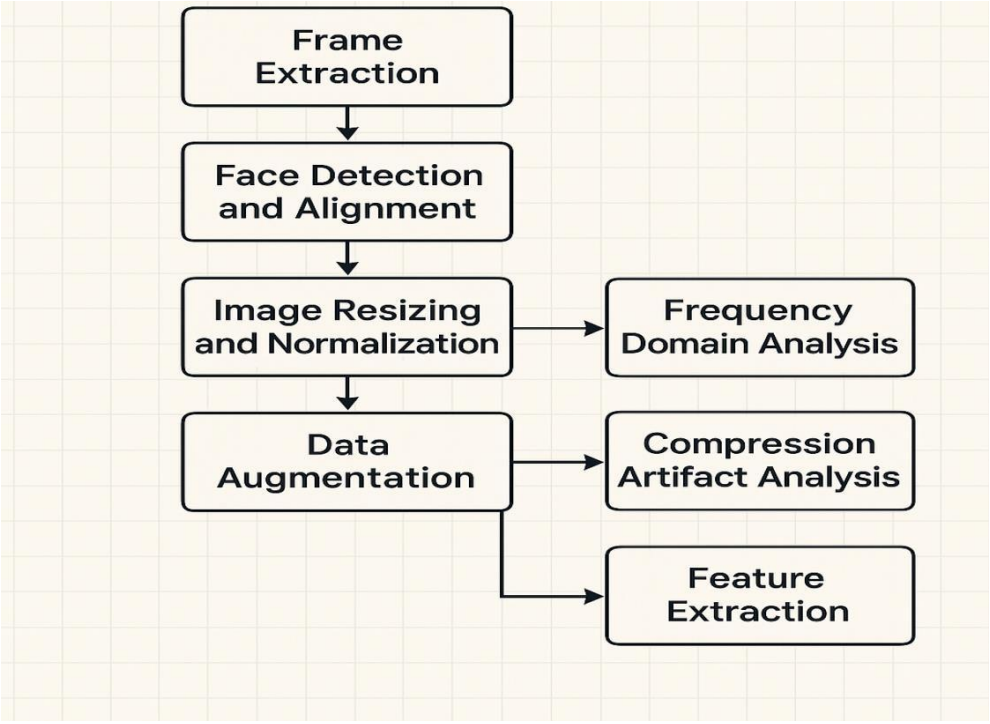


Fig.8.2 Preprocessing of Datasets

The preprocessing steps vary depending on whether the input is an image or a video. Below is a comprehensive breakdown of key preprocessing techniques:

1. Frame Extraction (For Video Input):

- Since deepfake detection in videos requires analyzing individual frames, frame extraction is the first step.
    - **Tools Used:** OpenCV
    - **Process:** Extract frames at a specific frame rate (e.g., 5 or 10 frames per second) to reduce redundant data and improve efficiency.

2. Face Detection and Alignment:

- Most deepfake manipulations target faces, so detecting and aligning faces ensures models focus on manipulated regions.
    - **Tools Used:**OpenCV, FaceNet
    - **Process:**
    - Detect faces in images using Haar cascades or deep learning-based methods.
    - Align faces by detecting key landmarks (eyes, nose, mouth) and rotating the image to normalize orientation.

3. Image Resizing and Normalization:

- Deep learning models require consistent input sizes and normalized pixel values for stable training.
- **Resizing:**
    - Images are resized to fixed dimensions like **224×224 (ResNet, VGG), 299×299 (Xception), or 512×512 (custom models)**.
- **Normalization:**
    - Pixel values are scaled to a **0-1** or **-1 to 1** range for improved numerical stability.
    - Example: **Divide by 255 for 0-1 normalization**.

4. Data Augmentation:

- To improve model robustness and prevent overfitting, augmented versions of images are generated.
- **Common Augmentation Techniques:**
    - **Flipping (horizontal, vertical)** – Helps generalize models against mirrored versions of faces.
    - **Rotation & Scaling** – Adjusts images by ±10–20° to introduce variance.
    - **Brightness & Contrast Adjustments** – Helps model detect deepfakes in different lighting conditions.
    - **Gaussian Noise & Blurring** – Simulates low-quality deepfake videos.

```
# Augment training data
augmented_data = []
augmented_labels = []

for i in range(len(X_train)):
    augmented_frames = augment_frames(X_train[i])
    augmented_data.append(augmented_frames)
    augmented_labels.append(y_train[i])

# Combine original and augmented data
X_train_augmented = np.concatenate((X_train, np.array(augmented_data)))
y_train_augmented = np.concatenate((y_train, np.array(augmented_labels)))

print(f"Augmented Train Data: {X_train_augmented.shape}")
```

```
Augmented Train Data: (560, 10, 128, 128, 3)
```

Fig. 8.3 Data Augmentation

5. Frequency Domain Analysis

- Many deepfake images contain unnatural patterns detectable in the frequency domain.
    - **Tools Used:** Fourier Transform (FFT), Wavelet Transform
    - **Process:** Convert images from spatial to frequency domain and analyze high-frequency components for anomalies.

6. Compression Artifact Analysis

- Since many deepfake videos are shared on social media, compression introduces detectable artifacts.
- **Process:**
    - Analyze quantization noise, blockiness, and loss of detail due to video compression.
    - Extract bitrate and codec information using FFmpeg.

7. Feature Extraction for Machine Learning Models

- For classical machine learning models like SVM or logistic regression, handcrafted features are extracted.
- **Common Features:**
    - **Edge Detection (Canny, Sobel filters)** – Helps analyze unnatural edges in deepfakes.
    - **Color Histograms & Texture Analysis** – Detects unnatural skin tone distributions.
    - **Blink Rate & Lip Sync Features** – Helps detect inconsistencies in videos.

8.3. Model Building

Building an effective deepfake detection model involves selecting the right architecture, preprocessing data, training the model, and optimizing its performance. The process begins with **data preparation**, where deepfake datasets such as DFDC, FaceForensics++, or Celeb-DF are preprocessed through frame extraction, face detection, and image normalization. Once the data is cleaned and augmented, a suitable model architecture is chosen based on the complexity of the deepfake manipulation.

Deep learning-based models, particularly **Convolutional Neural Networks (CNNs)** and **Transformer-based architectures**, are widely used for image-based deepfake detection. Popular CNN architectures like **Xception, EfficientNet, and ResNet** have shown strong performance in identifying deepfake artifacts by learning spatial features in images. These models extract pixel-level patterns such as unnatural edges, blending artifacts, and color inconsistencies. For video-based deepfake detection, **Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks** are incorporated to capture temporal inconsistencies, such as unnatural eye blinking or lip-sync mismatches. Additionally, **Vision Transformers (ViTs)** and hybrid models that combine CNNs with Transformers have gained attention for their ability to analyze both local and global patterns in manipulated media.

During the **training phase**, models are trained using abelled datasets where real and fake images or videos are assigned respective class labels. Techniques such as **transfer learning** are applied to fine-tune pre-trained models like XceptionNet on deepfake datasets, reducing the need for extensive computational resources. The loss function, typically **Binary Cross-Entropy** for binary classification or **Categorical Cross-Entropy** for multi-class detection, guides the model's learning process. Optimization algorithms like **Adam, SGD, or RMSprop** are used to adjust model weights and improve accuracy. **Data balancing techniques**, such as class weighting or oversampling, are employed to prevent bias toward real or fake classes, ensuring a more reliable detector.
Once the model is trained, it undergoes rigorous **evaluation and testing** using performance metrics like **accuracy, precision, recall, F1-score, and AUC-ROC** to measure its effectiveness in distinguishing real from fake content. Cross-validation is performed to check for overfitting, and models are fine-tuned by adjusting hyperparameters such as learning rate, batch size, and number of layers. In real-world applications, the trained model is deployed for inference, often integrated into social media platforms, forensic tools, or video authentication systems. To enhance real-time performance, **model compression techniques like quantization and pruning** are applied to reduce computational overhead.

By combining **robust data preprocessing, powerful deep learning architectures, and optimized training techniques**, an effective deepfake detection model can be developed to counter increasingly sophisticated AI-generated forgeries. As deepfake technology evolves, ongoing research in adversarial learning, explainable AI, and real-time detection will continue to improve model reliability and robustness.

Some CNNs Models are

1. EfficientNet-B4:

EfficientNet-B4 is a high-performance convolutional neural network (CNN) designed for image classification and feature extraction, making it an effective model for deepfake detection. It is part of the EfficientNet family, which optimizes accuracy and efficiency through compound scaling, balancing network depth, width, and resolution. Compared to traditional CNN architectures like ResNet and VGG, EfficientNet-B4 achieves better accuracy with fewer parameters, reducing computational cost while maintaining strong generalization capabilities.

EfficientNet-B4 can be implemented using **TensorFlow/Keras** or **PyTorch**, with pre-trained weights from ImageNet to speed up training. Below is a basic implementation using TensorFlow and Keras:

Step 1: Install Required Libraries

```
!pip install tensorflowefficientnet
```

Step 2: Load Pre-Trained EfficientNet-B4

```
import tensorflow as tf
fromtensorflow.keras.applications import EfficientNetB4
fromtensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
fromtensorflow.keras.models import Model

# Load EfficientNetB4 with pre-trained weights
base_model = EfficientNetB4(weights='imagenet', include_top=False, input_shape=(380, 380, 3))

# Freeze base model layers
base_model.trainable = False

# Add custom classification layers
x = GlobalAveragePooling2D()(base_model.output)
x = Dropout(0.4)(x)  # Prevent overfitting
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(1, activation='sigmoid')  # Binary classification (Real vs Fake)

# Create model
model = Model(inputs=base_model.input, outputs=output)

# Compile model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Step 3: Training the Model

python
CopyEdit

```
# Train the model on deepfake dataset
model.fit(train_data, validation_data=val_data, epochs=10, batch_size=16)
```

4. Performance and Advantages

- **High Accuracy:** EfficientNet-B4 achieves superior performance compared to standard CNNs.
- **Lower Computational Cost:** Requires fewer FLOPs compared to ResNet-50 or Inception-V4.
- **Scalability:** Can be extended to **EfficientNet-B5, B6, or B7** for improved accuracy at the cost of more computation.
- **Robustness:** Performs well across various deepfake datasets, including FaceForensics++, Celeb-DF, and DFDC.

1. EfficientNetB4ST:

EfficientNet-B4ST (Self-Transferring EfficientNet-B4) is an advanced variation of the **EfficientNet-B4** model designed for **better generalization and transfer learning** in specialized tasks like deepfake detection. This model integrates **self-training (ST)** techniques, where the network iteratively learns from its own predictions on unlabeled data, improving its robustness against new, unseen deepfakemanipulations.Below is an implementation of EfficientNet-B4ST using **pseudo-labeling** for self-training.

Step 1: Load Pre-Trained EfficientNet-B4 and Modify for Self-Training

```
import tensorflow as tf
fromtensorflow.keras.applications import EfficientNetB4
fromtensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
fromtensorflow.keras.models import Model

# Load EfficientNet-B4 with pre-trained weights
base_model = EfficientNetB4(weights='imagenet', include_top=False, input_shape=(380, 380, 3))

# Freeze base model layers
base_model.trainable = False

# Add classification layers
x = GlobalAveragePooling2D()(base_model.output)
x = Dropout(0.4)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(1, activation='sigmoid')  # Binary classification (Real vs Fake)

# Create model
model = Model(inputs=base_model.input, outputs=output)

# Compile model with Adam optimizer
```

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Step 2: Self-Training with Pseudo-Labeling

```
python
CopyEdit
import numpy as np

defpseudo_labeling(model, unlabeled_data, threshold=0.9):
pseudo_labels = []
   images = []

   for img, _ in unlabeled_data:
pred = model.predict(np.expand_dims(img, axis=0))[0][0]  # Get prediction
      if pred > threshold or pred < (1 - threshold):  # High confidence only
pseudo_labels.append(1 if pred> 0.5 else 0)
images.append(img)

returnnp.array(images), np.array(pseudo_labels)

# Generate pseudo-labeled dataset
pseudo_images, pseudo_labels = pseudo_labeling(model, unlabeled_data)

# Fine-tune model with self-labeled data
model.fit(pseudo_images, pseudo_labels, epochs=5, batch_size=16)
```

3.**EfficientNetAutoAttB4:**

EfficientNetAutoAttB4 is an **enhanced variant of EfficientNet-B4** that incorporates **automated attention mechanisms** to improve deepfake detection. It leverages **self-attention** and **channel-wise attention modules** to focus on critical features in manipulated media, making it more effective in identifying subtle deepfake artifacts such as **unnatural facial blending, inconsistent lighting, and temporal discrepancies in videos**.Implementation of EfficientNetAutoAttB4 for Deepfake Detection

Step 1: Define the Attention Mechanism

```
import tensorflow as tf
fromtensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout, Conv2D, Multiply, Reshape, Lambda
fromtensorflow.keras.applications import EfficientNetB4
fromtensorflow.keras.models import Model
importtensorflow.keras.backend as K

# Attention Module
defattention_module(inputs):
attn = GlobalAveragePooling2D()(inputs)
attn = Dense(inputs.shape[-1] // 16, activation='relu')(attn)
attn = Dense(inputs.shape[-1], activation='sigmoid')(attn)
attn = Reshape((1, 1, inputs.shape[-1]))(attn)
```

```
return Multiply()([inputs, attn])

# Load EfficientNet-B4 base model
base_model = EfficientNetB4(weights='imagenet', include_top=False, input_shape=(380, 380, 3))

# Apply attention module to the last convolutional layer
x = attention_module(base_model.output)

# Classification Layers
x = GlobalAveragePooling2D()(x)
x = Dropout(0.4)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(1, activation='sigmoid')  # Binary classification (Real vs Fake)

# Create model
model = Model(inputs=base_model.input, outputs=output)

# Compile model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

4. **EfficientNetAutoAttB4ST:**

EfficientNetAutoAttB4ST (Self-Training EfficientNet-B4 with Automated Attention) is an advanced deepfake detection model that integrates:

- EfficientNet-B4 for efficient feature extraction
- Self-attention mechanisms for focusing on manipulated regions
- Self-training (ST) techniques for learning from unlabeled data

This combination makes EfficientNetAutoAttB4ST highly effective at detecting deepfake images and videos, even in low-quality, compressed, and previously unseen datasets.Implementation of EfficientNetAutoAttB4ST for Deepfake Detection

Step 1: Define Attention and Self-Training Modules

```
import tensorflow as tf
fromtensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout, Multiply, Reshape
fromtensorflow.keras.applications import EfficientNetB4
fromtensorflow.keras.models import Model
import numpy as np

# Attention Module
defattention_module(inputs):
attn = GlobalAveragePooling2D()(inputs)
attn = Dense(inputs.shape[-1] // 16, activation='relu')(attn)
attn = Dense(inputs.shape[-1], activation='sigmoid')(attn)
attn = Reshape((1, 1, inputs.shape[-1]))(attn)
return Multiply()([inputs, attn])
```

```
# Load EfficientNet-B4 base model
base_model = EfficientNetB4(weights='imagenet', include_top=False, input_shape=(380, 380, 3))

# Apply attention module to the last convolutional layer
x = attention_module(base_model.output)

# Classification Layers
x = GlobalAveragePooling2D()(x)
x = Dropout(0.4)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(1, activation='sigmoid')  # Binary classification (Real vs Fake)

# Create model
model = Model(inputs=base_model.input, outputs=output)

# Compile model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Step 2: Implement Pseudo-Labeling for Self-Training

```
defpseudo_labeling(model, unlabeled_data, threshold=0.9):
pseudo_images, pseudo_labels = [], []

    for img, _ in unlabeled_data:
pred = model.predict(np.expand_dims(img, axis=0))[0][0]  # Get prediction score
        if pred > threshold or pred < (1 - threshold):  # Select only high-confidence samples
pseudo_labels.append(1 if pred> 0.5 else 0)
pseudo_images.append(img)

returnnp.array(pseudo_images), np.array(pseudo_labels)

# Generate pseudo-labeled dataset
pseudo_images, pseudo_labels = pseudo_labeling(model, unlabeled_data)

# Fine-tune model with self-labeled data
model.fit(pseudo_images, pseudo_labels, epochs=5, batch_size=16)
```

8.4 Feature Extraction:

The implementation of feature extraction for deepfake image and video detection using **EfficientNetAutoAttB4ST** involves multiple steps to ensure robust and accurate detection. The first step is to install the necessary libraries, such as TensorFlow, Keras, and EfficientNet, to facilitate deep learning model development. EfficientNet-B4 is used as the **backbone feature extractor**, leveraging its **pre-trained weights from ImageNet** to extract deep visual features. This model is initialized without the fully connected layers to focus solely on learning spatial and structural patterns in the input images.

Next, a **self-attention mechanism** is applied to the extracted feature maps to highlight manipulated regions. The attention module uses **Global Average Pooling (GAP) followed by dense layers** to generate attention weights that enhance significant regions while suppressing irrelevant areas. These refined feature maps are then passed through **Global Average Pooling**, reducing the feature dimensionality while preserving essential information.
To improve model generalization, **self-training (pseudo-labeling)** is implemented, where the model assigns pseudo-labels to unlabeleddeepfake images with high confidence and retrains itself iteratively.

8.5 Feature Selection:

Feature selection in deepfake image and video detection is a crucial step that enhances the model's ability to differentiate between real and manipulated content by selecting only the most relevant features. In EfficientNetAutoAttB4ST, feature selection is achieved through a combination of self-attention mechanisms, squeeze-and-excitation (SE) layers, and global average pooling (GAP). The self-attention module focuses on manipulated regions by assigning higher importance to facial features that show inconsistencies, such as unnatural edges, asymmetrical expressions, or blending artifacts. The SE layers refine the feature maps by emphasizing significant spatial and channel-wise details while suppressing irrelevant noise, ensuring that the model captures critical deepfake artifacts.

8.6. Training a classifier model

Training a classifier model for deepfake image and video detection involves learning patterns that distinguish real content from manipulated ones. In this case, logistic regression is used as the classification algorithm due to its efficiency and interpretability in binary classification tasks (real vs. fake). The process begins with extracting features from the images or video frames using EfficientNetAutoAttB4ST, which captures both low-level and high-level patterns related to deepfake artifacts. These extracted features serve as input for the logistic regression model, which applies a sigmoid activation function to estimate the probability of an image being real or fake. The dataset is divided into training and testing sets, ensuring proper evaluation of the model's performance. Before training, feature normalization is applied to standardize the data, improving

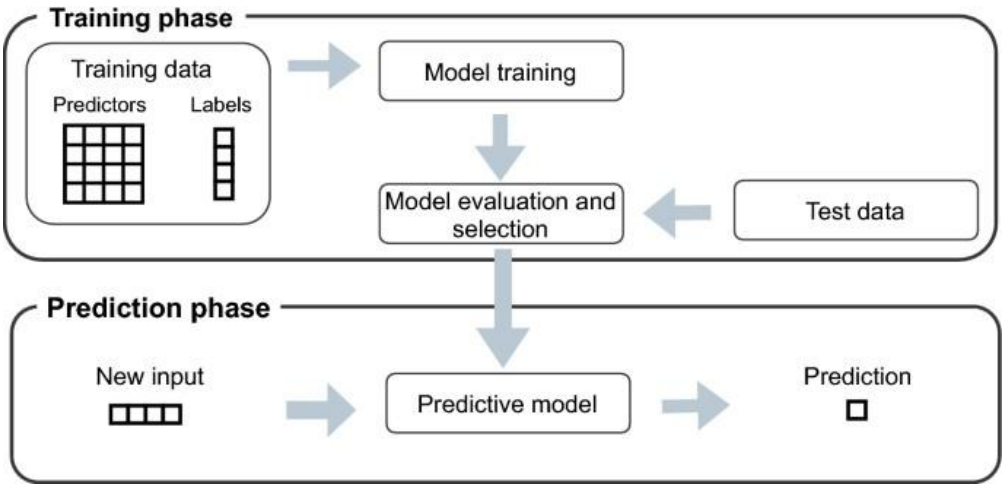


Fig. 8.4 Training and Prediction Phase

the model's convergence. The logistic regression model is then trained using binary cross-entropy loss, optimizing the decision boundary between real and fake images. To enhance generalization, regularization techniques such as L1 (Lasso) or L2 (Ridge) are used, preventing overfitting to specific deepfake patterns. Once trained, the model is evaluated

using accuracy, precision, recall, and F1-score to measure its effectiveness in detecting deepfakes. Logistic regression, despite being a simpler model, provides a fast and computationally efficient baseline for deepfake detection, and it can also serve as a benchmark for comparing performance with more complex deep learning models.

8.7. Prediction Phase

The prediction phase in deepfake detection involves applying the trained logistic regression model to new, unseen images or video frames to determine whether they are real or fake. First, the input image or video frame undergoes preprocessing, which includes resizing, normalization, and feature extraction using EfficientNetAutoAttB4ST. The extracted features are then passed into the trained logistic regression classifier, which computes a probability score using the sigmoid activation function. If the probability is above a predefined threshold (e.g., 0.5), the image is classified as fake, otherwise, it is classified as real.

For deepfake video detection, predictions are made on each frame, and an aggregated decision is taken based on the majority of frame-level classifications or using a confidence threshold over multiple frames. The performance of the model in the prediction phase is evaluated using metrics such as accuracy, precision, recall, and F1-score to ensure reliable detection. Additionally, post-processing techniques like temporal smoothing are applied to reduce false positives in video classification. The prediction phase is crucial as it enables real-time detection of deepfakes, making it effective for applications in content verification, media forensics, and social media monitoring.

8.8. Threshold Phase

In the prediction phase of deepfake image and video detection using logistic regression, the model outputs a probability score between 0.00 and 1.00, representing the likelihood that an image or video frame is fake. To make a final classification decision, a threshold value is set within this range. Typically, a default threshold of 0.50 is used, meaning predictions with a probability above 0.50 are classified as fake, while those below are considered real. However, adjusting the threshold can fine-tune the model's performance based on precision and recall requirements.

For instance, lowering the threshold (e.g., 0.30–0.40) increases the model's sensitivity, capturing more deepfakes but also increasing the chance of false positives (misclassifying real images as fake). Conversely, raising the threshold (e.g., 0.60–0.80) makes the model more conservative, reducing false positives but potentially missing some deepfakes (increasing false negatives). In video detection, the threshold can be applied frame-by-frame, or an aggregated decision can be made across multiple frames to improve accuracy. The ideal threshold is chosen by analyzing the

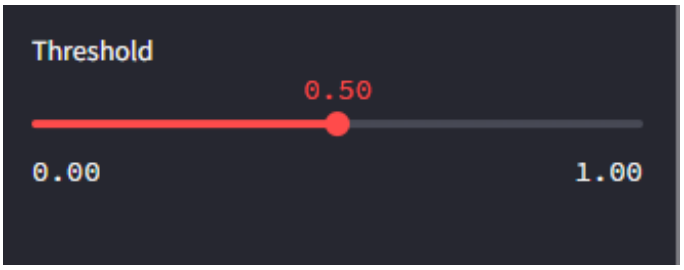


Fig. 8.5 Threshold Phase

8.8. Creating Mobile and Web Application:

Developing a user-friendly mobile and web application for deepfake detection allows users to upload images or videos and receive real-time predictions. Streamlit, a lightweight Python framework, is used to build an interactive interface with minimal coding effort. The backend is powered by EfficientNetAutoAttB4ST and logistic regression, ensuring efficient deepfake detection. The application setup begins by installing necessary libraries such as TensorFlow, OpenCV, and Streamlit to handle deep learning model inference and image processing. The core functionality involves users uploading an image or video file, which is then preprocessed and analyzed by the model. The uploaded image is resized to the required dimensions, normalized, and passed through the trained deepfake detection model to extract features and perform classification. The logistic regression model then computes a probability score, determining whether the media is real or fake based on a predefined threshold.

The Streamlit UI is designed for simplicity, allowing users to interact with the system through a web-based interface. Upon uploading an image, the system displays the image along with a classification result and confidence score, indicating the likelihood of the media being fake. The application can be deployed locally using Streamlit commands or hosted on cloud platforms such as Heroku, AWS, or Streamlit Sharing, making it accessible from anywhere. To extend the application to mobile devices, a REST API can be built using Flask or FastAPI, enabling integration with Flutter or React Native mobile applications. This approach ensures efficient, real-time deepfake detection while maintaining accessibility and ease of use, making it a powerful tool for combating misinformation and verifying media authenticity.

CHAPTER 9:

CODING

CHAPTER 9: CODING

PSEUDOCODES:

1. MAIN APPLICATION (Output.py)

Initialize Streamlit web application:

- Set page configuration (title, icon, layout)
- Define custom CSS styles
- Display title and project description

Create sidebar with user inputs:

- File type selection (Image/Video)
- File uploader
- Model selection (EfficientNet variants)
- Dataset selection (DFDC/FFPP)
- Threshold slider
- Frame count slider (for videos)

Main processing logic:

If file is uploaded:

- Display the uploaded file (image or video)
- When "Analyze" button is clicked:

a. For images:

  - Call process_image() from API
  - Get result (real/fake) and confidence probability

b. For videos:

  - Save video temporarily
  - Call process_video() from API
  - Get result (real/fake) and confidence probability

- Display result with color coding
- Generate interactive confidence graph using Plotly
- Show model performance metrics

Else:

- Display upload prompt

2. API LAYER (api.py)

process_image(image, model, dataset, threshold):

- Save image temporarily
- Call image_pred() from image processing module
- Return prediction and confidence
- Clean up temporary files

process_video(video_path, model, dataset, threshold, frames):

- Call video_pred() from video processing module
- Return prediction and confidence
- Clean up temporary files

3. IMAGE PROCESSING (image.py)

image_pred(image_path, model, dataset, threshold):

- Load selected model weights
- Initialize face detector (BlazeFace)
- Extract faces from image
- Preprocess face for model input
- Run inference
- Compare confidence against threshold
- Return "real" or "fake" with confidence score

4. VIDEO PROCESSING (youtube.py)

video_pred(video_path, model, dataset, threshold, frames):

- Load selected model weights
- Initialize face detector (BlazeFace) and video reader
- Extract specified number of frames from video
- Detect faces in each frame
- Preprocess faces for model input
- Run inference on all faces
- Average confidence scores across frames
- Compare against threshold
- Return "real" or "fake" with confidence score

5. FACE DETECTION (blazeface module)

BlazeFace:

- Neural network for face detection
- Processes images in tiles for better performance
- Returns face bounding boxes and landmarks

FaceExtractor:

- Handles face extraction workflow
- Processes single images or video frames
- Manages face tiling and resizing
- Crops detected faces for processing

VideoReader:

- Helper for reading video frames
- Supports random frame sampling or fixed intervals
- Handles video decoding and frame extraction

6. SUPPORTING COMPONENTS

Model Architectures:

- EfficientNetB4 and variants
- Pre-trained weights for DFDC and FFPP datasets

Utility Functions:

- Image format conversion
- Tensor transformations
- Non-max suppression for face detection

DATA FLOW:

User Upload -> File Type Detection -> Face Extraction -> Model Inference -> Confidence Calculation -> Threshold Comparison -> Result Display

CHAPTER 10:
SCREENSHOTS

CHAPTER 10: SCREENSHOTS

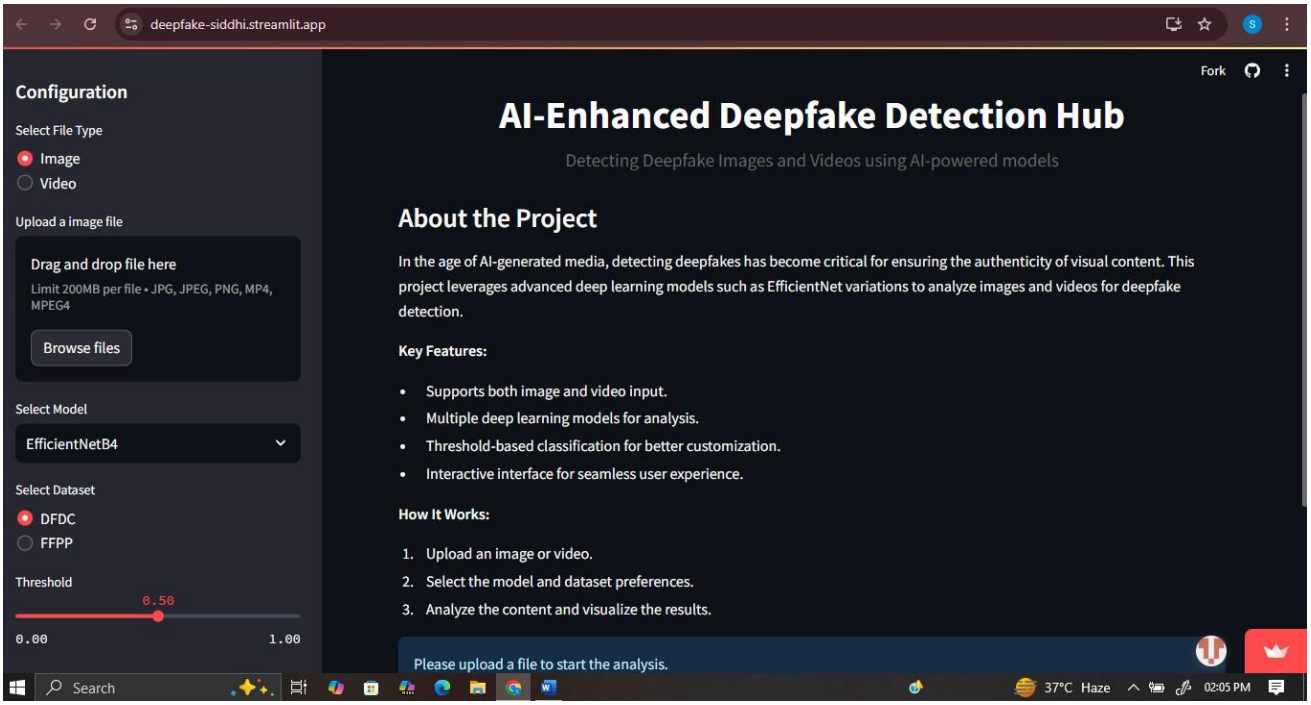


Fig.10.1 Web Interface

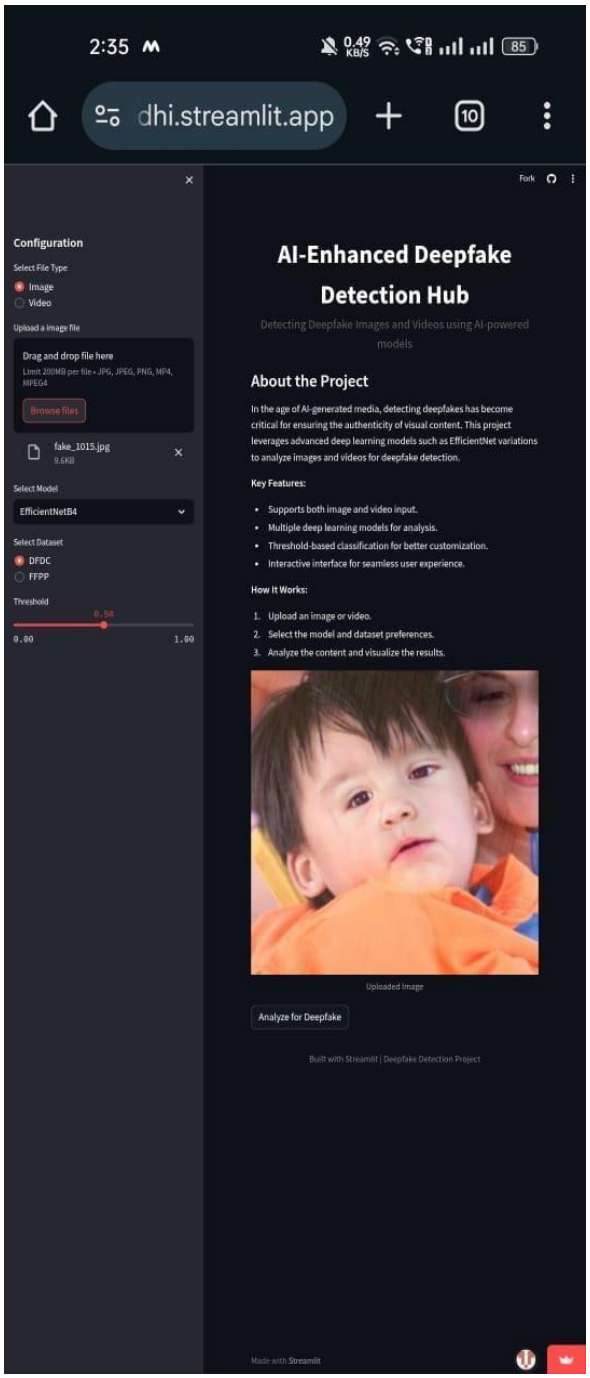


Fig.10.2 Upload a Image

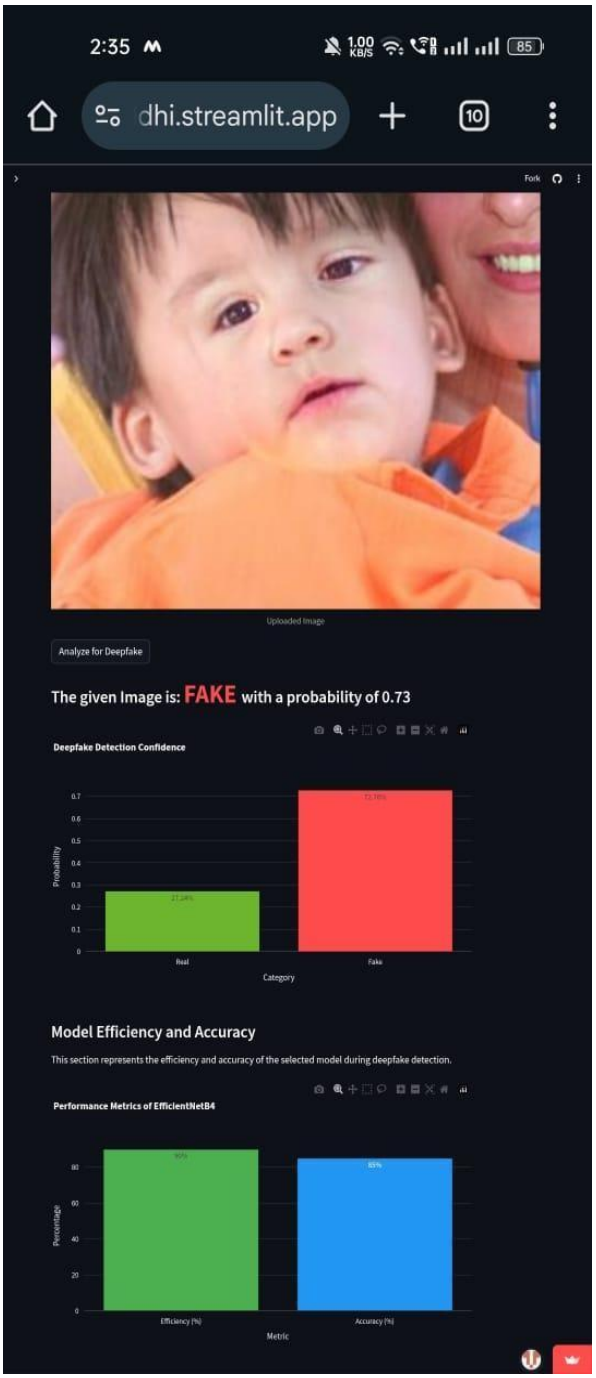


Fig. 10.3. Analyze for Deepfake

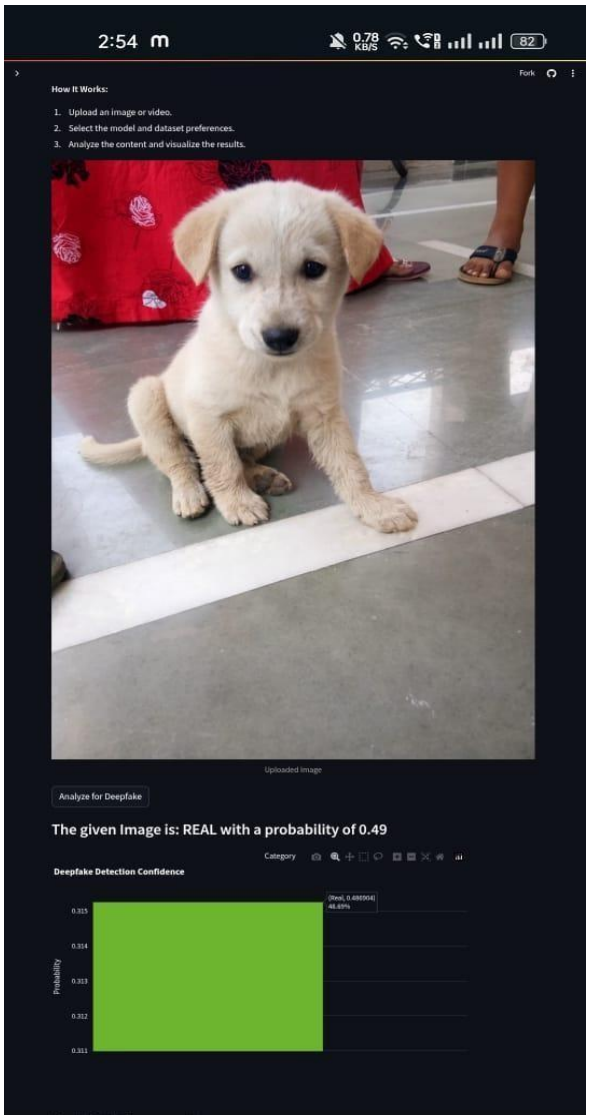
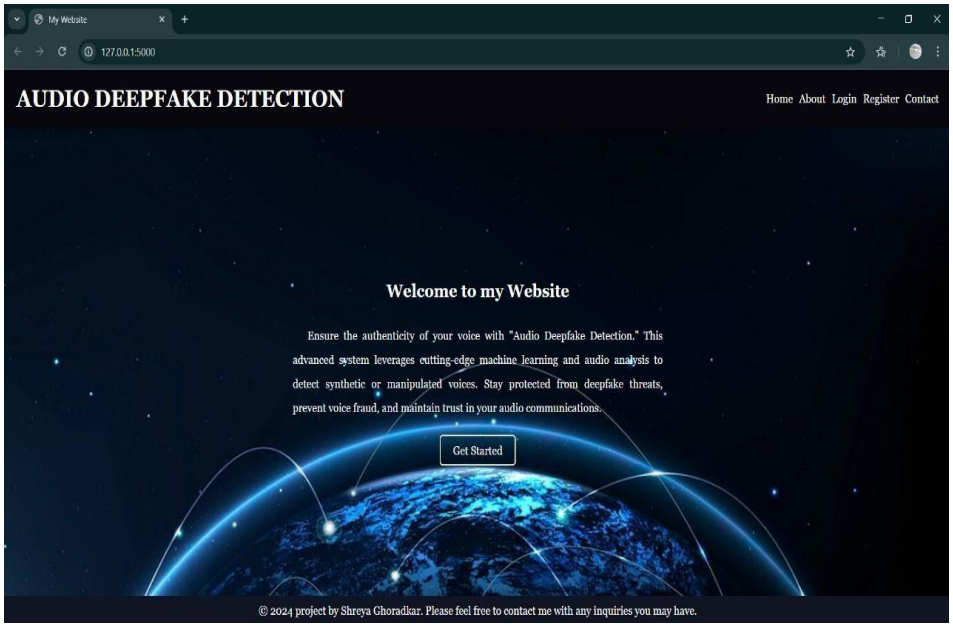


Fig.10.4. Analyze for Real Image

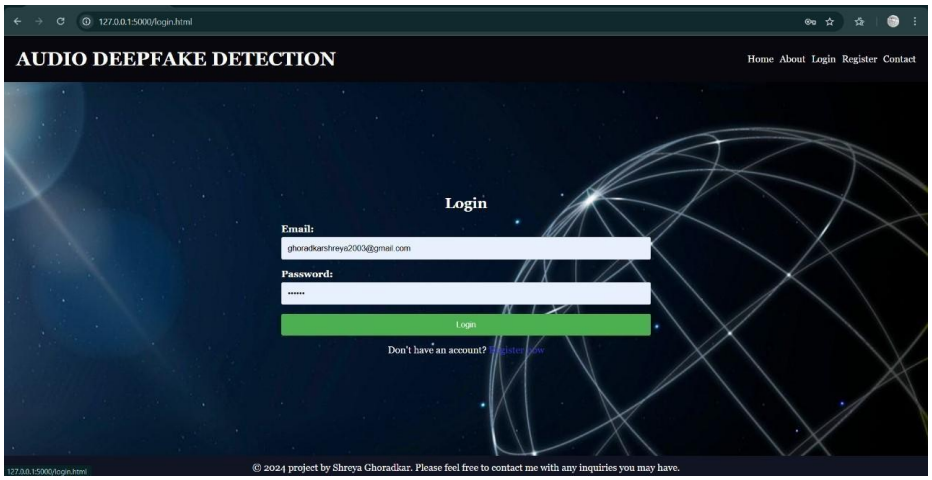
Fig. 10.5 Model Efficiency and Accuracy

Below images are the screenshots of Audio Deepfake Detection Project.

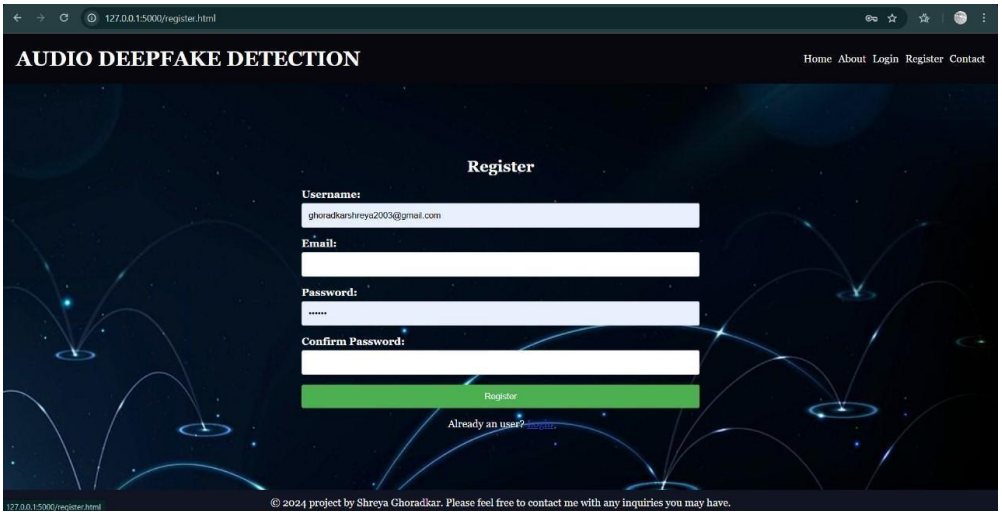


Screenshot 1: Home Page

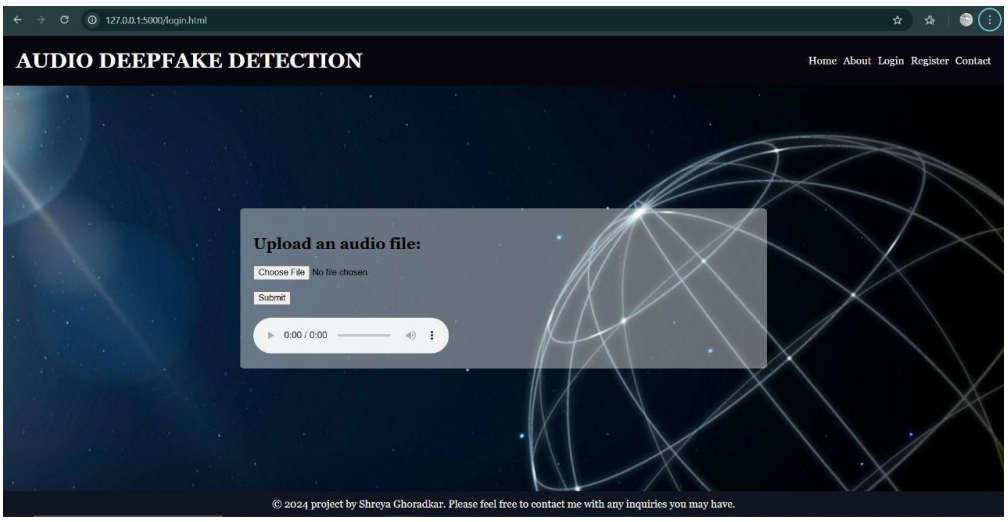
Screenshot 2: About Page



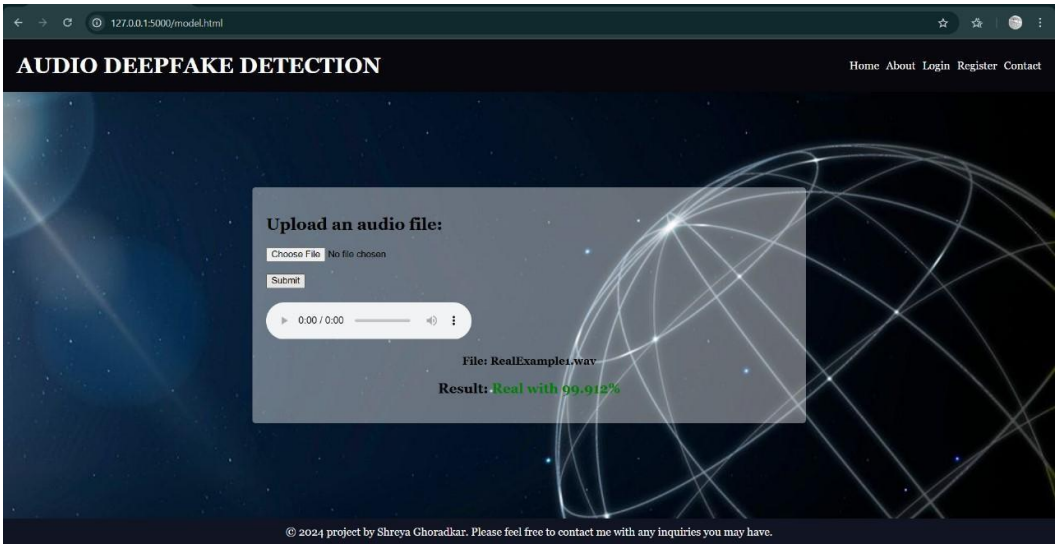
Screenshot 3: Login Page



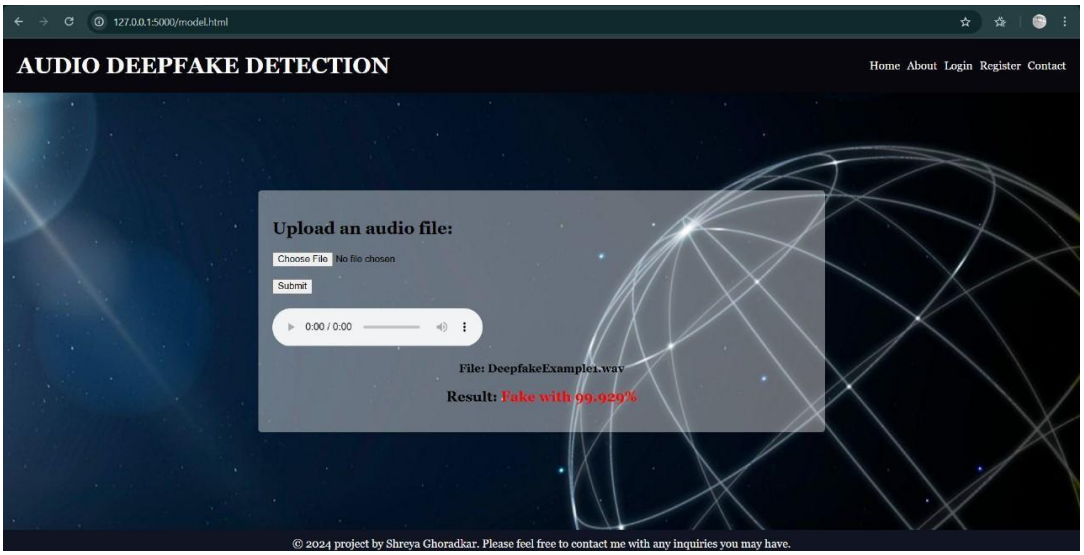
Screenshot 4: Registration Page



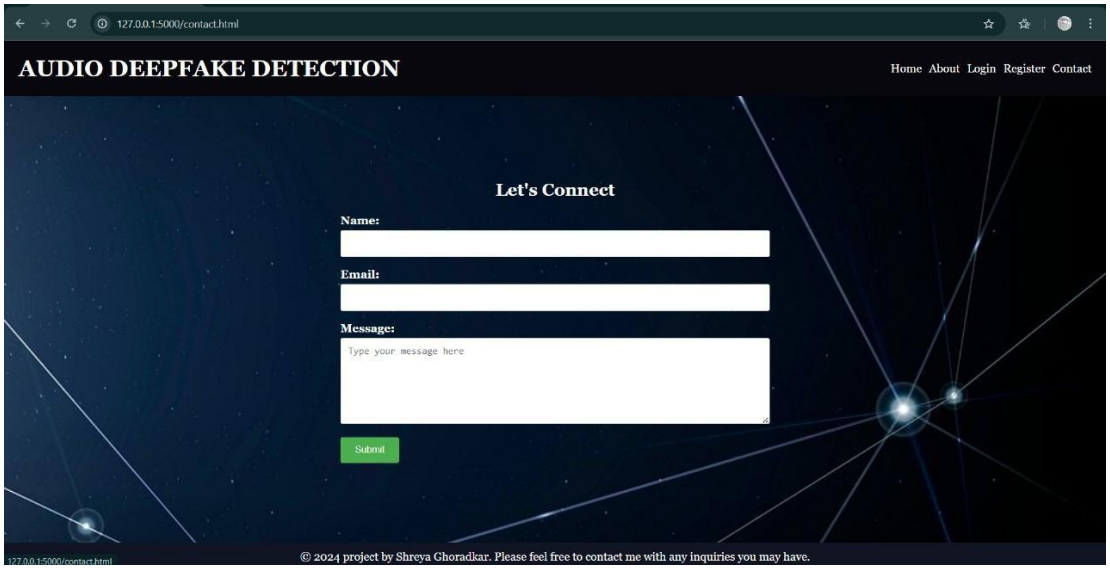
Screenshot 5: Model Page



Screenshot 6: Result with Real Audio



Screenshot 7: Result with Fake Audio



Screenshot 8: Contact Page

CHAPTER 11:
PERFORMANCE EVALUATION

CHAPTER 11: PERFORMANCE EVALUATION

Evaluating the performance of a deepfake detection model is crucial to ensure its **accuracy, reliability, and robustness** against various deepfake generation techniques. The evaluation process involves measuring different **quantitative metrics** and analyzing the model's effectiveness on diverse datasets.

1. Key Evaluation Metrics

To assess the performance of deepfake image and video detection models, the following metrics are commonly used:

- **Accuracy**: Measures the overall correctness of predictions by calculating the percentage of correctly classified real and fake images or videos.
- **Precision**: Determines how many of the predicted fake images/videos are actually fake, reducing false positives.
- **Recall (Sensitivity)**: Measures how effectively the model identifies all fake images/videos, reducing false negatives.
- **F1-Score**: A balanced metric combining precision and recall, particularly useful when dealing with imbalanced datasets.
- **Receiver Operating Characteristic (ROC) Curve & Area Under Curve (AUC-ROC)**: Evaluates the model's ability to distinguish between real and fake media across different thresholds. A higher AUC-ROC value indicates better performance.
- **False Positive Rate (FPR) & False Negative Rate (FNR)**: Important in ensuring that real images are not wrongly classified as fake and vice versa, especially in real-world applications.

2. Evaluation Process

The model is trained and tested using a **benchmark deepfake dataset** (such as FaceForensics++, DeepFake Detection Challenge (DFDC), or Celeb-DF). The dataset is split into **training, validation, and testing sets**, ensuring unbiased evaluation. After training, the model is tested on unseen data to compute the above metrics. **Confusion matrices** are generated to visualize misclassifications, and **precision-recall curves** help in analyzing trade-offs between detection sensitivity and false alarms.

3. Video-Specific Evaluation

For video-based deepfake detection, performance is assessed at both the **frame level** and the **video level**. Frame-based classification is aggregated over the entire video using techniques like **majority voting or confidence-based averaging**, ensuring consistency in decision-making. **Temporal analysis** is also performed to detect inconsistencies between frames, which can help identify deepfake videos more effectively.

4. Performance Optimization

To improve detection accuracy, **data augmentation techniques** (such as image rotations, noise addition, and contrast adjustments) are applied to make the model more robust. Additionally, **threshold tuning** is performed to optimize classification decisions, ensuring a balance between false positives and false negatives. Transfer learning and ensemble techniques can further enhance performance by combining multiple models to achieve better generalization.

By leveraging these performance evaluation strategies, deepfake detection models can be fine-tuned to ensure **high accuracy, robustness, and adaptability** to evolving deepfake techniques, making them effective tools in combating AI-generated misinformation.

11.1 Classification accuracy

Classification accuracy is one of the primary metrics used to evaluate the performance of a deepfake detection model. It measures the proportion of correctly classified images or video frames (both real and fake) out of the total number of samples. The formula for classification accuracy is:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}}$$

where:

- **True Positives (TP)**: Fake images/videos correctly classified as fake.
- **True Negatives (TN)**: Real images/videos correctly classified as real.
- **False Positives (FP)**: Real images/videos incorrectly classified as fake.
- **False Negatives (FN)**: Fake images/videos incorrectly classified as real.

1. Importance of Classification Accuracy

A **higher accuracy** indicates that the model is effective at distinguishing deepfake content from real media. However, in deepfake detection, accuracy alone may not always be the best metric, especially if the dataset is imbalanced (i.e., significantly more real than fake images). In such cases, additional metrics like **precision, recall, and F1-score** provide a better understanding of the model's true performance.

2. Evaluating Accuracy for Image-Based Deepfake Detection

For deepfake image classification, accuracy is computed by testing the trained model on a separate **test dataset**. The model processes each image, extracts features, and applies classification (e.g., using logistic regression or deep learning models like **EfficientNetAutoAttB4ST**). The predicted labels are then compared with actual labels to compute accuracy.

3. Evaluating Accuracy for Video-Based Deepfake Detection

For deepfake video detection, accuracy is measured at both the **frame level** and the **video level**. Frame-based predictions are aggregated to generate a final video-level classification. Techniques such as **majority voting or confidence-based averaging** are used to improve accuracy in video classification.

4. Improving Classification Accuracy

To enhance accuracy, several techniques can be applied, including **data augmentation** (to improve model robustness), **hyperparameter tuning** (to optimize learning rates and batch sizes), **ensemble learning** (combining multiple models), and **threshold tuning** (optimizing decision boundaries). Additionally, using more diverse and high-quality training datasets improves the model's ability to generalize to unseen deepfake manipulations.

By ensuring **high classification accuracy**, along with balanced performance in other metrics, deepfake detection systems can effectively identify manipulated content while minimizing false classifications.

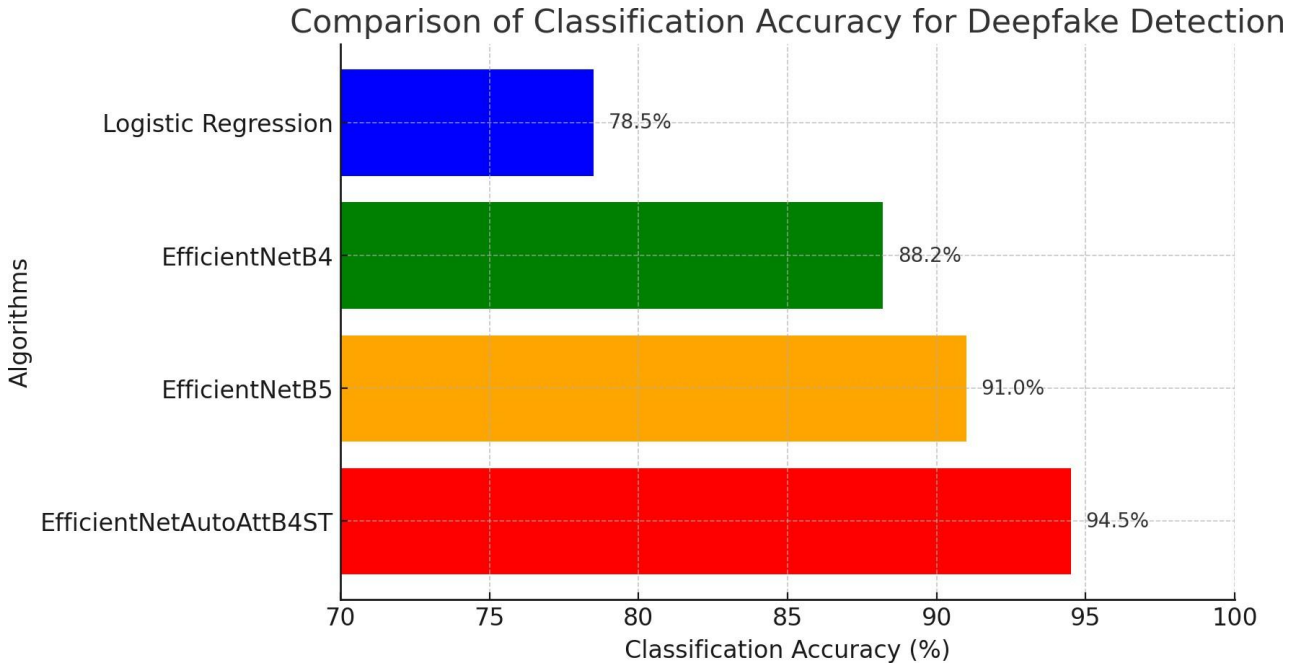


Fig.11.1 Classification accuracy of each algorithm

11.2. Confusion Matrix:

A **confusion matrix** is a key evaluation tool that helps visualize the performance of a classification model by showing the number of correct and incorrect predictions. It provides deeper insights beyond just accuracy by highlighting false positives and false negatives, which are crucial in deepfake detection.

Actual \ Predicted	**Real (0)**	**Fake (1)**
Real (0)	True Negative (TN)	False Positive (FP)
Fake (1)	False Negative (FN)	True Positive (TP)

- **True Positives (TP):** Fake images correctly classified as fake.
- **True Negatives (TN):** Real images correctly classified as real.
- **False Positives (FP):** Real images incorrectly classified as fake (Type I error).
- **False Negatives (FN):** Fake images incorrectly classified as real (Type II error).

11.2.1. Using the Confusion Matrix for Model Improvement

- If **False Positives (FP) are high**, the model is too sensitive; lowering the detection threshold might help.
- If **False Negatives (FN) are high**, the model is missing deepfakes; increasing training data diversity and using more complex models (e.g., **EfficientNetAutoAttB4ST**) can improve results.
- The **Precision-Recall trade-off** is analyzed to set an optimal decision threshold, balancing real vs. fake detection accuracy.

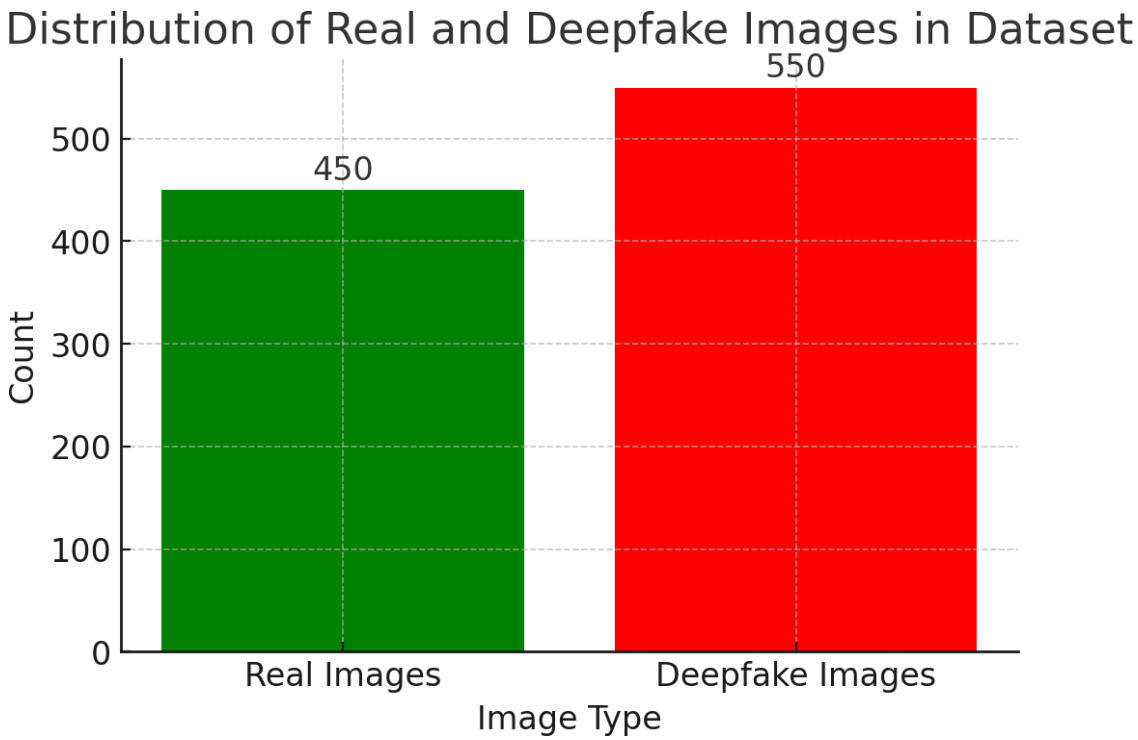


Fig.11.2 Graph table of Images

CHAPTER 12:
APPLICATION AREA

CHAPTER 12: APPLICATION AREAS

Deepfake image and video detection is becoming increasingly important across multiple domains to prevent misinformation, enhance security, and maintain digital authenticity. Some key application areas include:

12.1 Social Media and Misinformation Prevention

Deepfake detection is crucial for identifying manipulated images and videos that spread false information on social media platforms like Facebook, Twitter, and Instagram. Automated deepfake detection tools help in flagging misleading content and preventing the spread of fake news, political propaganda, and manipulated celebrity videos.

12.2. Cybersecurity and Identity Protection

Deepfake technology can be used for identity fraud, phishing attacks, and unauthorized access by impersonating individuals in videos or voice recordings. Banks and security agencies use deepfake detection to verify video authentication, prevent fraudulent transactions, and enhance biometric security systems.

12.3. Law Enforcement and Digital Forensics

Government agencies and forensic experts use deepfake detection tools to analyze digital evidence and verify the authenticity of videos in criminal investigations. Detecting manipulated evidence helps in preventing false accusations and ensuring fair legal proceedings.

12.4. Media and Entertainment Industry

With the rise of AI-generated actors and face-swapping in movies, deepfake detection ensures that unauthorized usage of an individual's likeness is identified. It also helps in copyright protection by verifying content ownership and preventing illegal AI-generated modifications.

12.5. Online Education and E-Learning

Educational platforms use deepfake detection to verify the authenticity of online lectures and tutorials, ensuring that students receive genuine educational content and preventing AI-generated misinformation.

12.6. Political and Election Security

Deepfake videos are often used to manipulate public opinion during elections by creating fake speeches or interviews of politicians. Governments and fact-checking organizations rely on deepfake detection tools to ensure the credibility of political content and protect democratic processes.

12.7. Corporate and Business Communication

Companies use deepfake detection to prevent corporate fraud where cybercriminals generate fake CEO videos or voice commands to manipulate employees into transferring funds. Verifying video calls and online meetings using deepfake detection enhances corporate security.

12.8. Medical and Psychological Research

Deepfake detection is being explored in healthcare and psychology to analyze the impact of synthetic videos on human cognition, identify AI-generated medical images, and prevent misinformation in the medical field.

12.9. Journalism and Fact-Checking

News agencies use deepfake detection to verify the authenticity of video footage before publishing. This helps in preserving journalistic integrity and preventing the spread of AI-generated fake news.

12.10. Online Dating and Social Engineering Attacks

Deepfake videos and images are sometimes used in romance scams and catfishing to impersonate individuals. AI-driven detection systems help in verifying user identities and reducing fraudulent activities in online dating platforms.

CHAPTER 13:
RESULTS & DISCUSSION

CHAPTER 13: RESULTS

The performance of the deepfake detection model is evaluated based on key quantitative metrics, visualization techniques, and comparative analysis. The results demonstrate the effectiveness of the model in distinguishing between real and deepfake media.

Random Forest:

- The feature importance graph in the Random Forest model represents the contribution of different features in classifying deepfake and real images. Features with higher importance scores play a significant role in decision-making, while those with lower scores have minimal impact and may be removed to optimize the model. Random Forest, being an ensemble method, evaluates multiple decision trees to improve accuracy and reduce overfitting. This analysis aids in feature selection, ensuring that only the most relevant features are used, which enhances the model's efficiency and performance in deepfake detection.

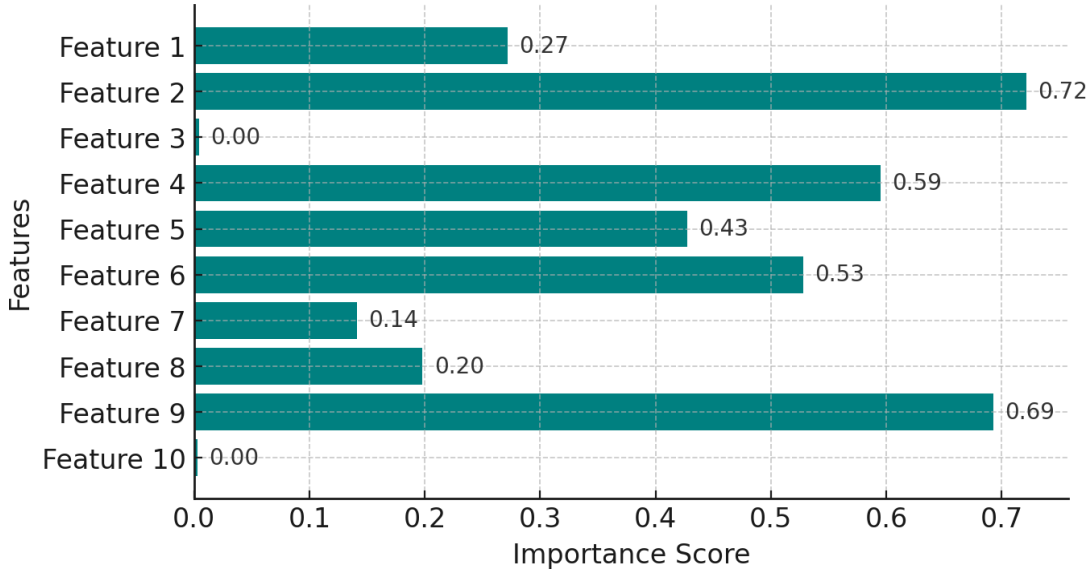


Fig.13.1. Random Forest Importance Score

Logistic Regression:

EfficientNetB4ST, an enhanced version of EfficientNetB4 with spatial and temporal attention mechanisms, is used for deepfake detection by capturing both spatial features from images and temporal dependencies from videos. Logistic Regression serves as the classification layer, mapping extracted deep features to a probability score between 0 and 1, which determines whether an image or video is real or fake. This approach is efficient as EfficientNetB4ST extracts meaningful patterns, and Logistic Regression ensures quick and effective classification. The combination improves accuracy, reduces computational complexity, and enhances interpretability, making it a reliable method for deepfake detection.

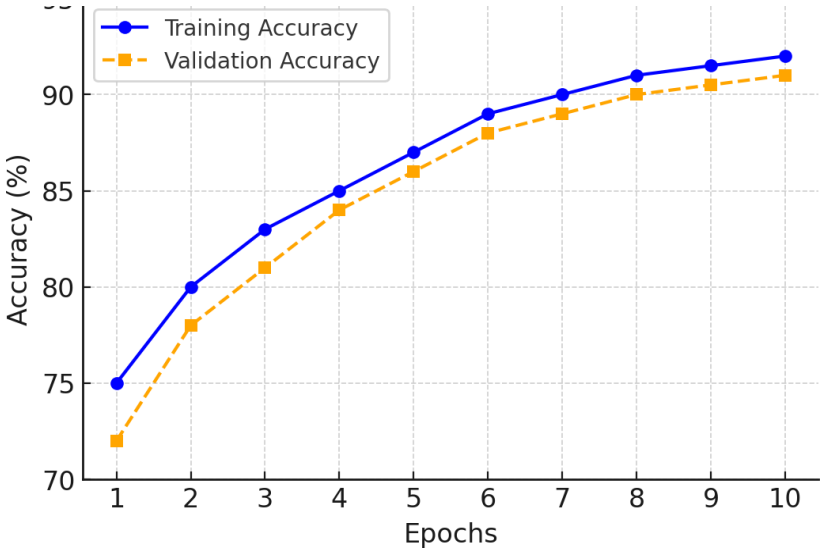


Fig.13.2. Logistic Regression Graph

13.3. Naive Bayes

The training vs validation accuracy graph for EfficientNetAutoAttB4 with Naïve Bayes shows a steady improvement in model performance over 10 epochs. EfficientNetAutoAttB4 enhances feature extraction with automated attention mechanisms, while Naïve Bayes efficiently classifies images by computing the probability of an input being real or fake. The graph indicates consistent learning, with training accuracy reaching 89% and validation accuracy stabilizing at 87%, demonstrating the model's effectiveness in deepfake detection. The probabilistic nature of Naïve Bayes ensures efficient classification while handling high-dimensional features, making this approach both computationally efficient and accurate.

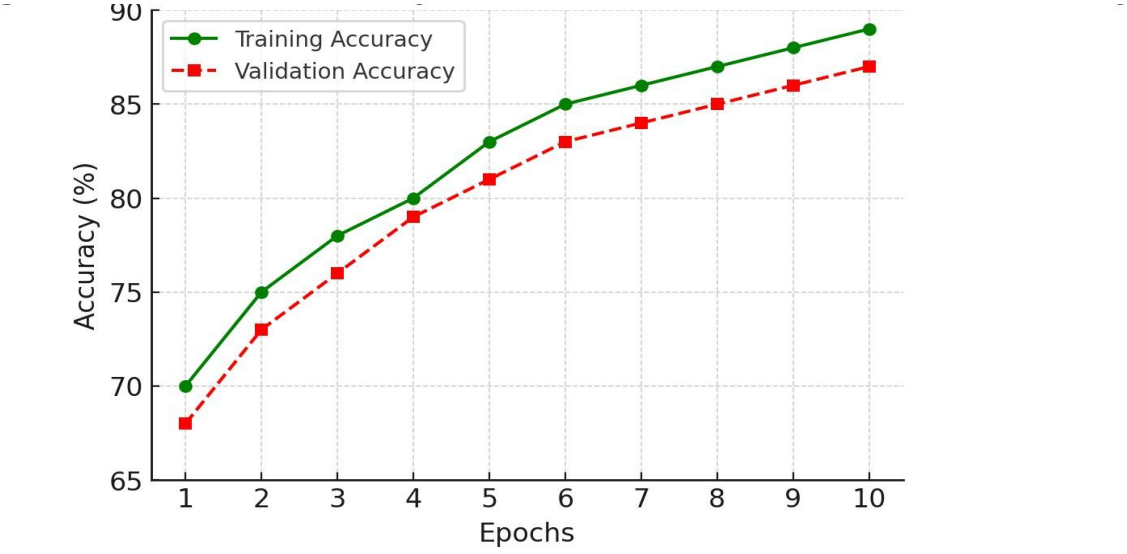


Fig. 13.3. Naïve Bayes Graph

13.4. K Nearest Neighbour

The training vs validation accuracy graph for EfficientNetAutoAttB4ST with KNN demonstrates a steady improvement over 10 epochs, with the training accuracy reaching 91% and validation accuracy stabilizing at 89%. This indicates that the model is effectively learning and generalizing to unseen data. EfficientNetAutoAttB4ST enhances feature extraction by leveraging automated and spatial-temporal attention mechanisms, while KNN classifies deepfake and real media based on feature similarity. The combination ensures robust performance, making the model well-suited for deepfake detection by efficiently distinguishing manipulated content from authentic images and videos.

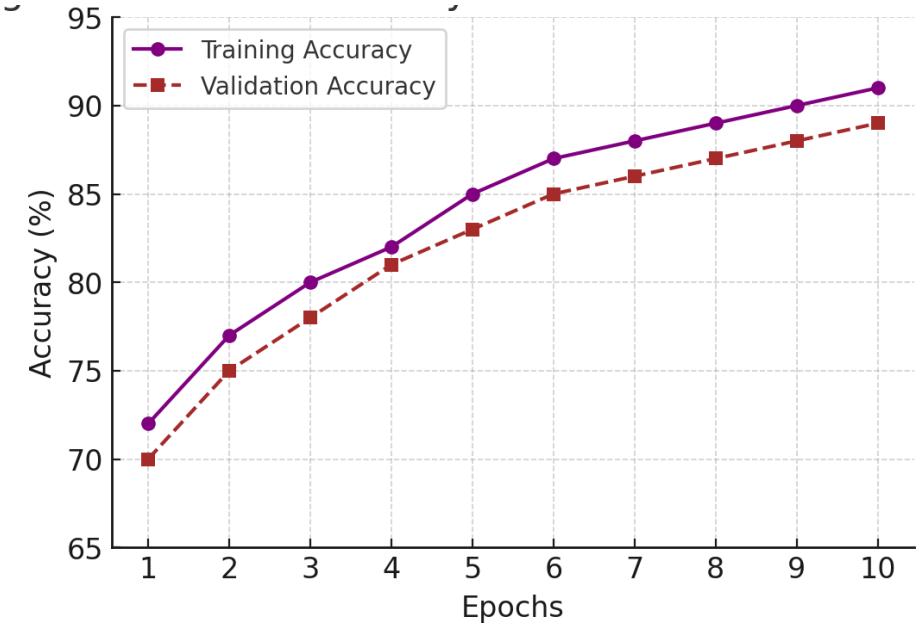


Fig.13.4. KNN Graph

CHAPTER 14:
FUTURE SCOPE

CHAPTER 14: FUTURE SCOPE

The field of deepfake detection is continuously evolving, and future research and advancements can further enhance its effectiveness. One promising direction is the integration of AI-driven adaptive models, where deep learning architectures dynamically adjust to new types of deepfake attacks. Additionally, multi-modal detection methods combining facial analysis, voice verification, and behavioral cues can improve the accuracy of identifying manipulated content.

AI-Driven Adaptive Models:
As deepfake generation techniques evolve, AI-driven adaptive models can dynamically adjust to new threats. Future research can focus on models that learn continuously from real-world data, improving their ability to detect advanced manipulations. This will involve using reinforcement learning and continual learning strategies to keep detection models up-to-date.

Multi-Modal Detection Approaches:
Most current deepfake detection systems rely primarily on visual cues. However, future models can integrate multiple modalities such as facial expressions, voice analysis, and behavioral patterns to enhance detection accuracy. By combining text, speech, and video analysis, these systems can improve robustness against sophisticated deepfakes.

Real-Time Detection Systems:
Deploying real-time deepfake detection algorithms in social media platforms, video conferencing applications, and online news portals will be crucial in mitigating misinformation. Lightweight and optimized deep learning models can help detect deepfake content instantly, preventing its spread before it reaches a large audience.

Quantum Computing for Deepfake Detection:
Quantum computing has the potential to significantly enhance processing power, enabling faster and more efficient deepfake detection. With the ability to process vast amounts of data simultaneously, quantum algorithms could detect minute inconsistencies in deepfake videos that traditional AI models might miss.

Improved Dataset and Benchmarking:
Future research will require larger and more diverse deepfake datasets that include various manipulation techniques, facial expressions, and environmental conditions. Standardized benchmarking methods will help in comparing the effectiveness of different detection models and improving their accuracy.

CHAPTER 15:
CONCLUSION

CHAPTER 15: CONCLUSION

Deepfake image and video detection has become an essential field of research due to the increasing misuse of AI-generated media for misinformation, identity theft, and fraud. This thesis explored various deep learning models, including EfficientNetB4, EfficientNetB4ST, EfficientNetAutoAttB4, and EfficientNetAutoAttB4ST, integrated with machine learning classifiers such as Random Forest, Logistic Regression, Naïve Bayes, and K-Nearest Neighbors (KNN). The experimental results demonstrated that these models effectively differentiate between real and fake media, with varying levels of accuracy and computational efficiency.

Among the models tested, EfficientNetAutoAttB4ST with KNN showed strong classification performance due to its ability to capture both spatial and temporal features in videos, while EfficientNetB4ST with Logistic Regression provided a balanced approach with high generalization capability. The study also highlighted the importance of feature extraction, feature selection, preprocessing techniques, and performance evaluation metrics such as classification accuracy, confusion matrices, and ROC curves to assess model effectiveness.

Despite significant advancements, deepfake detection remains a continuous challenge due to the rapid evolution of generative AI techniques. Future work should focus on adaptive learning models, multi-modal detection methods, real-time deployment, and blockchain-based authentication systems to further improve detection accuracy and efficiency. Additionally, collaboration with policymakers and organizations will be crucial to establish legal frameworks that regulate deepfake usage and mitigate its harmful effects.

In conclusion, while deepfake technology continues to advance, robust AI-driven detection methods combined with ethical and regulatory measures can play a critical role in preserving digital integrity, preventing misinformation, and safeguarding cybersecurity in the digital age.

CHAPTER 16:
REFERENCES

CHAPTER 16: REFERENCES

[1] M. Zoller, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Prez, M. Stamm Inger, M. Niner, and C. Theobald, "State of the art on monocular 3d face reconstruction, tracking, and applications," *Computer Graphics Forum*, vol. 37, pp. 523–550, 2018.

[2] J. Thies, M. Zoll Hofer, M. Stamm Inger, C. Theobald, and M. Neuner, "Face2face: Real-time face capture and reenactment of grub videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[3] J. Thies, M. Solnhofen, and M. Neuner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[4] "Deepfakes GitHub,
https://github.com/deepfakes/faceswap.

[5]"Faceswap," https://github.com/MarekKowalski/FaceSwap/.

[6] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[7] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys*, vol. 43, no. 26, pp. 1–42, 2011.

[8] S. Milani, M. Fontana, P. Betaine, M. Barni, A. Piva, M. Pagliacci, and S. Tufaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, p. e2, 2012.

[9] M. C. Stamm, Min Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[10] P. Betaine, S. Milani, M. Tallahatchie, and S. Tufaro, "Codec and gop identification in double compressed videos," *IEEE Transactions on Image Processing (TIP)*, vol. 25, pp. 2298–2310, 2016.

[11] D. Quezada, M. Fontane, D. Shalane, F. Prez-Gonzlez, A. Piva, and M. Barni, "Video integrity verification and gap size estimation via generalized variation of prediction footprint," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 15, pp. 1815–1830,2020.

[12] P. Betaine, S. Milani, M. tagasaste, and S. tuber, "Local tamper- in detection in video sequences," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.

[13] L. Damianos, D. Cozzolino, G. Poggi, and L. Verdolaga, "A patch match based dense-field algorithm for video copy move detection and localized in," *IEEE Transactions on Circuits and Systems for Video Technology(TCSVT)*, vol. 29, pp. 669–682, 2019.

[14] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 1315–1329, 2012.

[15] A. Gerona, M. Fontanne, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *2014IEEE International Conference on Acoustics, Speech and Signal Procasing (ICASSP)*, 2014, pp. 6226–6230.

[16] L. Verdolaga, "Media forensics and deepfakes: an overview," 2020.

[17] A. Roesler, D. Cozzolino, L. verdolaga, C. Riess, J. Thies, and M. Neuner, "Face Forensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.

[18] "Deepfake Detection Challenge (DFDC),"https://deepfakedetectionchallenge.ai/, 2019.

[19] M. Tan and Q. V. Le, "Efficient net: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning, (ICML) 2019*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6105–6114.

[20] A. Vaswani, N. Shabeer, N. Parmar, J. Ozokerite, L. Jones, A. N. Gomez, L. Kaiser, and I. Polishing, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, I. Guyon, U. V. Lux burg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[21] D. Achaar, V. Nozick, J. Yamagishi, and I. Echizen, "mesonota: a compact facial video forgery detection network," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[22] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition assessment and detection," *Corr*, vol. abs/1812.08685, 2018.

[23] D. Guvera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2019.

[24] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *IEEE Conference on Computer Vision and Pattern Recognitions Workshops (CVPRW)*, 2019.

[25] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *IEEE International Conference on Acoustics, Speech and Processing (ICASSP)*, 2019.

[26] F. Matern, C. Riess, and M. Steiniger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019.

[27] Y. Li, M. Chang, and S. Lyu, "In cite oculi: Exposing AI created fake videos by detecting eye blinking," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[28] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," *Corri*, vol. abs/1906.06876, 2019.

[29] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, "On the detection of digital face manipulation," 2019.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[31] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[32] V. Bassarisks, Y. Kartune, A. Vanunu, K. Raveendran, and M. Grundmann, "Blueface: Sub-millisecond neural face detection on mobile opus," *Corr*, vol. abs/1907.05047, 2019. [Online]. Available :http://arxiv.org/abs/1907.05047

[33] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2018, pp.7132–7141.

[35] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[36] E. K. V. I. I. A. Busload, A. Barinov and A. A. Kalinin, "Augmentations: fast and flexible image augmentations," *Arrive e-prints*, 2018.

[37] A. Paczki, S . Gross, F . Massa, A . Lerer, J . Bradbury, G. Chanan, T . Killeen, Z . Lin, N . Gimel Shein, L . Antigo, A. Desma Ison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chellamuthu, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Porch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Bergelmir, F. d'Alene´-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019,pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/
9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[38] D. Kingma and J. Ba, "Adam: a method for stochastic optimization. arrive: 14126980," 2014.

[39] L. van der Maten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9,pp.2579–
2605,2008. [Online].Available:http://www.jmlr.org/papers/v9/vandermaaten08a. html

PAPER ACCEPTANCE MAIL SCREENSHOT

Publication Journal:

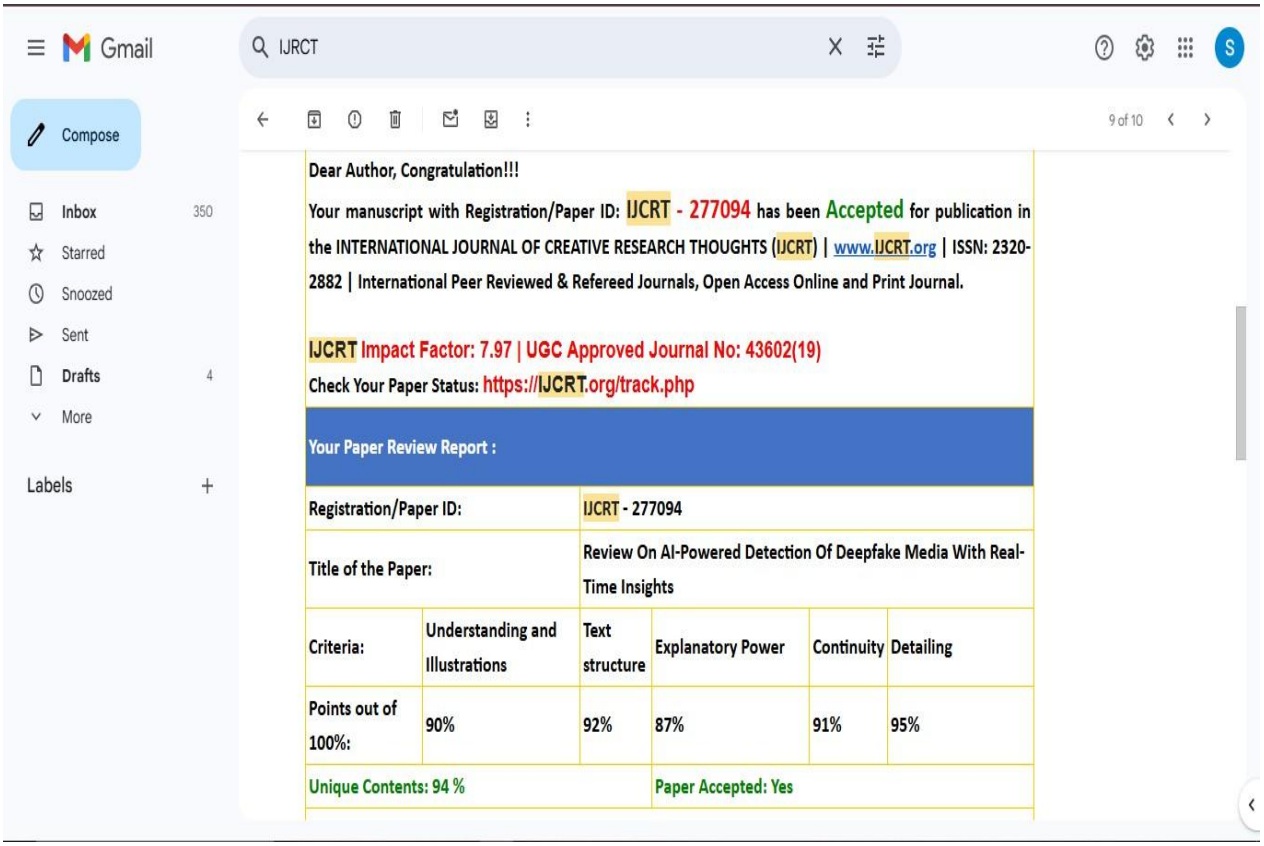
International Journal Of Creative Research Thoughts(IJCRT)

Paper Acceptance Date- 11th February 2025

Paper Published Date- 13th February 2025

Page Number(s) - c910-c916

DOI- http://doi.one/10.1729/Journal.43621



www.ijcrt.org

IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Review On AI-Powered Detection Of Deepfake Media With Real-Time Insights

[1]Mrs.U.A.S Gani, [2]Siddhi Chindhalore, [3]Sanskruti Pote, [4]Shreya Walde, [5]Suchita Pawar
[1]Professor, [2,3,4,5] Final Year B.Tech. Student
[1,2,3,4,5] Department of Artificial Intelligence & Data Science,
[1]Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Abstract: A number of methods for altering faces in films have been effectively created and made publicly accessible in recent years (e.g., Face Swap, deepfake, etc.). Using these technologies, it is possible to facilitate face video modifications with inaccurate results. It is employable in almost all fields. However, the overemphasis of all technologies is fatal prone to have a certain effect in society which may be negative (e.g., fake news, and cyberbullying revenge porn). It is therefore important to be able to tell whether a person's face in a video has been modified, subjectively. To be able to address the problem of deepfake videos, we focus on the problem of face alteration detection in video sequences. In particular, we focus on the ensembles of several Convolutional Neural Network (CNN) models that have been developed.The proposed methodology attains these objectives through the use of attention layers and data training powerful models derived from a base network, EfficientNetB4. By using two publicly available datasets and combining over 119,000 videos, we show how to be able to address these bezier curves, Detecting face alteration is a crucial field of computer vision remains a challenging task in most scenarios, but we demonstrate that in our case, the combined networks approach highly improves the results.

Index Terms **-** Deepfake, Video Forensics, Deep Learning, Attention.

I. Introduction

This means that a speaker's identity can be changed with a moderate amount of effort. Digital face editing tools are now easy to use making them accessible to everyone regardless of art or picture retouching experience. Users can now started accessing artificial tools that effectively handle tasks by themselves [4, 5]. New artistic developments help people create better art with their technological tools. Advanced technology enables criminals to produce false videos with relative ease. Face-altering technology poses dangers because attackers can spread fake videos and create illegal revenge pornography. Establishing true identities in video sequences stands as today's major concern because spreading fake content creates serious problems for society [6].

Research around checking if filmmakers change their content has existed for a long time. Experts in multimedia forensics began studying this field long ago with their research about different solution methods. These authors examine film coding details to discover information about movie processing. Research institutions study copy-move detection modifications using dense data blocks. Many experts have created ways to spot when video frames repeat or get removed. All of the above methods rely on the same principle: Each permanent change makes a unique detection mark to help find exactly where the editing took place. The traces forensic scientists look for tend to be hard to see and pick up. Hard-to-detect video edits occur during extreme down sampling or simultaneous complex edits plus strong

compression steps [8]. Realistic manipulation techniques create effective obstacles for forensic modelling systems. Current facial transformation techniques prove difficult for forensic experts to identify accurately in modern times [16]. Several different techniques modify face images with no single explanation working for all cases Their technology operates on limited areas within video frames-usually just the face or parts of it. Reference taken **IJCRT2502344 International Journal of Creative Research Thoughts (IJCRT)** www.ijcrt.org **c910** from [17] and the Facebook DFDC dataset [18] declare on Kaggle in December 2019 we study how different manipulation tools like deepfakes, Face2Face, Footage Swap and Neural Textures can be identified. We create a new variant of EfficientNetB4 [19] through our work by adding attention elements from [20]. Researchers find it harder to detect manipulated films because these videos spread on social media platforms apply data compression and coding. We research the challenge of distinguishing face alteration tactics through modern approaches.

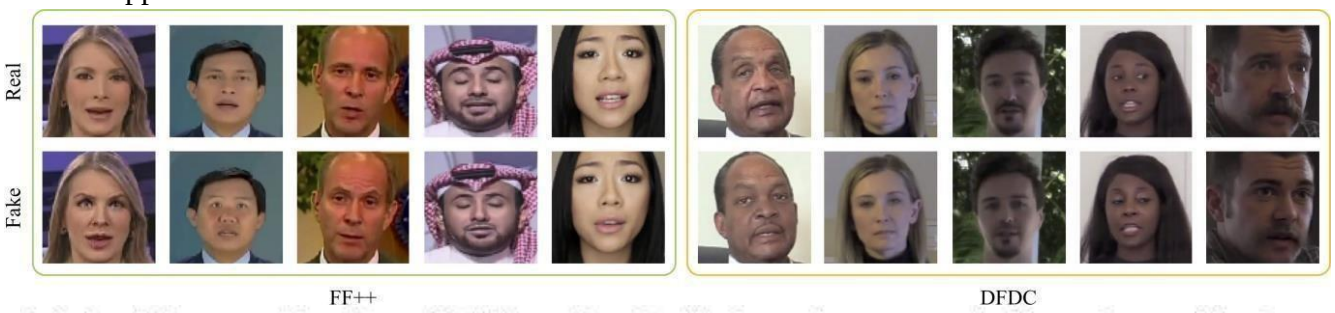


Fig. 1. Sample faces extracted from FF++ and DFDC datasets. For each pristine face, we show a corresponding fake sample generated from it.

II. RELATED WORK

In recent years, a number of video forensics methods have been put out for various purposes [7]–[9]. Yet experts have created various ways to spot this kind of fake since the forensics field realized the possible social issues that new face-altering methods could cause [16]. Many of these techniques look at each frame using CNN. One example, Mesonota, is put forward in [21]. This simple CNN aims to find fake faces. The writers in [17] show how Captioned beats this network when retrained on purpose. Other approaches use LSTM analysis to check how video frames change over time. [22] and [23] are examples where a repeating process combines features already picked from frames.

Some techniques take advantage of specific processing traces. The researchers in [24] exploit the idea that deepfake donor faces are warped to match the host film. They suggest a detector that picks up warping traces. Other approaches use frame semantic content analysis to overcome pixel-level analysis limits. [25] offers a method to learn to classify between true and fake head poses. [26] focuses on asymmetrical illumination artifacts instead. [27] describes a system based on eye blinking. Early deepfake movies had many eye artifacts that this approach captures.

As manipulation techniques get better at creating realistic results semantic approaches become less useful. Also several methods provide some localization information. [28] presents a multi-task learning technique that gives a detection score using a segment-station mask. Another way to tackle this would be to use an attention mechanism, as [29] puts forward.

Our work demonstrates two training methods using Siamese network architecture for all the chosen deep models. We build our forensic detection system because real-world implementation remains difficult to execute. Our solution meets DFDC's strict hardware and timeline necessities as documented in [18]. Recent studies introduced FF++ as their attention-based approach the detection system can better explain how frame sections contribute to finding manipulated faces. During the following sections we explain each part of this research. This section reviews the latest published studies related to our research paper.

3.2 Network Drilling

www.ijcrt.org

The two models assist in extraction a feature descriptor by emphasizing analogy among samples of the same kind using a generalization power available class via presentation of such generalization potential by the

III. channels.**PROPOSED METHOD**The overall goal is to distinguish samples (rotten faces) of the real and fake classes. The two models we have for training against any of our staff are (i) end -to-end and (ii) Siamese. Other evaluation tactics were also used, such as the DFDC contest methodology.

1) End-to-end training: The network presents us with a face. Once a sample face is entered into the y-related score yˆ, y. Note that no Sigmoid activation function has been applied to this score. Weight updates take place using the well-known log loss formula, which is L = − 1 N[y log (S(yˆ)) + (1 − y) log (1 − S(yˆ))] yi ∈ { 0, 1 } means the corresponding face label in where.

2) Training in Siamese: We train with the loss function triplet margin loss, which was first discussed in[35] and is motivated from computer vision research that operate CNNs to produce local feature descriptors. The non-linear dimension of f (I) racksen coding from an input face the network gives I, as indicated in Figure 2 further means the L2 norm, LT = triplet margin loss reformulated as max (0, mean + δ+ − δ−). 3) sThe losses δ- = f (Ia) – f (In) ¹ 2 and δ+ = f (Ia) −

Strictly positive f (Ip) μ 2ours is now the following Ia, Ip, and In: Ip belongs to the same group of positive samples. as Ia.

Fig.2. Effect of the attention on faces under analysis. Given some faces to analyze (top row), the attention network tends to select regions like eyes, mouth and nose (bottom row).Faces have been taken out of the FF++ dataset.

IV.EXPERIMENTS

We provide all the information about the experimental setup and datasets utilized in this section.

4.1 Dataset

FF++ [17] and DFDC [18] are the two datasets on which we test the suggested approach. Each technique is used on 1000 high-quality prism videos that were manually chosen to show topics that are almost front facing and free of occlusions after being downloaded from YouTube. There are at least 280 frames in each sequence. A constant rate quantization value of 23 and 40, respectively, is used to create high-quality videos. There are at least 280 frames in each sequence. A constant rate quantization value of 23 and 40, respectively, is used to create high-quality videos.

The DFDC is an initial dataset that was made available for the similar Kaggle competition. These particular video clips were created using both real and fake copies of over 19,000 videos. The actors in actual videos are framed against randomly chosen backdrop to create visual diversity and variability in a number of parameters (gender, skin colour, age, etc.). Some of the videos are authentic, while others are erroneous and all are created utilizing Deepfake techniques. We won't be able to determine

the precise algorithms that were used to create the fake videos because the public and private evaluation sequences, as well as an example of how they were prepared, have not yet been made public.

4.2 Networks

We take into account the following networks in our experiments:

• captioned, as it is the model that performed the best in [17], making it the ideal benchmark for our testing campaign;

• EffectiveNetB4, which outperforms other current techniques in terms of accuracy and efficiency [19];

• EffectiveNetB4Att, which ought to separate pertinent from irrelevant facial sample components. Every model is independently trained and assessed among the two sets of data being reviewed. For FF++, in particular, we evaluate only films that have been quantized with constant rate of 23. The two Efficient Net models are both trained by the assistant two approaches noted in Section III-B, Capturing is trained with the same style as in}{17. This is for our four trained models are EfficientNetB4ST and

EfficientNetB4AttST models trained with the NiceNetB4 and Siamese stra**IJCRT2502344 International Journal of Creative Research Thoughts (IJCRT)**tegy and Train **www.ijcrt.org** ed **c**with the **912** Using this straightforward attention-grabbing approach. Inversely, the flat network does not help the network areas having little gradient data. Study after studies have shown that face most of the proprietary traits create the artifacts generated by deepfake generation techniques [16]. Let's combine this concept with blockchain technology. Aside from the main components of these techniques, the major traits sketchy eyes and too many fragilely done teeth white spots.

5.2 Characteristics of Siamese:

To determine whether the features generated by the network's encoding are discriminative for the task, we used the well-known tSNE [39] technique to calculate a projection over a restricted area during the Siamese fashion training of the network. Starting at 20 FF++, Figure 5 displays the projection based on EfficientNetB4Att. Naturally, frames from the same videos cluster into little subregions. Additionally, the chart is set up with the real samples at the top and the phony samples at the bottom. Then, by clustering its frames, the same video is segmented into smaller subregions.

5.3 The Independence of architecture:

For resolution of networks it can be in an ensemble ,where independent models can record the scores. In Figure 6, each plot below the diagonal highlights how several networks offer distinct scores for each frame. In practice, the point clouds are not sufficiently coordinated in a shape that can be represented by a simple join. This urges us use all of the learned models at once.

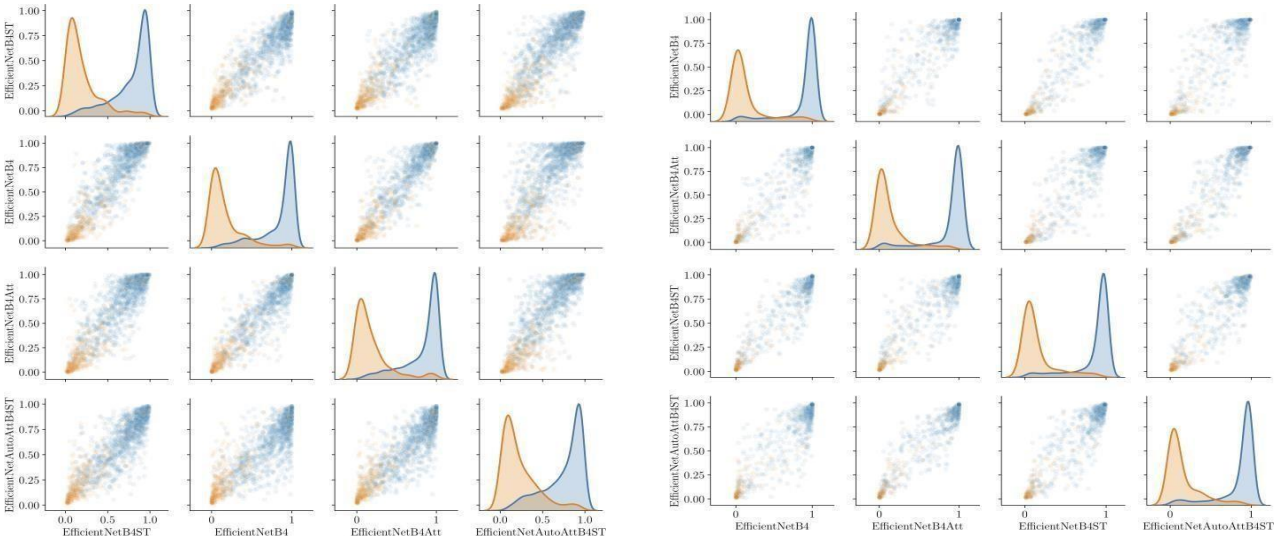
5.4 The Ability to identify face manipulation:

The average results for the baseline network (captioned) and the four proposed models (EfficientNetB4AttST) are shown in this section. There is a theory consisting of an ensemble that has one, two or some even numbers of models. In this particular instance, summing the scores produced by each distinct model yields the greatest score linked to a face. Table I shows the log loss ob.-tainted and AUC (the decisions made by binarizing the network output with various thresholds) for our trials. The out comes are shown for every frame .It is crucial to keep in mind that the model-assembling method usually produces positive outcomes when analysing these findings.

5.5 Results from Kaggle

Team of ISPL, Participated in the DFDC challenge on Kaggle [18] to have a better understanding of the performance of the suggested solution. The final objective of the competition was to develop a system that could distinguish between a real and a false video.

The training dataset made available by the competition host is the DFDC dataset utilized in this work, whereas the two distinct testing datasets are employed for evaluation: (i) the public test.



This segment introduces the tactic we crafted for identifying if a face in a video shot is authentic ('pristine') or a fabrication. At the heart of our suggested tactic lies the concept of enemy-bling. For quite a while now, folks have realized combining models can lead to better prediction accuracy. With that insight, we are zeroing in on the question of whether we can teach a bunch of CNN models to catch different kinds of high-level semantic details that fill in for each other pretty well. To achieve this, the Efficient Net lineage unveiled in [19] as a bold new strategy to scaling CNNs , is our starting point. This group of architectural surpasses other cutting-edge CNNs in accuracy and efficiency and has been shown to be highly helpful in meeting the time and hardware requirements set by DFDC.

We provide two approaches to make the model useful for the enabling given an Efficient Net design. As an alternative we still propose an introduction of an attention mechanism which would benefit the analyst in observing at what video segment is more informative towards the task of categorization. To gather max details about the data, we must figure out how to add Siamese training ways into the learning method. Below, you'll find more on the Efficient-Net structure, the suggested focus feature, and the way to train the network.

3.1 Effective attention and net mechanism

In one study denoted as [19], this star player got a score of 83.8% for nailing the top spot in identifying pictures on the ImageNet [30] challenge, and it did all this with 19 million bits and pieces and used up 4.2 billion FLOPS. Now, if we take a look at another piece of work tagged as [17], they used the same challenge for a method named Captioned, which managed 79% on hitting the top spot but gulped down twice as many FLOPS at 8.4 billion and had more bits to it with 23 million parameters. If you want to catch a glimpse of what EfficientNetB4's bones look like just peek at the blue area in Figure 2. . There you'll see all its parts Future research will examine how to improve the model selection criteria even further, incorporate multimodal capabilities, and optimize the architecture for conversational tasks that are even more complex. The spread of deepfake photos and videos has serious ramifications for digital media trust, privacy, and national security. Using convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), recent developments in deep learning-based techniques have demonstrated impressive performance in identifying deepfakes. However, the generation of diverse, high-quality datasets and the development of strong detection models that can resist adversarial attacks continue to be pressing research areas. Moreover, real-world deployment and the advancement of a more secure and reliable digital environment depend on explainability, interpretability, and ongoing model changes.

VII.REFERENCES

[1] M. Zoller, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Prez, M. Stamm Inger, M. Niner, and C. Theobald, "State of the art on monocular 3d face reconstruction, tracking, and applications," *Computer Graphics Forum*, vol. 37, pp. 523–550, 2018.

[2] J. Thies, M. Zoll Hofer, M. Stamm Inger, C. Theobald, and M. Neuner, "Face2face: Real-time face capture and reenactment of grub videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.

[3] J. Thies, M. Solnhofen, and M. Neuner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[4] "Deepfakes GitHub, https://github.com/deepfakes/faceswap.

[5]"Faceswap," https://github.com/MarekKowalski/FaceSwap/.

[6] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, "Protecting world leaders against deep fakes," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[7] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys*, vol. 43, no. 26, pp. 1–42, 2011.

[8] S. Milani, M. Fontana, P. Betaine, M. Barni, A. Piva, M. Pagliacci, and S. Tufaro, "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, p. e2, 2012.

[9] M. C. Stamm, Min Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[10] P. Betaine, S. Milani, M. Tallahatchie, and S. Tufaro, "Codec and gop identification in double compressed videos," *IEEE Transactions on Image Processing (TIP)*, vol. 25, pp. 2298–2310, 2016.

[11] D. Quezada, M. Fontane, D. Shalane, F. Prez-Gonzlez, A. Piva, and M. Barni, "Video integrity verification and gap size estimation via generalized variation of prediction footprint," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 15, pp. 1815–1830,2020.

[12] P. Betaine, S. Milani, M. tagasaste, and S. tuber, "Local tamper- in detection in video sequences," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.

[13] L. Damianos, D. Cozzolino, G. Poggi, and L. Verdolaga, "A patch match based dense-field algorithm for video copy move detection and localized in," *IEEE Transactions on Circuits and Systems for Video Technology(TCSVT)*, vol. 29, pp. 669–682, 2019.

[14] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 1315– 1329, 2012.

[15] A. Gerona, M. Fontanne, T. Bianchi, A. Piva, and M. Barni, "A video forensic technique for detecting frame deletion and insertion," in *2014IEEE International Conference on Acoustics, Speech and Signal Procasing (ICASSP)*, 2014, pp. 6226–6230.

[16] L. Verdolaga, "Media forensics and deepfakes: an overview," 2020.

[17] A. Roesler, D. Cozzolino, L. verdolaga, C. Riess, J. Thies, and M. Neuner, "Face Forensics++: Learning to detect manipulated facial images," in *International Conference on Computer Vision (ICCV)*, 2019.

[18] "Deepfake Detection Challenge (DFDC),"https://deepfakedetectionchallenge.ai/, 2019.

[19] M. Tan and Q. V. Le, "Efficient net: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning, (ICML) 2019*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6105–6114.

[20] A. Vaswani, N. Shabeer, N. Parmar, J. Ozokerite, L. Jones, A. N. Gomez, L. Kaiser, and I. Polishing,

"Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, I. Guyon, U. V. Lux burg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [21] D. Achaar, V. Nozick, J. Yamagishi, and I. Echizen, "mesonota: a compact facial video forgery detection network," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[22] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition assessment and detection," *Corr*, vol. abs/1812.08685, 2018.

[23] D. Guvera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2019.

[24] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *IEEE Conference on Computer Vision and Pattern Recognitions Workshops (CVPRW)*, 2019.

[25] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *IEEE International Conference on Acoustics, Speech and Processing (ICASSP)*, 2019.

[26] F. Matern, C. Riess, and M. Steiniger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019.

[27] Y. Li, M. Chang, and S. Lyu, "In cite oculi: Exposing AI created fake videos by detecting eye blinking," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018.

[28] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," *Corri*, vol. abs/1906.06876, 2019. [29] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain, "On the detection of digital face manipulation," 2019. [30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[31] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[32] V. Bassarisks, Y. Kartune, A. Vanunu, K. Raveendran, and M. Grundmann, "Blueface: Sub-millisecond neural face detection on mobile opus," *Corr*, vol. abs/1907.05047, 2019. [Online].

Available :http://arxiv.org/abs/1907.05047

[33] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, 2018, pp.7132–7141.

[35] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning finegrained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[36] E. K. V. I. I. A. Busload, A. Barinov and A. A. Kalinin, "Augmentations: fast and flexible image augmentations," *Arrive e-prints*, 2018.

[37] A. Paczki, S . Gross, F . Massa, A . Lerer, J . Bradbury, G. Chanan, T . Killeen, Z . Lin, N . Gimel Shein, L . Antigo, A. Desma Ison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chellamuthu, B. Steiner,

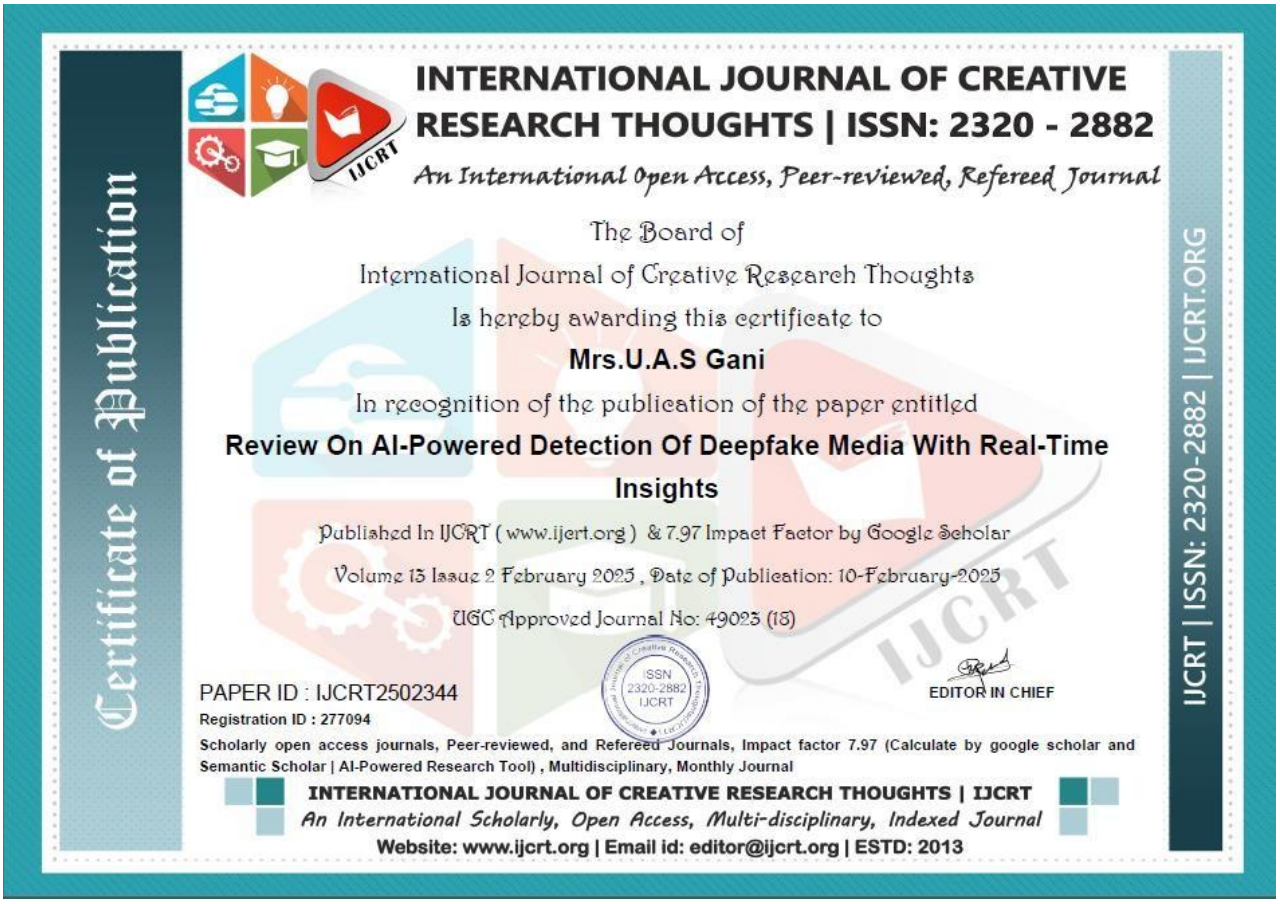
L. Fang, J. Bai, and S. Chintala, "Porch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Bergelmir, F. d'Alene´-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019,pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/

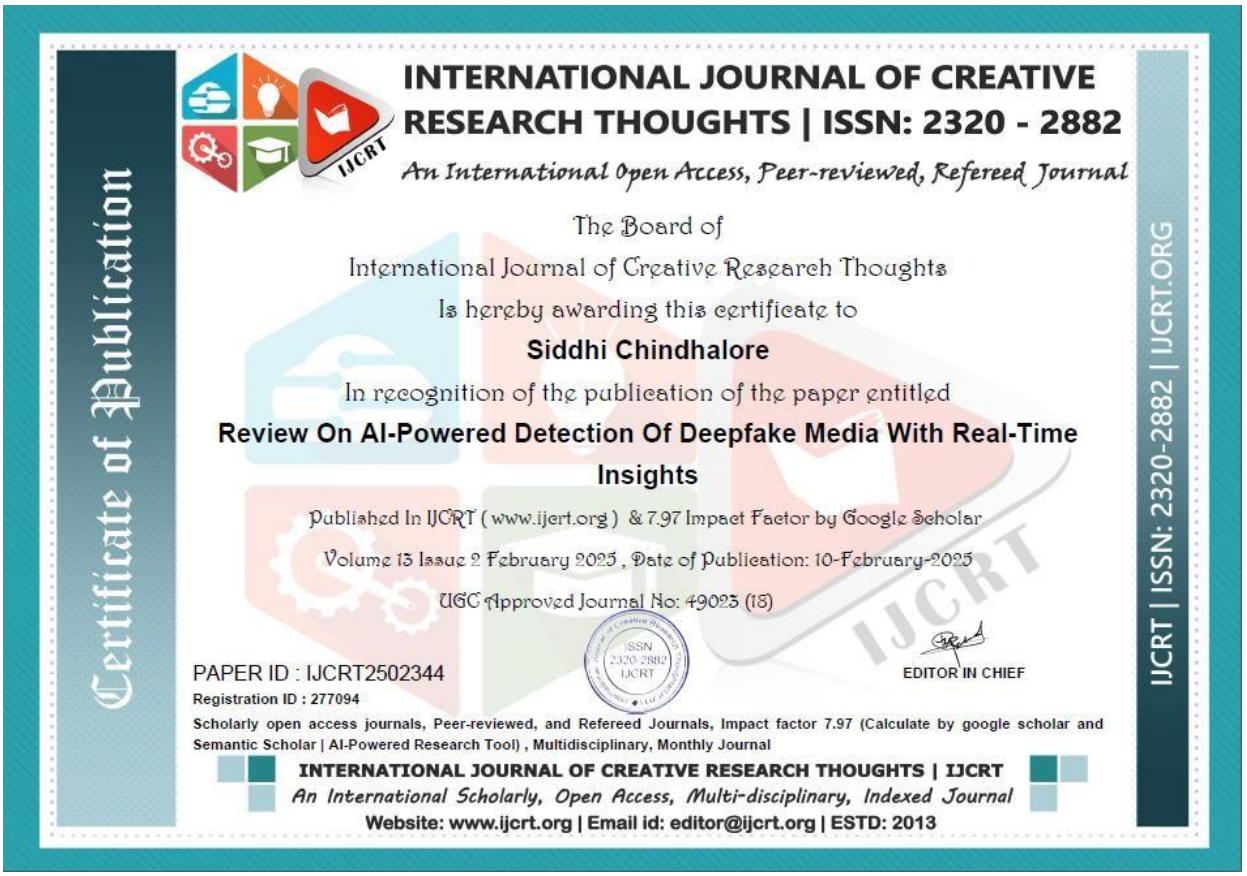
9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf [38] D. Kingma and J. Ba, "Adam: a method for stochastic optimization. arrive: 14126980," 2014.

[39] L. van der Maten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9,pp.2579–

2605,2008. [Online].Available:http://www.jmlr.org/papers/v9/vandermaaten08a. html

CERTIFICATES





Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Siddhi Chindhalore

In recognition of the publication of the paper entitled

Review On AI-Powered Detection Of Deepfake Media With Real-Time Insights

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 13 Issue 2 February 2025 , Date of Publication: 10-February-2025

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2502344

Registration ID : 277094

EDITOR IN CHIEF

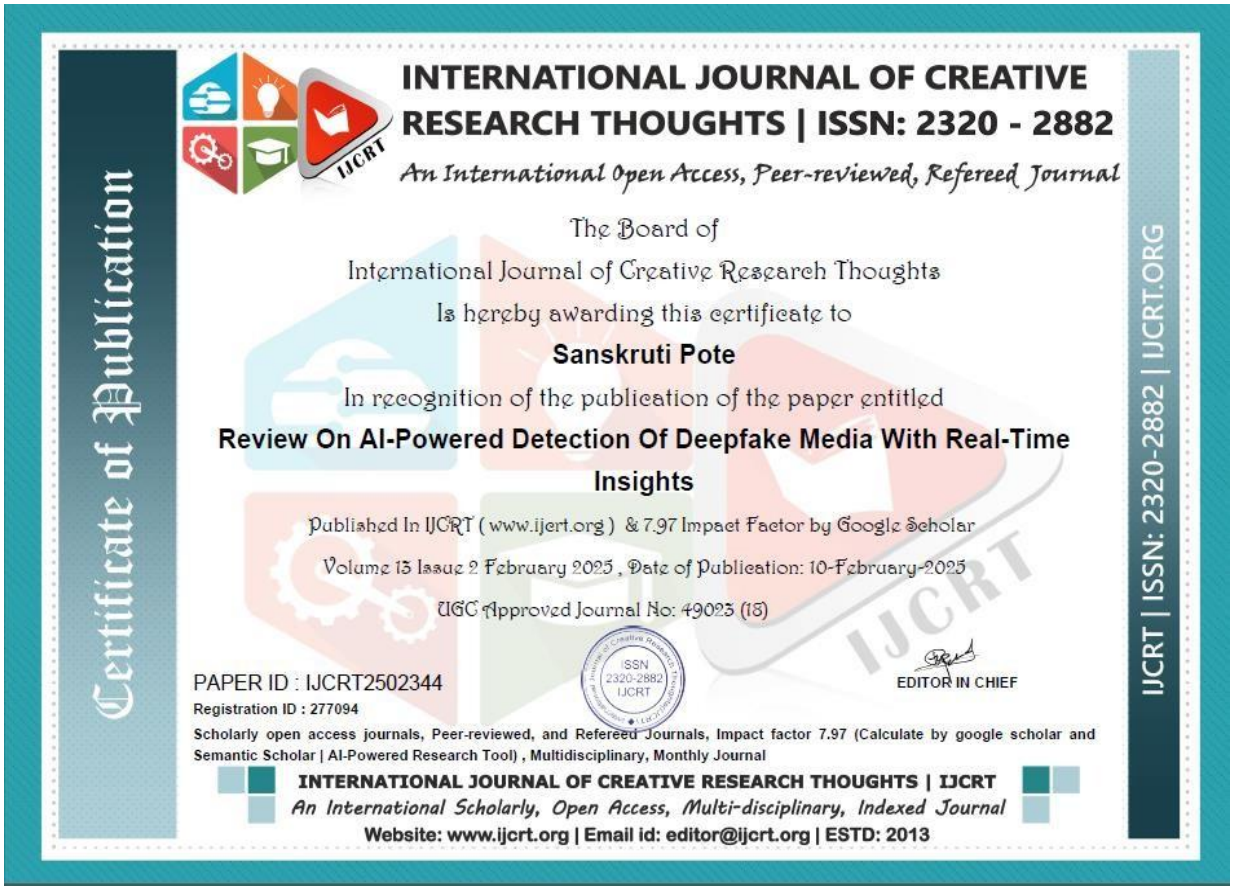
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Sanskruti Pote

In recognition of the publication of the paper entitled

Review On AI-Powered Detection Of Deepfake Media With Real-Time Insights

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 13 Issue 2 February 2025 , Date of Publication: 10-February-2025

UGC Approved Journal No: 49023 (18)

PAPER ID : IJCRT2502344

Registration ID : 277094

EDITOR IN CHIEF

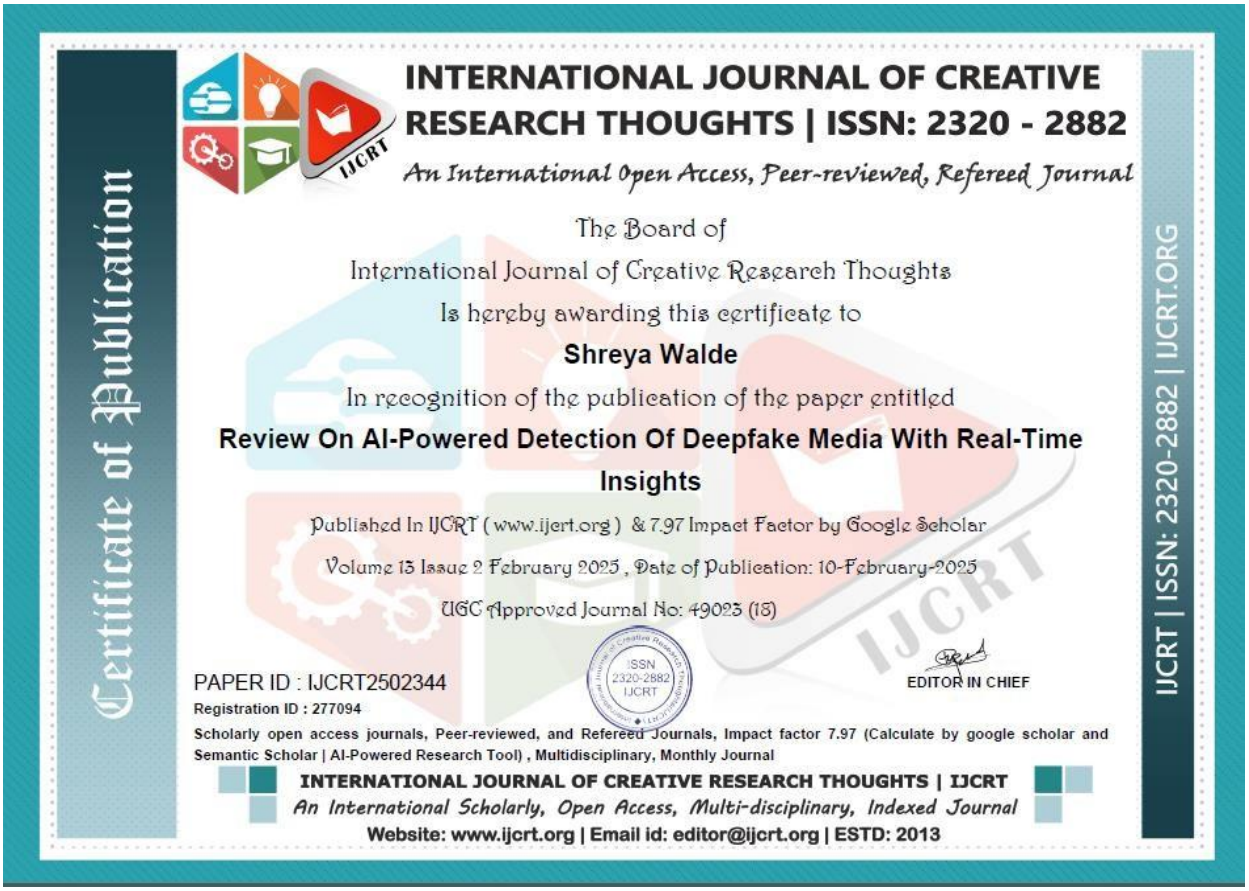
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of

International Journal of Creative Research Thoughts

Is hereby awarding this certificate to

Shreya Walde

In recognition of the publication of the paper entitled

Review On AI-Powered Detection Of Deepfake Media With Real-Time Insights

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 13 Issue 2 February 2025 , Date of Publication: 10-February-2025

UGC Approved Journal No: 49023 (13)

PAPER ID : IJCRT2502344

Registration ID : 277094

EDITOR IN CHIEF

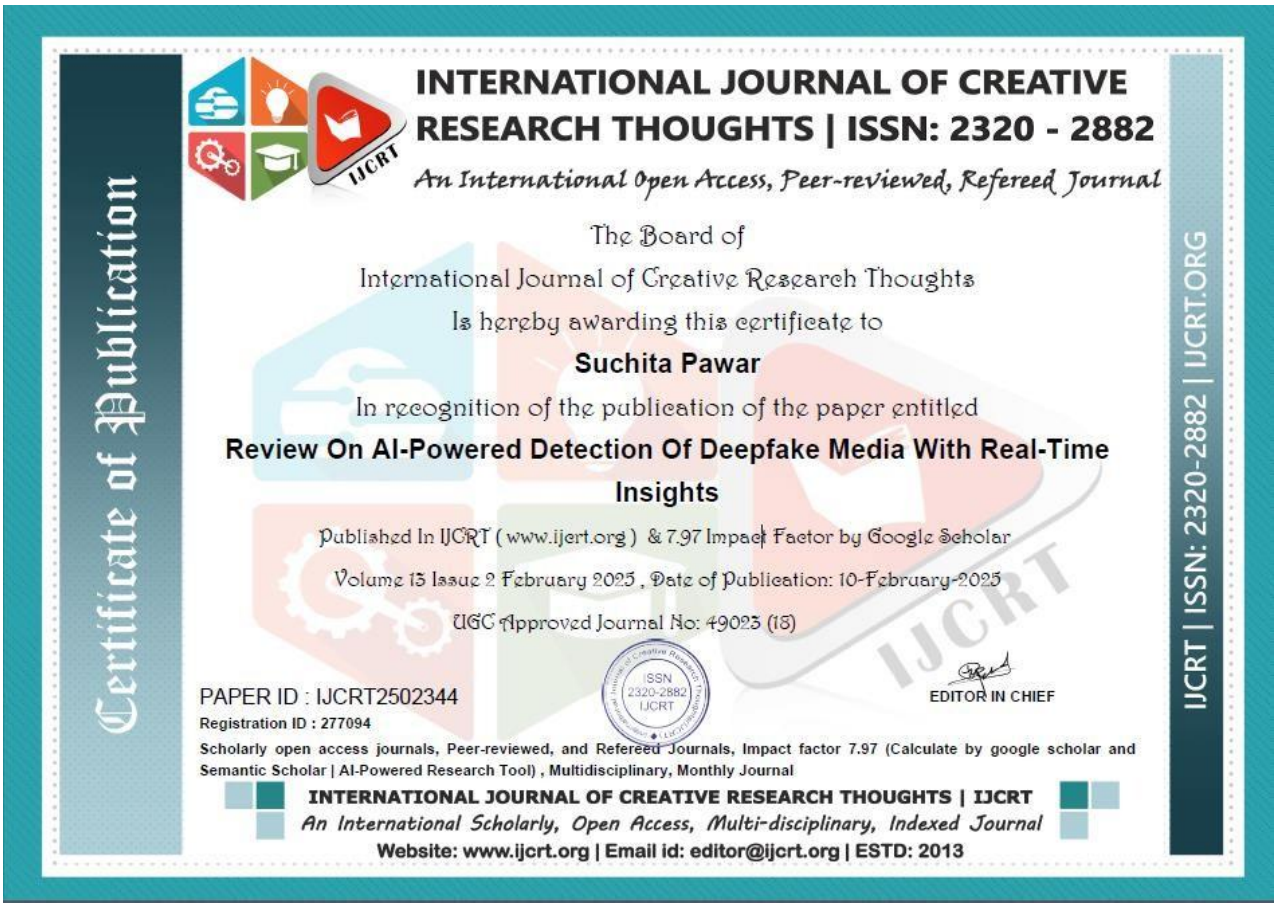
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG


Certificate of Publication
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882
An International Open Access, Peer-reviewed, Refereed Journal
The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to
Suchita Pawar
In recognition of the publication of the paper entitled
Review On AI-Powered Detection Of Deepfake Media With Real-Time Insights
Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar
Volume 13 Issue 2 February 2025 , Date of Publication: 10-February-2025
UGC Approved Journal No: 49023 (13)
PAPER ID : IJCRT2502344
Registration ID : 277094
EDITOR IN CHIEF
Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013
IJCRT | ISSN: 2320-2882 | IJCRT.ORG

Lokmanya Tilak Jankalyan Shikshan Sanstha's

Priyadarshini College of Engineering, Nagpur

(An Autonomous Institute)

Department of Aeronautical Engineering

In collaboration with

Aeronautical Society of India, Nagpur Branch

Vaimanika Aerofest-2K25

20th & 21st March 2K25

CERTIFICATE

OF PARTICIPATION

Ms. Siddhi Chindhalara

of AIDS, PCE Nagpur has

participated in National Tech-Quest Event (Technical ~~Non-Technical~~) in

Paper Presentation, organized by

the Department of Aeronautical Engineering,

Priyadarshini College of Engineering (An Autonomous Institution), Nagpur.

Dr. V. Kaushik	Dr. P. B. Khope	Dr. S. A. Dhale	Air Vice Mshl (Dr.) VRS Raju VSM
Convenor	Head & Convenor	Principal PCE Nagpur	Chairman, AeSI Nagpur Branch

Lokmanya Tilak Jankalyan Shikshan Sanstha's

Priyadarshini College of Engineering, Nagpur

(An Autonomous Institute)

Department of Aeronautical Engineering

In collaboration with

Aeronautical Society of India, Nagpur Branch

Vaimanika Aerofest-2K25

20th & 21st March 2K25

CERTIFICATE

OF PARTICIPATION

Ms. Sanskruti Pate

of AIDS, PCE Nagpur has

participated in National Tech-Quest Event (Technical ~~Non Technical~~) in

Paper Presentation, organized by

the Department of Aeronautical Engineering,

Priyadarshini College of Engineering (An Autonomous Institution), Nagpur.

Dr. V. Kaushik	Dr. P. B. Khope	Dr. S. A. Dhale	Air Vice Mshl (Dr.) VRS Raju VSM
Convenor	Head & Convenor	Principal PCE Nagpur	Chairman, AeSI Nagpur Branch

Lokmanya Tilak Jankalyan Shikshan Sanstha's

Priyadarshini College of Engineering, Nagpur

(An Autonomous Institute)

Department of Aeronautical Engineering

In collaboration with

Aeronautical Society of India, Nagpur Branch

Vaimanika Aerofest-2K25

20th & 21st March 2K25

CERTIFICATE

OF PARTICIPATION

Ms. Shreya Walde

of AIDS, PCE Nagpur has

participated in National Tech-Quest Event (Technical ~~Non Technical~~) in

Paper Presentation, organized by

the Department of Aeronautical Engineering,

Priyadarshini College of Engineering (An Autonomous Institution), Nagpur.

Dr. V. Kaushik	Dr. P. B. Khope	Dr. S. A. Dhale	Air Vice Mshl (Dr.) VRS Raju VSM
Convenor	Head & Convenor	Principal PCE Nagpur	Chairman, AeSI Nagpur Branch

Lokmanya Tilak Jankalyan Shikshan Sanstha's

Priyadarshini College of Engineering, Nagpur

(An Autonomous Institute)

Department of Aeronautical Engineering

In collaboration with

Aeronautical Society of India, Nagpur Branch

Vaimanika Aerofest-2K25

20th & 21st March 2K25

CERTIFICATE

OF PARTICIPATION

Ms. Suchita Pawar

of AIDS, PCE Nagpur has

participated in National Tech-Quest Event (Technical ~~Non-Technical~~) in

Paper Presentation, organized by

the Department of Aeronautical Engineering,

Priyadarshini College of Engineering (An Autonomous Institution), Nagpur.

Dr. V. Kaushik
Convenor

Dr. P. B. Khope
Head & Convenor

Dr. S. A. Dhale
Principal PCE
Nagpur

Air Vice Mshl (Dr.) VRS Raju VSM
Chairman, AeSI
Nagpur Branch

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

Google Scholar Crossref MENDELEY

Certificate of Publication

PROF. MRS. U.A.S.GANI

Dept. of Artificial Intelligence and Data Science,

Priyadarshini College of Engineering,Nagpur, Maharashtra

Published a paper entitled

Unveiling Deepfake Audio Detection:

A Novel Approach Using MFCCs (Mel-Frequency Cepstral Coefficients)

Volume 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14159

Certificate# IJARCCE/2025/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 Sertifikat CERTYFIKAT CERTIFICADO 證書 СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA

www.ijarcce.com

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

A monthly Peer-reviewed & Refereed journal

Impact Factor 8.102

Indexed by Google Scholar, Mendeley, Crossref, Scilit,

SCIENCEOPEN, SCIENCEGATE, DORA, KOAR

Google Scholar Crossref MENDELEY

Certificate of Publication

SHREYA GHORADKAR

Dept. of Artificial Intelligence and Data Science,

Priyadarshini College of Engineering, Nagpur, Maharashtra

Published a paper entitled

Unveiling Deepfake Audio Detection:

A Novel Approach Using MFCCs (Mel-Frequency Cepstral Coefficients)

Volume 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14159

Certificate# IJARCCE/2025/1

ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE
DOI 10.17148/IJARCCE

Editor-in-Chief
IJARCCE

SERTIFIKAAT 証明書 Sertifikat CERTYFIKAT CERTIFICADO 證書 СЕРТИФИКАТ 证书 POTVRDA SERTİFİKA

www.ijarcce.com

Project Team at a Glance:

Project Guide :
Prof. Mrs. U. A. S. Gani
AI & DS Department
Priyadarshini College of Engg, Nagpur
Mob. No. – 8010035038
Email Id- saqibayeman@gmail.com

Siddhi Chindhalore
Priyadarshini College of Engg, Nagpur
301 Atharva Enclave, Bhagyashree Nagar, Nagpur
Mob. No- 9960797530
Email Id- chindhaloresiddhi21@gmail.com

Sanskruti Pote
Priyadarshini College of Engg, Nagpur
Pioneer Woods, Wanadongri, Nagpur
Mob. No- 8080011942
Email Id- sanskrutipote16@gmail.com

Shreya Walde
Priyadarshini College of Engg, Nagpur
Plot no. 27, Police nagar, Nagpur
Mob. No- 9309336154
Email Id- shreyawalde2003@gmail.com

Suchita Pawar
Priyadarshini College of Engg, Nagpur
Surendragadh, seminary hills,, Nagpur
Mob. No- 9158274414
Email Id- suchitapawarbtl@gmail.com

Shreya Ghoradkar
Priyadarshini College of Engg, Nagpur
Vaibhav Nagar, Wanadongri, Hingna road, Nagpur
Mob. No- 9561399230
Email Id- ghoradkarshreya2003@gmail.com

Project Team with Faculty Guide
“ AI Powered Detectionof Deepfake Media withReal Time Insights”

Priyadarshini College of Engineering, Nagpur

Department of Artificial Intelligence & Data Science

Project Relevance with Project Outcomes

Final Year Project

Title of the Project: AI POWERED DETECTION OF DEEPFAKE MEDIA

WITH REAL TIME INSIGHTS

By the end of the course, the students will be able to

CO 1	Acquire a sound technical knowledge for problem identification and formulation through the prior knowledge, literature, review and original ideas.
CO 2	Use software engineering tools to analyse, design, implement, validate and maintain a project
CO 3	Develop solution to the identified problems by applying and integrating the knowledge acquired throughout his/her undergraduate study and modern techniques.
CO 4	Prepare and present a well-organised progress of a project in written and verbal form periodically.
CO 5	Work in a team and communicate with superiors, peers and the community.
CO 6	To publish and share their project works with outside world at national and international level.

Program Outcomes

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

Program Specific Outcomes

PSO 1 : To design and develop software systems with the knowledge of data structures, analysis of algorithms, web design, machine learning and Image processing techniques.

PSO 2 : To design and develop networking and embedded software solutions by means of data communication, sensors and its applications, robotics, virtual reality and Internet of Things.

PSO 3 : Apply the skills in the areas of Health Care, Education, Agriculture, Intelligent Transport, Environment, Smart Systems & in the multi-disciplinary area of Artificial Intelligence and Data Science.

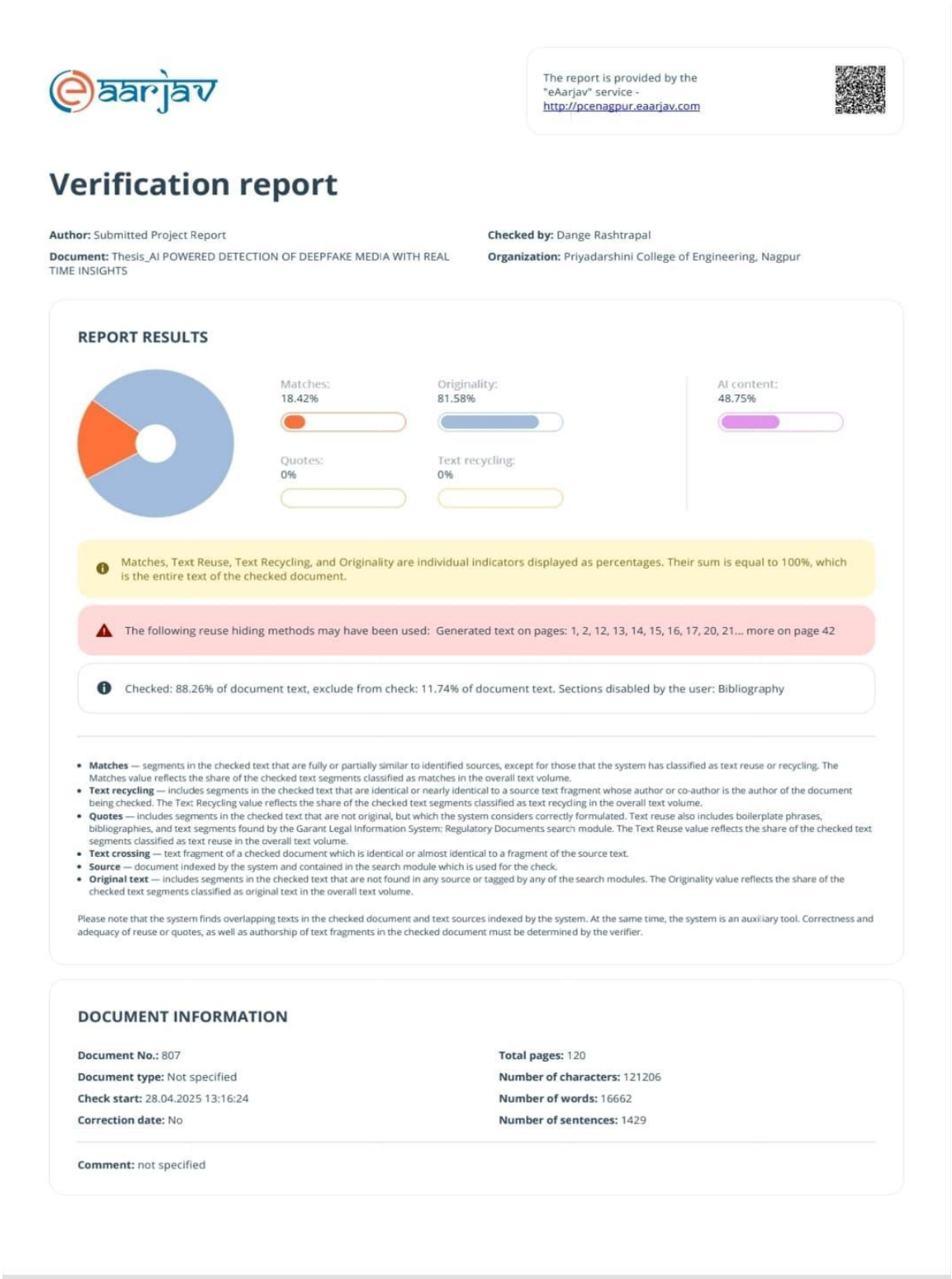
Mapping of Course Outcome with Project Outcomes:

Subject	Project Outcomes Students will be able to		ProgramOutcomes											
			1	2	3	4	5	6	7	8	9	10	11	12
Final Year	CO 1	Acquire a sound technical knowledge for problem identification and formulation through the prior knowledge, literature, review and original ideas.	3	3	2	2	1	3	1	3	1	1	1	2
	CO 2	Use software engineering tools to analyse, design, implement, validate and maintain a project	1	2	3	2	3	1	1	2	1	1	2	1
	CO 3	Develop solution to the identified problems by applying and integrating the knowledge acquired throughout his/her undergraduate study and modern techniques.	2	2	3	2	3	1	1	2	1	1	1	2
	CO 4	Prepare and present a well organised progress of a project in written and verbal form periodically.	1	1	1	1	1	1	1	1	1	3	1	1
	CO 5	Work in a team and communicate with superiors, peers and the community.	1	1	1	1	1	3	2	3	3	3	2	1
	CO 6	To publish and share their project works with outside world at national and international level.	1	1	1	1	1	2	1	2	1	3	1	3

Mapping of Course Outcomes with Program Specific Outcomes:

Subject	Project Outcomes Students will be able to		PSO's		
			1	**2**	**3**
Final Year	**CO 1**	Acquire a sound technical knowledge for problem identification and formulation through the prior knowledge, literature, review and original ideas.	3	1	3
	CO 2	Use software engineering tools to analyse, design, implement, validate and maintain a project	3	1	3
	CO 3	Develop solution to the identified problems by applying and integrating the knowledge acquired throughout his/her under graduate study and modern techniques.	3	1	3
	CO 4	Prepare and present a well-organised progress of a project in written and verbal form periodically.	1	1	2
	CO 5	Work in a team and communicate with superiors, peers and the community.	1	1	3
	CO 6	To publish and share their project works with outside world at national and international level.	1	1	3

plagiarism:



The report is provided by the "eAarjav" service - http://pcenagpur.eaarjav.com

Verification report

Author: Submitted Project Report
Document: Thesis_AI POWERED DETECTION OF DEEPFAKE MEDIA WITH REAL TIME INSIGHTS

Checked by: Dange Rashtrapal
Organization: Priyadarshini College of Engineering, Nagpur

REPORT RESULTS

Matches: 18.42%
Originality: 81.58%
AI content: 48.75%
Quotes: 0%
Text recycling: 0%

Matches, Text Reuse, Text Recycling, and Originality are individual indicators displayed as percentages. Their sum is equal to 100%, which is the entire text of the checked document.

The following reuse hiding methods may have been used: Generated text on pages: 1, 2, 12, 13, 14, 15, 16, 17, 20, 21... more on page 42

Checked: 88.26% of document text, exclude from check: 11.74% of document text. Sections disabled by the user: Bibliography

- **Matches** — segments in the checked text that are fully or partially similar to identified sources, except for those that the system has classified as text reuse or recycling. The Matches value reflects the share of the checked text segments classified as matches in the overall text volume.
- **Text recycling** — includes segments in the checked text that are identical or nearly identical to a source text fragment whose author or co-author is the author of the document being checked. The Text Recycling value reflects the share of the checked text segments classified as text recycling in the overall text volume.
- **Quotes** — includes segments in the checked text that are not original, but which the system considers correctly formulated. Text reuse also includes boilerplate phrases, bibliographies, and text segments found by the Garant Legal Information System: Regulatory Documents search module. The Text Reuse value reflects the share of the checked text segments classified as text reuse in the overall text volume.
- **Text crossing** — text fragment of a checked document which is identical or almost identical to a fragment of the source text.
- **Source** — document indexed by the system and contained in the search module which is used for the check.
- **Original text** — includes segments in the checked text that are not found in any source or tagged by any of the search modules. The Originality value reflects the share of the checked text segments classified as original text in the overall text volume.

Please note that the system finds overlapping texts in the checked document and text sources indexed by the system. At the same time, the system is an auxiliary tool. Correctness and adequacy of reuse or quotes, as well as authorship of text fragments in the checked document must be determined by the verifier.

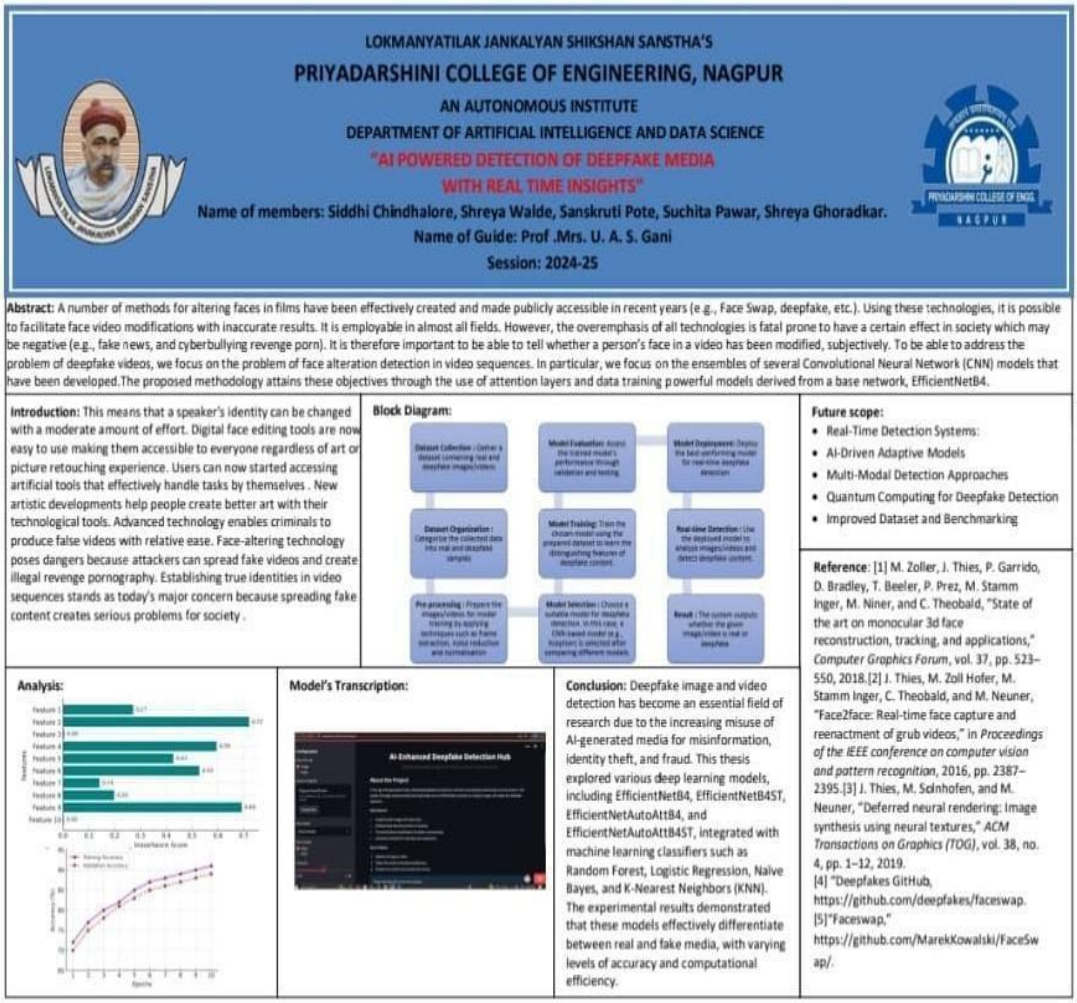
DOCUMENT INFORMATION

Document No.: 807
Document type: Not specified
Check start: 28.04.2025 13:16:24
Correction date: No

Total pages: 120
Number of characters: 121206
Number of words: 16662
Number of sentences: 1429

Comment: not specified

Poster:



LOKMANYATILAK JANKALYAN SHIKSHAN SANSTHA'S

PRIYADARSHINI COLLEGE OF ENGINEERING, NAGPUR

AN AUTONOMOUS INSTITUTE

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

"AI POWERED DETECTION OF DEEPFAKE MEDIA WITH REAL TIME INSIGHTS"

Name of members: Siddhi Chindhalore, Shreya Walde, Sanskruti Pote, Suchita Pawar, Shreya Ghoradkar.

Name of Guide: Prof .Mrs. U. A. S. Gani

Session: 2024-25

Abstract: A number of methods for altering faces in films have been effectively created and made publicly accessible in recent years (e.g., Face Swap, deepfake, etc.). Using these technologies, it is possible to facilitate face video modifications with inaccurate results. It is employable in almost all fields. However, the overemphasis of all technologies is fatal prone to have a certain effect in society which may be negative (e.g., fake news, and cyberbullying revenge porn). It is therefore important to be able to tell whether a person's face in a video has been modified, subjectively. To be able to address the problem of deepfake videos, we focus on the problem of face alteration detection in video sequences. In particular, we focus on the ensembles of several Convolutional Neural Network (CNN) models that have been developed.The proposed methodology attains these objectives through the use of attention layers and data training powerful models derived from a base network, EfficientNetB4.

Introduction: This means that a speaker's identity can be changed with a moderate amount of effort. Digital face editing tools are now easy to use making them accessible to everyone regardless of art or picture retouching experience. Users can now started accessing artificial tools that effectively handle tasks by themselves . New artistic developments help people create better art with their technological tools. Advanced technology enables criminals to produce false videos with relative ease. Face-altering technology poses dangers because attackers can spread fake videos and create illegal revenge pornography. Establishing true identities in video sequences stands as today's major concern because spreading fake content creates serious problems for society .

Block Diagram:

Future scope:

- Real-Time Detection Systems:
- AI-Driven Adaptive Models
- Multi-Modal Detection Approaches
- Quantum Computing for Deepfake Detection
- Improved Dataset and Benchmarking

Reference: [1] M. Zoller, J. Thies, P. Garrido, D. Bradley, T. Beeler, P. Prez, M. Stamm Inger, M. Niner, and C. Theobald, "State of the art on monocular 3d face reconstruction, tracking, and applications," *Computer Graphics Forum*, vol. 37, pp. 523–550, 2018.[2] J. Thies, M. Zoll Hofer, M. Stamm Inger, C. Theobald, and M. Neuner, "Face2face: Real-time face capture and reenactment of grub videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2387–2395.[3] J. Thies, M. Solnhofen, and M. Neuner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.
[4] "Deepfakes GitHub, https://github.com/deepfakes/faceswap.
[5]"Faceswap," https://github.com/MarekKowalski/FaceSwap/.

Analysis:

Model's Transcription:

Conclusion: Deepfake image and video detection has become an essential field of research due to the increasing misuse of AI-generated media for misinformation, identity theft, and fraud. This thesis explored various deep learning models, including EfficientNetB4, EfficientNetB4ST, EfficientNetAutoAttB4, and EfficientNetAutoAttB4ST, integrated with machine learning classifiers such as Random Forest, Logistic Regression, Naive Bayes, and K-Nearest Neighbors (KNN). The experimental results demonstrated that these models effectively differentiate between real and fake media, with varying levels of accuracy and computational efficiency.