FrameGuard: AI-Based Authentication of Visual Media

Two-Agent Architecture for Deepfake Detection

Efficient detection of deepfakes through intelligent frame selection and specialized agent collaboration

Sampling Agent

Detection Agent

Introduction to Deepfake Detection

What are Deepfakes?

Videos or audio that have been digitally altered to change a person's appearance or voice using deep learning methods.

Increasing Realism

Deepfakes are becoming increasingly realistic and difficult to detect with the human eye.

Detection Challenges

Small visual flaws or audio issues often occur across several frames, making detection difficult.

Why Detection Systems Matter

- Combating misinformation
- Preventing fraud
- Stopping identity theft

Challenges in Deepfake Detection

Detecting deepfakes is increasingly difficult due to several technical and computational challenges:

High-Quality Outputs

Modern deepfake models produce high-quality and consistent outputs over time, making them increasingly difficult to distinguish from authentic media.

Temporal Consistency

Small visual flaws or audio issues often persist across multiple frames, requiring sophisticated analysis to detect inconsistencies over time.

Computational Demands

Analyzing large video files requires substantial computing power, which increases costs and processing time for effective detection.

Real-Time Processing

The computational requirements make it challenging to implement effective deepfake detection in real-time or at a large scale.

Limitations of Full-Frame Processing

Traditional deepfake detection systems analyze each frame of a video independently, regardless of content similarity.

Redundant Processing

Many consecutive frames contain similar information, especially in static scenes, leading to unnecessary computational work.

Computational Costs

Processing similar frames increases computing costs, making real-time deepfake detection impractical for many applications.

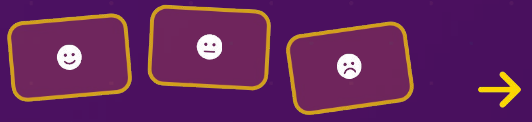
Memory Usage

Full-frame processing requires substantial memory resources to store and analyze consecutive frames unnecessarily.

Key Observation Behind the Proposed Approach

Deepfake artifacts are **more visible during motion** and scene changes.

Frames with Motion



- Facial expressions
- Lip movements
- Sudden scene changes

Static Frames

- Adds little extra information
- High redundancy with neighbors
- Not essential for detection

Adaptive Frame Sampling Concept

Dynamic Frame Selection

Adaptive frame sampling chooses frames based on their information value rather than processing all frames at a fixed rate.

Frame Selection Process

F1 F2 F3 F4 F5 F6 F7 F8

Selected frames analyzed further

Frame Differencing

Identifies changes between consecutive frames by comparing pixel differences.

Motion Detection

Captures facial movements and scene changes to identify informative frames.

Why a Two-Agent Architecture is Required

To implement adaptive sampling and deepfake detection efficiently, the system divides tasks between two specialized agents, ensuring optimal performance and resource utilization.

Agent 1: Adaptive Sampling

Lightweight task execution

- Resource-efficient processing
- Intelligent frame selection

Agent 2: Detection

Heavy inference operations

- Deepfake detection models
- Visual and audio analysis

Key Benefits of Two-Agent Approach

Enhanced Performance
Separates light from heavy tasks

Improved Scalability
Independent scaling of components

Better Organization
Clear task separation

Live Demo URL: https://frameguard.onrender.com

Deployment Notes:

- Cloud-hosted Flask backend (Render)
- Lite deployment mode enabled
- Full forensic pipeline runs locally
- Heavy ML/video processing disabled on free tier.

Technology Stack Used

Programming Language

Python – core logic, model inference, preprocessing, and the part where everything breaks if you miss a colon.

Backend / Server

Flask – lightweight web framework for handling image uploads, routing, and model responses.

Machine Learning / AI

TensorFlow / Keras – building and running the deepfake detection model.

CNNs – extracting visual features to identify synthetic patterns in images.

Image Processing

OpenCV – image resizing, normalization, and basic preprocessing.

Pillow (PIL) – handling image formats and conversions.

Frontend

HTML – structure of the web interface.

CSS – styling the UI so it doesn't look like a punishment.

JavaScript – handling form submissions and dynamic responses.

Model & Data Handling

NumPy – numerical operations on image arrays.

Pre-trained Models / Custom CNN – used for feature extraction and classification.

Development & Deployment

Virtual Environment (venv) – dependency isolation.

Localhost Testing – development and debugging environment.

Git & GitHub – version control and collaboration.

Agent 1 – Adaptive Sampling Agent

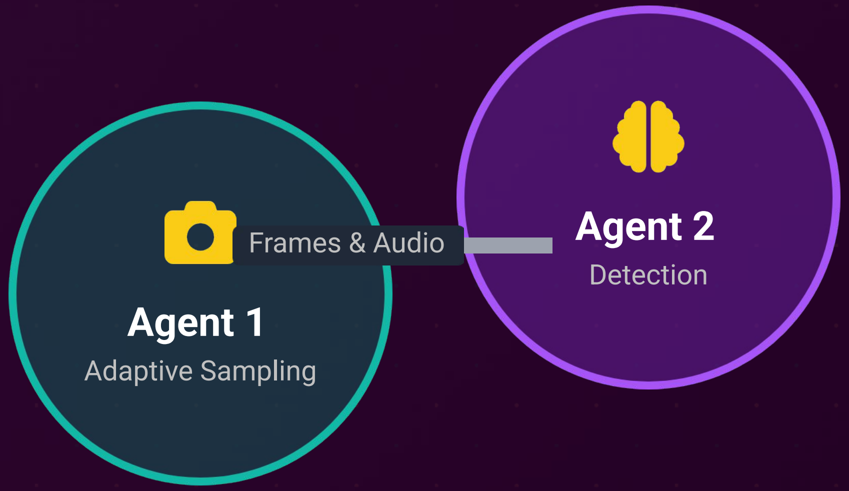
Primary Responsibilities

Analyzes Video Stream
Identifies high-information frames that contain meaningful content for deepfake detection.

Lightweight Processing
Uses simple motion detection methods instead of complex deep learning models.

Frame Selection
Extracts only the most informative frames and their corresponding audio segments.

Agent Position in Architecture

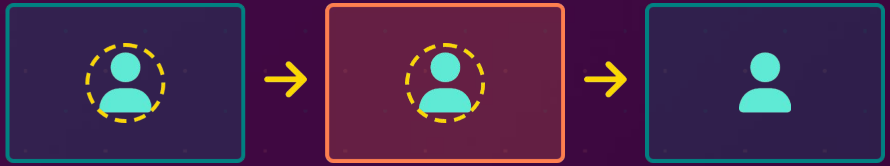


Key Advantage
By focusing on high-information frames, Agent 1 reduces computational load while preserving the most relevant content for deepfake identification.

Lightweight Motion and Change Detection

Frame Differencing

A lightweight technique that identifies changes between consecutive frames by comparing pixel differences.



Resource Efficiency

Computationally lightweight, reducing processing demands.

Motion Detection

Effectively captures facial movements and lip synchronization.

Scene Change Recognition

Identifies sudden changes in background that may indicate manipulation.

Implementation in Adaptive Sampling

- Frames with significant changes are selected for analysis
- Similar frames are skipped to reduce computational load

Selected Skipped

Agent 2 – Detection Agent

Detection Agent

Specialized for deepfake analysis

The Detection Agent processes selected frames and audio segments to identify deepfakes with high accuracy.

Processing Sampled Data

Receives selected frames and audio segments from the Adaptive Sampling Agent, focusing only on high-information content.

Synchronizing Audio-Video Data

Uses timestamps to ensure precise alignment between video frames and audio segments for accurate analysis.

Applying Detection Models

Runs deepfake detection models to identify visual flaws, audio issues, or lip-sync problems that indicate manipulation.

Generating Final Prediction

Compiles analysis results into a final determination of whether the media is authentic or fake.

Synchronization Between the Two Agents

Why Synchronization Matters

Coordination between agents ensures audio and video data remain properly aligned, preventing misalignment that could lead to incorrect detection results.

Agent 1

Adaptive Sampling

- Analyzes video stream
- Detects high-info frames
- Extracts framed with stamps

Data Queue

- Frame 1 (T1)
- Audio 1 (T1)
- Frame 2 (T2)
- Audio 2 (T2)

Agent 2

Deepfake Detection

- Receives synced data
- Applies deepfake detection
- Generates final prediction

Timestamp Synchronization

Timestamps ensure frames and audio segments are correctly paired, maintaining alignment throughout processing.

Buffer Management

A shared buffer allows efficient data transfer between agents while handling varying processing speeds.

Benefits of the Proposed System

Reduced Computational Demands

The system avoids processing redundant frames, significantly reducing the computational load required for deepfake detection.

Faster Processing

By focusing only on high-information frames, the system dramatically decreases processing time without sacrificing detection accuracy.

Improved Resource Efficiency

The adaptive approach maintains detection accuracy while optimizing resource usage, making it suitable for deployment in resource-constrained environments.

Enhanced Scalability

The modular two-agent design simplifies system expansion and makes it easier to scale the solution to handle increasing video loads.

Applications of the System

The adaptive frame sampling with two-agent architecture for deepfake detection has versatile applications across multiple domains:

Social Media Moderation

Automated detection of deepfakes in user-generated content to maintain platform integrity and combat misinformation.

Digital Forensics

Investigation and analysis of digital media to determine authenticity in criminal and civil cases.

Online Identity Verification

Secure authentication systems that verify user identity and detect synthetic or deepfaked identities.

Content Authentication

Verification of media authenticity for news outlets, publishers, and content creators to combat deepfake misinformation.

The system's optimized structure makes it suitable for both offline analysis and real-time monitoring, providing flexible deployment options.

Conclusion

By skipping full-frame processing and using adaptive frame sampling, our system effectively focuses on high-information areas while maintaining detection accuracy.

Selective Frame Analysis

Focuses on informative frames during motion, reducing unnecessary processing of static frames.

Computational Efficiency

Significantly reduces computing costs while maintaining detection accuracy.

Specialized Agent Collaboration

The synchronized two-agent architecture ensures accurate and scalable deepfake detection.

Practical Solution

Provides a workable solution to current deepfake detection challenges.

This method offers a promising approach for real-time and large-scale deepfake detection systems.