BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT

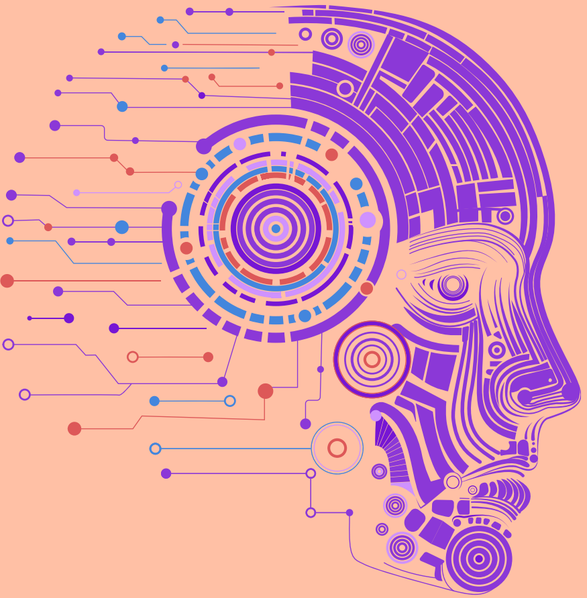
(An Autonomous Institution affiliated to VTU, Belagavi)
Yelahanka, Bengaluru-560064

Department of Computer Science and Business Systems

INNOVATION AND DESIGN THINKING - 24IDT18

Presentation
On

"VeriFake"



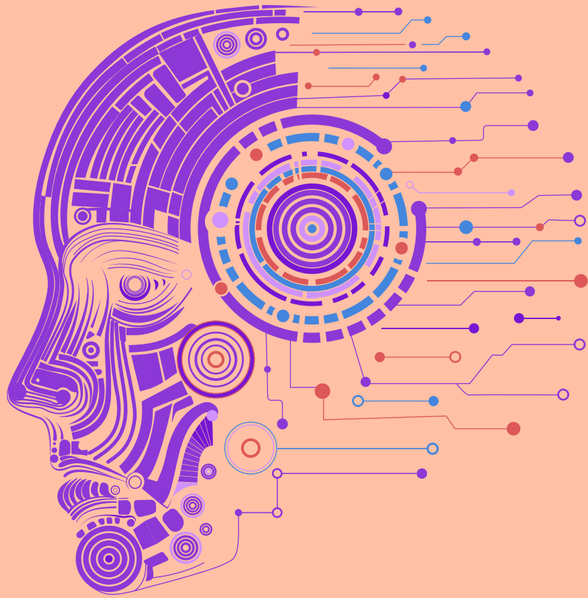
Student name:

By

Ayush Kumar
Harsh Jha
Kshitiz Khandelwal
Sameer Barala

Roll No:

24UG1BYBS023-T
24UG1BYBS024-T
24UG1BYBS009-T
24UG1BYBS026-T



Under the Guidance of

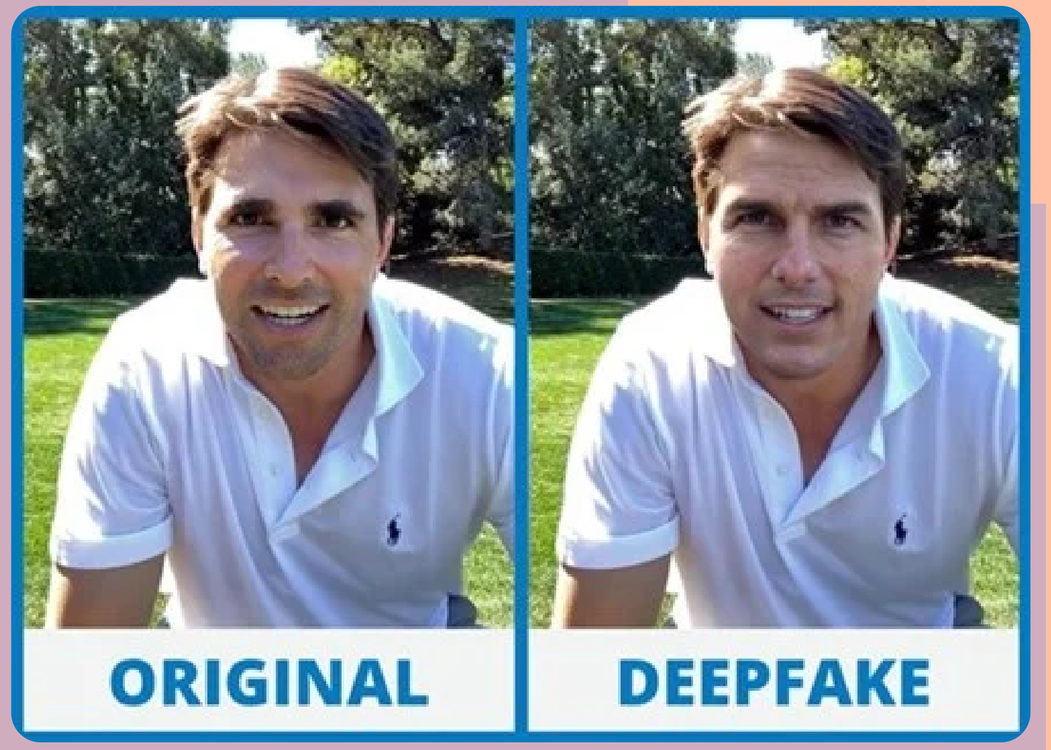
Guides/Course Coordinators	HOD
Udayaprasad P K Assistant Professors	Dr. Vishwakiran S Associate Professor

Contents

Abstract

Deepfake technology, powered by AI, enables the creation of highly realistic yet manipulated videos and images, posing significant threats to societal trust, privacy, and security. VeriFake is a deepfake detection system designed to identify these manipulations by analyzing inconsistencies in media using advanced machine learning models. The project aims to provide a scalable, efficient, and user-friendly solution to ensure media authenticity and combat the misuse of AI-generated content.

Introduction

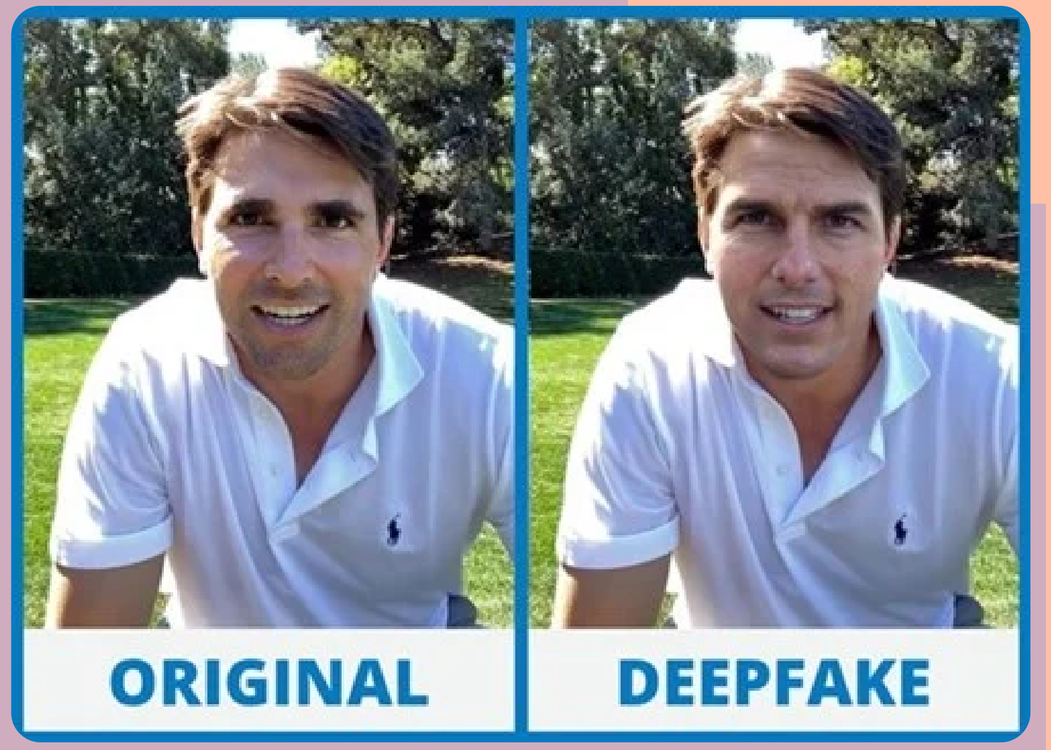


Deepfakes refer to hyper-realistic synthetic media (videos, images, and audio) generated using **Artificial Intelligence (AI) and Deep Learning**. They leverage **Generative Adversarial Networks (GANs)** and **Autoencoders** to manipulate visual and auditory content, making it almost indistinguishable from real media.

While deepfake technology has positive applications in film production, accessibility tools, and virtual reality, its misuse poses major threats, including:

- **Identity Theft** – Impersonating individuals for fraud or manipulation.
- **Political Disinformation** – Fabricating speeches or actions of political figures.
- **Financial Fraud** – Using synthetic voices or deepfake videos in scams.
- **Cybercrime & Reputation Damage** – Generating non-consensual deepfake content.

Introduction



Why *VeriFake?*

As deepfake sophistication increases, traditional detection methods fail. VeriFake utilizes AI-driven **forensic analysis, real-time CNN detection,** and **blockchain**-based content verification to distinguish between real and manipulated content. By analyzing **facial expressions, texture mismatches,** and **voice irregularities**, VeriFake ensures digital media integrity.

How Deepfakes Are Created?

Deepfake Creation Process & Technologies

Deepfake generation relies on **AI-powered frameworks** that manipulate images, videos, and voices to create hyper-realistic media. Below are the key steps and technologies involved:

Step 1: Data Collection

- Large datasets of images, videos, and voice recordings are gathered for training.
- Examples: CelebA dataset, VGGFace2, VoxCeleb (for voice synthesis).

Step 2: Face Extraction & Alignment

- Facial landmark detection algorithms identify key facial features.
- Technologies Used: OpenCV, Dlib, MediaPipe Face Mesh.

How Deepfakes Are Created?

Step 3: Deep Learning Model Training

- Uses Generative Adversarial Networks (GANs), consisting of:
  - Generator: Creates synthetic content.
  - Discriminator: Evaluates and improves the fake content.
- Technologies Used: TensorFlow, PyTorch, DeepFaceLab, StyleGAN, First-Order Motion Model.

Step 4: Post-Processing & Refinement

- AI models enhance lighting, facial blending, and motion consistency.
- Tools: DeepFaceLab, Adobe After Effects, First-Order Motion Model.

Step 5: Real-Time Rendering (For Live Deepfakes)

- Uses deep learning-based facial reenactment for real-time applications.
- Technologies Used: NVIDIA DeepStream, DeepFaceLive, FaceSwap.

Motivation

Rising Threats:

- Deepfakes undermine trust in digital media, making videos and images unreliable as evidence.
- They are used in fraudulent schemes, including blackmail, fake political propaganda, and corporate scams.

Challenges with Current Systems:

- Current systems struggle with low-resolution videos and compressed deepfakes.
- High computational costs make deepfake detection inaccessible to general users.
- Many AI models are not robust enough to detect newer deepfake generation techniques (GANs, StyleGAN3).

Motivation

Vision of VeriFake:

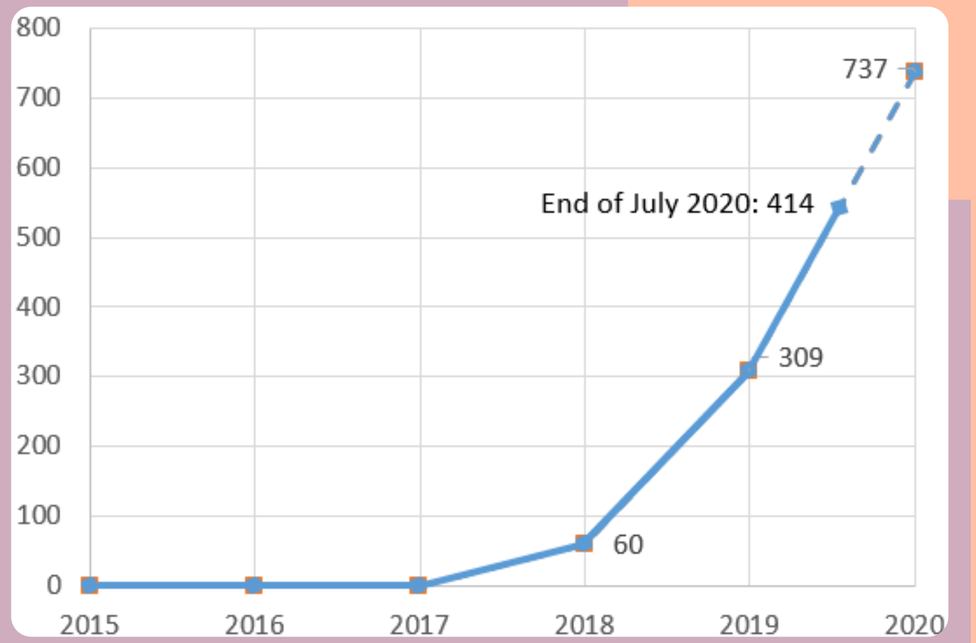
✓ Uses AI-powered forensic tools (CNNs, RNNs, and blockchain verification) to ensure accuracy.

✓ Detects synthetic facial anomalies, unnatural lip-syncing, and voice manipulations.

✓ Provides real-time & batch processing, making it useful for journalists, law enforcement, and social media platforms.

✓ Helps in educating users on deepfake awareness and countering misinformation.

Literature Survey



Existing Approaches:

- Convolutional Neural Networks (CNNs): Analyze facial features for inconsistencies.
- Temporal Analysis: Detect unnatural movements or mismatches in consecutive frames (e.g., blinking irregularities).
- Audio-Visual Discrepancies: Analyze mismatched lip movements and audio.

Challenges Identified:

- High computational costs for real-time detection.
- Limited datasets available for diverse deepfake styles.
- Difficulty in detecting high-quality deepfakes generated by advanced GANs.

VeriFake improves upon these methods by integrating advanced preprocessing and feature extraction for greater accuracy.

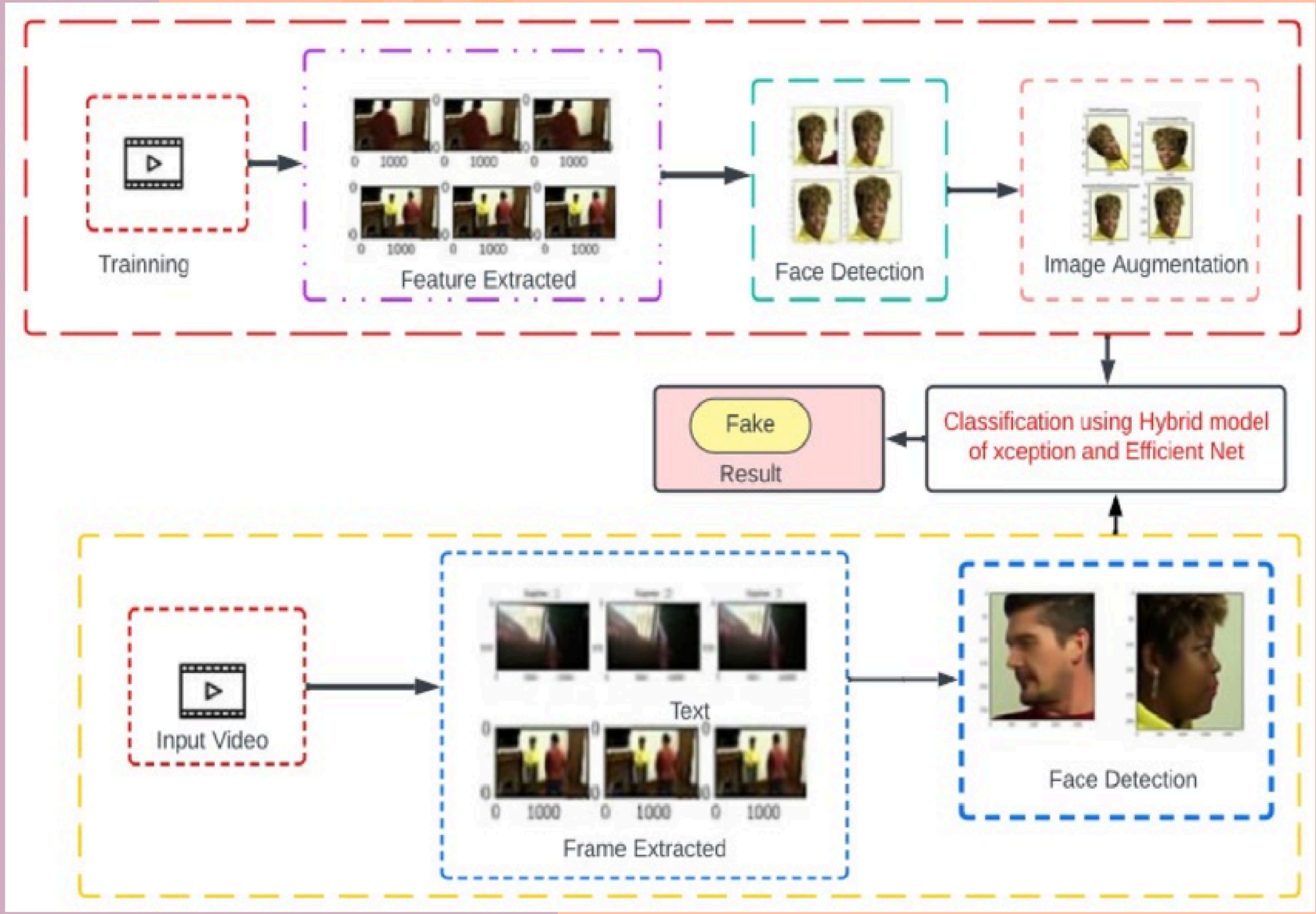
Problem Identification and Problem Statement

Problem Identification:

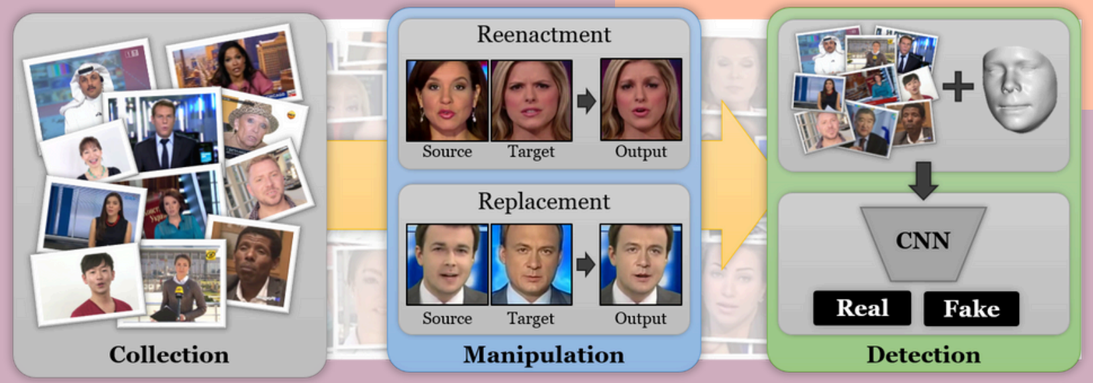
- With advancements in GANs, deepfake quality continues to improve, making them harder to detect.
- The lack of accessible and efficient tools leads to misuse in critical areas like media, finance, and politics.
- Problem Statement:

Develop a deepfake detection system, VeriFake, capable of analyzing media for manipulated content with high accuracy, low computational overhead, and real-time usability.

Proposed System



Proposed System



System Overview: VeriFake is designed to process video and image inputs, extract key features, and use a trained deep learning model to detect deepfakes.

Steps:

1. **Input Media:** Users upload a video or image.
2. **Preprocessing:** Frames are extracted and normalized. Facial regions are detected and prepared for analysis.
3. **Feature Extraction: Identifies anomalies like unnatural lighting, pixel mismatches, and facial inconsistencies.**
4. **Deep Learning Model:** A neural network trained on a labeled dataset of real and fake media classifies inputs.
5. **Output:** A confidence score is displayed, with markers highlighting manipulated sections.

Design and Methodology

System Design:

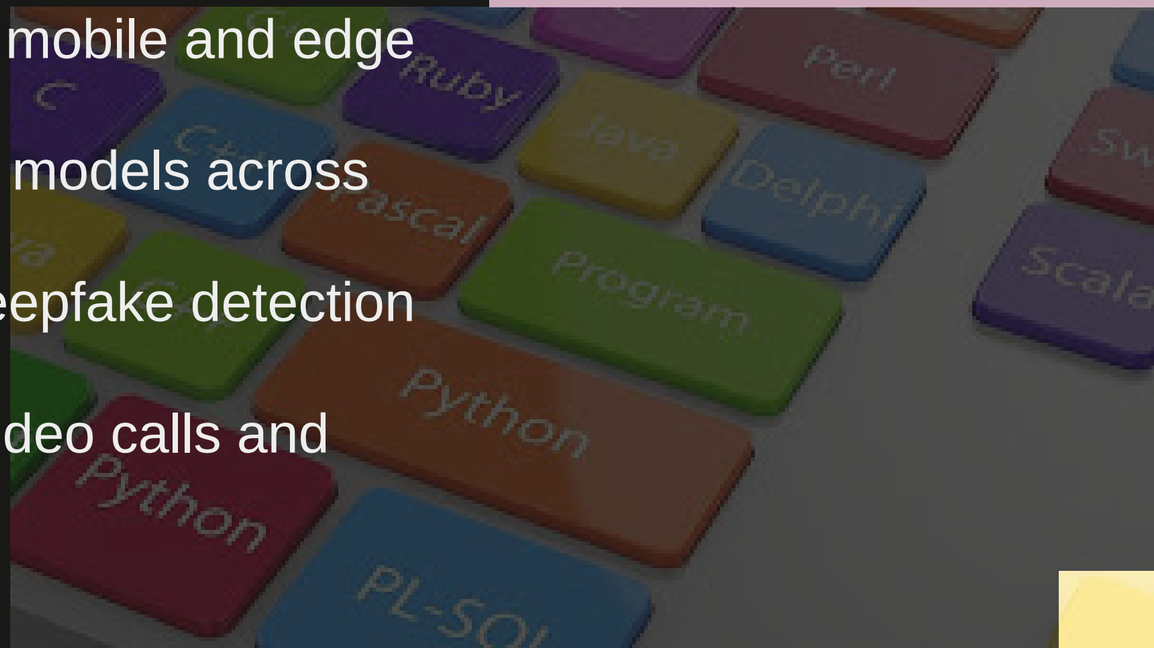
- **Input Layer:** Accepts videos or images from the user.
- **Preprocessing Module:** Extracts individual frames, detects faces, and prepares input for analysis.
- **Feature Extraction Module:** Uses machine learning to analyze inconsistencies such as:
  - Abnormal eye blinking.
  - Asymmetrical facial features.
  - Temporal discrepancies in movement.
- **Classification Module:** A deep neural network processes the features and determines whether the content is real or fake.
- **Output Layer:** Displays a confidence score and visualizes results.

Technologies Used:

- Python for implementation.
- OpenCV for image processing.
- TensorFlow/PyTorch for deep learning.

Programming Languages & Their Role in VeriFake

- **Python –** Core AI & deep learning development (TensorFlow, PyTorch).
- **JavaScript (React.js, Vue.js) –** Builds interactive dashboards for displaying detection results.
- **SQL (PostgreSQL, MySQL) –** Stores metadata, detection logs, and deepfake evidence.
- **FastAPI (Python) –** Provides backend APIs for real-time deepfake detection services.
- **TensorFlow Lite & ONNX –** Optimizes AI models for mobile and edge computing.
- **Docker & Kubernetes –** Containerizes and scales AI models across cloud servers.
- **CUDA & TensorRT (NVIDIA GPUs) –** Accelerates deepfake detection on high-performance GPUs.
- **WebRTC –** Enables real-time deepfake analysis for video calls and streaming services.

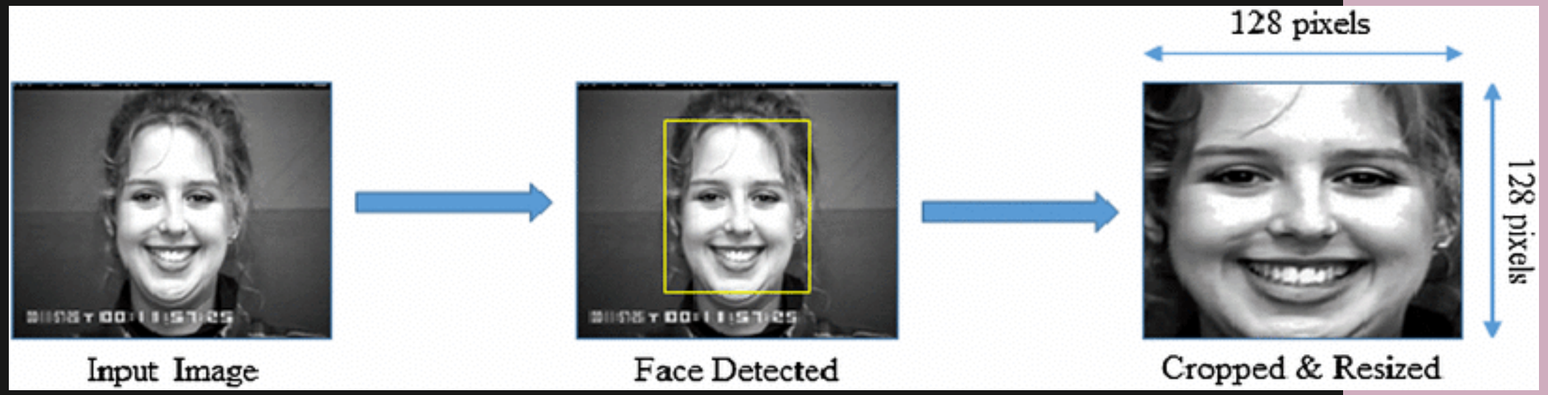


Module Split-up

1. **Data Collection:** Gather datasets like FaceForensics++, DeepFake Detection Challenge, and DFDC.
2. **Preprocessing:** Normalize media inputs, detect faces using Haar cascades or Dlib, and extract frames for analysis.
3. **Feature Extraction:** Identify anomalies such as pixel mismatches, unnatural lighting, or facial distortions.
4. **Model Training:** Use CNNs or RNNs trained on labeled datasets to classify content.
5. **User Interface:** Build a user-friendly dashboard to upload media, display detection results, and highlight manipulated areas.

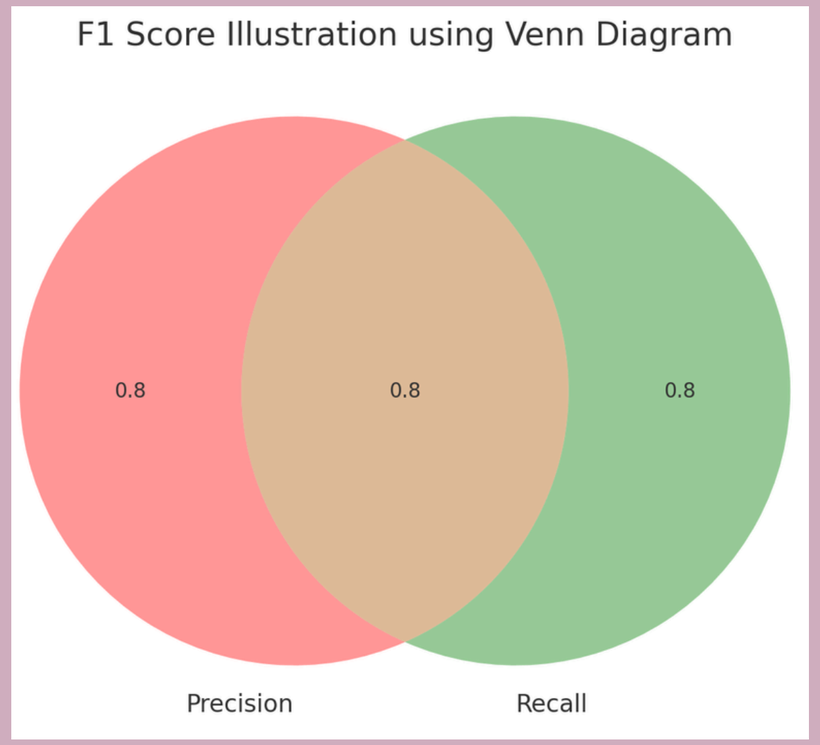
Implementation

Data Collection: Use datasets like FaceForensics++ and DFDC; augment data with noise and compression for robustness.
Preprocessing: Extract frames, detect faces with OpenCV/Dlib, and normalize them for consistency.
Feature Extraction: Analyze anomalies (e.g., unnatural blinking, pixel mismatches) using CNNs like XceptionNet.
Model Training:
Use labeled datasets (real vs. fake).
Train a neural network with binary cross-entropy loss and Adam optimizer.
User Interface: Create a web app (Flask/Django) for user uploads, displaying results with a confidence score and highlighted manipulated areas.



Tools

TensorFlow Model Analysis, PyTest, Matplotlib for visualization.



Testing

Unit Testing: Test individual modules (e.g., preprocessing or feature extraction).
Integration Testing: Check smooth interaction between modules like frame extraction and classification.
Performance Testing: Measure time taken to process videos on different hardware.
Stress Testing: Handle bulk detection for scalability.

Example Metrics:

- **Dataset:** FaceForensics++
- **Accuracy:** 94%, False Positives: 2%, False Negatives: 3%.

Application Area of Project

- **Media Authenticity:** Verify the credibility of news videos and images.
- **Social Media Platforms:** Prevent the spread of manipulated content.
- **Legal Evidence Validation:** Identify altered videos in criminal investigations.
- **Education and Awareness:** Educate the public on the dangers of deepfakes.
- **Digital Forensics:** Help law enforcement detect synthetic content in criminal investigations.
- **Corporate Security:** Safeguard executives from deepfake-based scams (e.g., fake voice calls for financial transactions).

Results and Discussions

Limitations

Performance on Low-Quality Inputs:

- Compression and noise degrade the model's accuracy.
- Detection is less reliable for older devices producing low-resolution videos.

Adversarial Attacks:Advanced GANs may bypass detection using adversarial tricks.

Dataset Bias: Models trained on limited datasets may struggle with novel manipulation techniques.

Computational Overhead: Real-time processing can be resource-intensive without optimizations like GPU acceleration.

Ethical Concerns: Detection systems might infringe on privacy during large-scale monitoring.

Results:

Tested on benchmark datasets such as FaceForensics++ and DFDC.

Achieved:

- 92% Accuracy on high-quality deepfakes.
- 88% Accuracy for low-resolution, heavily compressed media.

Model performance metrics:

- Precision: 93%
- Recall: 89%
- F1-Score: 91%

Comparative performance:

- XceptionNet-based models outperform traditional methods (e.g., handcrafted feature-based detection).

References

FaceForensics++ Dataset and Research Paper

- **Link:** https://github.com/ondyari/FaceForensics
- A large dataset for training and evaluating deepfake detection systems.

DeepFake Detection Challenge (DFDC)

- **Link:** https://www.kaggle.com/c/deepfake-detection-challenge
- Kaggle-hosted competition with datasets and baseline models for deepfake detection.

Detecting DeepFake Videos Using Recurrent Neural Networks (RNN)

- **Link:** https://ieeexplore.ieee.org/document/9054358
- A study on identifying temporal inconsistencies in deepfakes using RNNs.

DeepFake Detection Using XceptionNet

- **Link:** https://arxiv.org/abs/1905.00582
- Research paper on leveraging CNN architectures like XceptionNet for deepfake detection.

GAN-Fingerprint-Based Detection

- **Link:** https://arxiv.org/abs/1912.11035
- Discusses unique GAN fingerprints as a means of detecting synthetic media.

References

Deepfake example

- Link: https://www.youtube.com/watch?v=WzK1MBEpkJ0

Deepfake Misuse

- Link: https://www.youtube.com/watch?v=X17yrEV5sl4


THANK YOU