

TruthSense: Advanced Lie Detection System

Project Overview

Name: TruthSense (Lie Detection System with CNN-LSTM)

Goal: To provide an AI-powered system that detects deception by analyzing facial micro-expressions, voice patterns, and body language in videos.

Problem Solved: Traditional lie detection relies on subjective human judgment and can be inconsistent. TruthSense addresses this by using artificial intelligence to identify subtle cues invisible to the human eye, achieving approximately 85% accuracy in deception detection.

Target Users:

- Interview analysts (HR departments, journalism)
- Security and law enforcement professionals
- Academic researchers studying deception
- General users interested in analyzing truthfulness

Architecture & Tech Stack

System Architecture

The application follows a client-server architecture:

- **Frontend:** Single-page application (SPA)
- **Backend:** API server handling video processing and ML inference

Technology Stack

Frontend

- **Next.js:** React framework for production applications
- **TypeScript:** For type safety and better developer experience
- **Tailwind CSS:** Utility-first CSS framework for rapid UI development
- **Shadcn UI:** Component library for UI elements
- **Framer Motion:** Animation library

Backend

- **Python:** Primary programming language

- **FastAPI:** Modern, high-performance web framework
- **PyTorch:** Deep learning framework for neural networks
- **OpenCV:** Computer vision library for video and image processing
- **Librosa:** Audio processing for speech analysis

Deployment

- **Docker:** Containerization for consistent environments
- **Cloud Support:** Deployment options for AWS and Azure

Key Features & Modules

Core Components

1. Video Upload System

- Handles video file uploads (MP4, AVI, MOV, MKV formats)
- Secure storage and processing
- Found in `/components/upload.tsx`

2. Video Processing Pipeline

- Frame extraction from videos
- Facial detection and analysis
- Audio feature extraction
- Located in `/backend/video_processor.py`

3. ML Model Components

- CNN-LSTM architecture for analyzing visual and audio features
- Model training functionality in `/backend/model_trainer.py`
- Prediction service in `/backend/predictor.py`

4. Results Display

- Visual presentation of truth/deception prediction
- Confidence scoring
- Feature breakdown showing contributing factors
- Found in `/components/result-display.tsx`

Component Interactions

- The frontend sends video uploads to the backend API

- The backend processes videos and runs ML inference
- Results are returned to the frontend for visualization
- API client in `/lib/api.ts` manages communication

Data Handling

Data Types

- **Video Files:** User-uploaded content for analysis
- **Extracted Frames:** Facial images captured at intervals
- **Audio Features:** Processed speech patterns
- **Prediction Results:** Truth/lie classification with confidence scores

Data Storage

- **Uploaded Videos:** Temporarily stored in `/backend/uploads/`
- **Processed Data:** Saved in `/backend/processed/`
- **Trained Models:** Stored in `/backend/models/`
- **Result Caching:** Implemented via `/backend/result_cache.py`

Data Flow

1. Video upload → preprocessing → feature extraction → model prediction → results display

AI/ML Components

Model Architecture

The system uses a CNN-LSTM hybrid model:

Visual Processing

- **CNN Feature Extractor:**
 - Convolutional layers extract spatial features from face frames
 - Processes facial micro-expressions and visual cues
 - Architecture: 4 convolutional blocks (16→32→64→128 channels)

Temporal Processing

- **LSTM Sequence Analyzer:**
 - Processes sequences of facial features over time
 - Captures temporal patterns in expressions

- 2-layer bidirectional LSTM with 64 hidden units

Audio Processing

- **Audio Feature Network:**
 - Processes extracted audio features (MFCC, spectral features)
 - Captures vocal stress patterns
 - Uses fully connected layers with dropout

Fusion and Classification

- **Multimodal Fusion:**
 - Combines visual and audio features
 - Final classification layers with dropout
 - Binary output (truth vs. lie)

Training Process

- Dataset preparation with truth/lie video pairs
- Preprocessing to extract faces and audio features
- Model training with configurable parameters
- Evaluation based on accuracy metrics

Model Performance

- **Reported Accuracy:** 85%
- **Evaluation Method:** Standard metrics (accuracy, confusion matrix)
- **Performance Storage:** Model metadata files track training results

Security and Privacy

Security Measures

- API security (CORS, rate limiting recommendations)
- Data security for uploaded videos
- HTTPS recommended for all connections

Privacy Considerations

- Temporary storage of uploaded videos
- No permanent data storage of user content
- System designed for on-demand analysis

Deployment & DevOps

Deployment Options

1. **Docker Deployment** (recommended)
 - Container-based deployment for consistency
 - Docker Compose for orchestration
2. **Manual Deployment**
 - Separate frontend and backend deployment
 - Detailed in `DEPLOYMENT.md`
3. **Cloud Deployment**
 - AWS deployment options (ECS, EC2, S3, CloudFront)
 - Azure deployment options (App Service, Container Instances)

Environment Requirements

- **CPU:** 4+ cores (8+ preferred)
- **RAM:** 8GB minimum (16GB+ recommended)
- **Storage:** 10GB+ free space
- **GPU:** NVIDIA with CUDA support (recommended)
- **OS:** Ubuntu 20.04+ or Windows Server 2019+

Monitoring & Maintenance

- Logging recommendations include ELK stack
- Performance monitoring for API and resources
- Regular model retraining schedule

Challenges & Bottlenecks

Technical Challenges

1. **Video Processing Performance**
 - High computational requirements
 - GPU acceleration recommended
2. **Model Accuracy Limitations**

- Achieving high accuracy in lie detection is inherently challenging
- Current 85% accuracy leaves room for improvement

3. Dataset Quality Dependencies

- System performance depends on training data quality
- Diverse, balanced datasets required

Operational Challenges

1. Infrastructure Requirements

- GPU hardware recommended for optimal performance
- Scaling for multiple concurrent users

2. Deployment Complexity

- Multiple components require careful orchestration
- Environment consistency critical for ML components

Future Plans

Planned Features

1. Authentication System

- User authentication and access control
- JWT-based authentication for API

2. Improved ML Model

- Enhanced accuracy through model improvements
- Additional detection features

3. Result Storage

- Database integration for storing analysis history
- User dashboard for past analyses

4. Deployment Enhancements

- Comprehensive Docker Compose setup
- CI/CD pipeline for testing and deployment

5. Advanced Model Customizations

- Alternative CNN architectures (ResNet, EfficientNet)
- Attention mechanisms
- Transfer learning from pre-trained models

Additional Resources

- **Documentation:** Comprehensive guides for setup, deployment, and development
- **Training Guide:** Detailed instructions for model training in `backend/TRAINING_GUIDE.md`
- **Setup Summary:** Quick start guide in `SETUP_SUMMARY.md`

© 2025 TruthSense AI. All rights reserved.