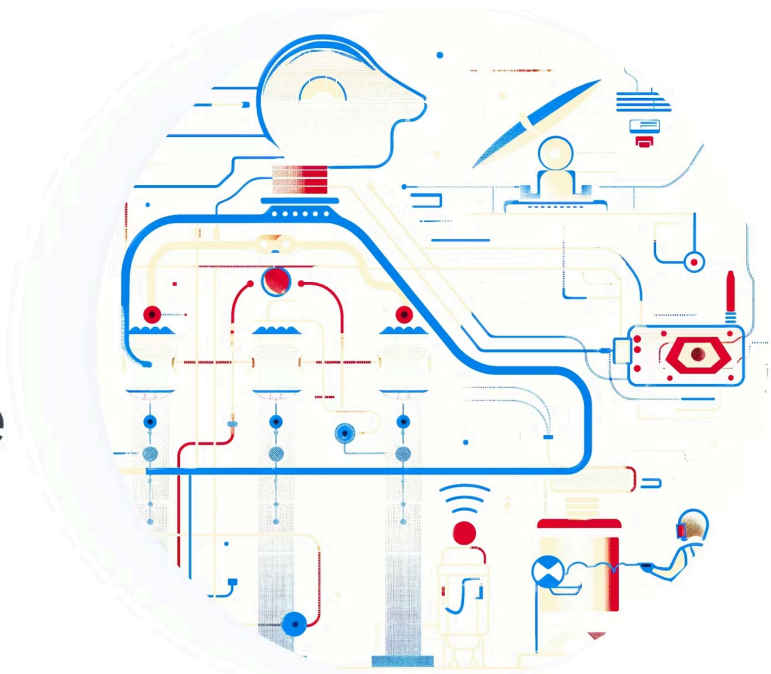


# MediaPipe Multi-Modal Detection System

A Professional, Production-Ready System for Real-Time Computer Vision Analysis

Python 3.8+ MediaPipe 0.10+ OpenCV 4.8+ License MIT










## Overview

An advanced multi-modal detection system integrating:

- **Face Detection & Emotion Recognition** - 7 emotions with 468 facial landmarks
- **Hand Tracking & Gesture Recognition** - 8+ gestures with 21 hand landmarks
- **Pose Estimation & Posture Analysis** - Full body with 33 pose landmarks
- **Object Detection** - 80+ objects from COCO dataset
- **Contextual Analysis** - Intelligent interpretation combining all modalities

## Key Features

### Core Detection

-  Real-time face mesh detection with 468 landmarks
-  Emotion recognition using DeepFace (happy, sad, angry, fear, surprise, disgust, neutral)
-  Hand tracking with 21 landmarks per hand (supports 2 hands)
-  Gesture recognition (Thumbs Up, Peace, OK, Fist, Open Hand, Pointing, Rock)
-  Full body pose estimation with 33 landmarks
-  Posture analysis (upright, slouching, tilted, forward head, neutral)
-  Object detection supporting 80+ categories

### Professional Features

- 📊 **Real-time Performance Monitoring** - FPS tracking and detailed metrics
- 🎯 **Contextual Analysis** - Intelligent interpretation (e.g., "Tech Frustration", "Positive Feedback")
- ⚙️ **Modular Architecture** - Clean separation of concerns
- 📝 **Comprehensive Logging** - Color-coded console + file logging
- 🔧 **Centralized Configuration** - Easy customization via config files
- 🛡️ **Robust Error Handling** - Graceful degradation
- 📸 **Frame Capture** - Save annotated frames with timestamp
- 🎨 **Beautiful UI** - Modern gradient overlays with status indicators

## 📁 Project Structure

```

Detection_MediaPipe_/
├── src/                                # Source code
│   ├── __init__.py
│   ├── core/                          # Core functionality
│   │   ├── __init__.py
│   │   ├── config.py                 # Configuration management
│   │   ├── logger.py                 # Logging system
│   │   └── performance.py            # Performance monitoring
│   ├── detectors/                    # Detection modules
│   │   ├── __init__.py
│   │   ├── base.py                   # Base detector class
│   │   ├── emotion.py                # Emotion detection (DeepFace)
│   │   ├── gesture.py                # Gesture recognition
│   │   ├── posture.py                # Posture analysis
│   │   └── context.py                # Context analysis
│   ├── visualization/                # Rendering
│   │   ├── __init__.py
│   │   └── renderer.py              # Visualization renderer
│   └── system.py                     # Main system orchestrator
├── models/                          # MediaPipe models (downloaded)
│   ├── README.md
│   ├── face_landmarker.task
│   ├── hand_landmarker.task
│   ├── pose_landmarker_lite.task
│   └── efficientdet_lite0.tflite
├── tests/                            # Unit tests
│   ├── __init__.py
│   └── test_detectors.py
├── scripts/                          # Utility scripts
│   ├── __init__.py
│   ├── download_models.py            # Auto-download models
│   └── check_models.py               # Verify model files
├── output/                           # Generated output
│   ├── logs/                         # Application logs
│   ├── frames/                       # Saved frames
│   └── videos/                       # Saved videos
├── docs/                             # Documentation
│   └── user_guide.md
├── app.py                            # Main entry point
└── requirements.txt                  # Python dependencies

```

```
├── setup.py                # Package configuration
├── .gitignore
└── README.md              # This file
```

## Quick Start

### Prerequisites

- **Python 3.8+** ([Download](#))
- **Webcam/Camera**
- **macOS, Linux, or Windows**

### Installation

#### Option 1: Standard Installation (Recommended)

```
# 1. Navigate to project directory
cd /Users/ahmedziada/Documents/Route/Detection_MediaPipe_

# 2. Create virtual environment
python3 -m venv venv

# 3. Activate virtual environment
source venv/bin/activate          # macOS/Linux
# OR
venv\Scripts\activate            # Windows

# 4. Install dependencies
pip install -r requirements.txt

# 5. Download MediaPipe models
python scripts/download_models.py

# 6. Run the application
python app.py
```





#### Option 2: Package Installation (Advanced)

```
# Install as a package (editable mode for development)
pip install -e .

# Run from anywhere
mediapipe-detect
```

### First Run

On first run, the system will:

- 1.  Validate all model files exist
- 2.  Load 4 MediaPipe models (~25 MB total)
- 3.  Initialize camera (10-frame warmup)
- 4.  Start real-time detection at 1280x720 resolution

 Usage

Keyboard Controls

| Key | Action                                      |
|-----|---|
| Q   | Quit application                            |
| S   | Save current frame with timestamp           |
| 1   | Toggle face detection & emotion analysis    |
| 2   | Toggle hand detection & gesture recognition |
| 3   | Toggle pose detection & posture analysis    |
| 4   | Toggle object detection                     |
| R   | Show detailed performance report            |

Display Layout

[LEFT PANEL]

- Emotion Info
- Confidence Bar
- Top 3 Emotions
- Gestures
- Posture
- Objects
- Context Analysis

[VIDEO FEED]

[PERF PANEL]

- FPS: 30.2
- Frame: 1234
- Time: 33ms

[STATUS BAR: Face ON | Hands ON | Pose ON | Objects ON]

Configuration

Edit `src/core/config.py` to customize:

```
# Camera Settings
@dataclass
class CameraConfig:
    index: int = 1          # Camera index (0, 1, 2...)
    width: int = 1280       # Resolution width
    height: int = 720       # Resolution height
    fps: int = 30           # Target FPS
```

```
# Processing Settings
@dataclass
class ProcessingConfig:
    emotion_frame_skip: int = 10 # Analyze emotion every N frames
    max_detection_faces: int = 1 # Max faces to detect
    max_detection_hands: int = 2 # Max hands to detect

# Detection Thresholds
@dataclass
class DetectionThresholds:
    object_confidence: float = 0.5 # Object detection threshold (0.0-1.0)
```

## Performance

| Metric  | Value      | Hardware          |
|---------|------------|-------------------|
| FPS     | 25-35      | MacBook Air M1    |
| Latency | <40ms      | MacBook Air M1    |
| Memory  | ~500MB-1GB | During operation  |
| CPU     | 40-60%     | Single core usage |

## Optimization Tips

### For Higher FPS:

```
# 1. Increase emotion analysis interval
processing_config.emotion_frame_skip = 20 # Analyze less frequently

# 2. Disable unused detectors
# Press 1, 2, 3, or 4 during runtime

# 3. Lower resolution (in config.py)
camera_config.width = 640
camera_config.height = 480
```

### For Better Accuracy:

```
# Lower detection thresholds
detection_thresholds.object_confidence = 0.3 # Detect more objects
```

## Detection Capabilities

### Emotions (7 Categories)

- Happy, Sad, Angry, Fear, Surprise, Disgust, Neutral
- Real-time confidence scores
- Emotion history tracking

## Gestures (8+ Types)

- 👍 Thumbs Up / 👎 Thumbs Down
- 🕊 Peace Sign
- 🏆 OK Sign
- 🖐 Open Hand
- 🦊 Fist
- 📌 Pointing
- 🖖 Rock Sign

## Postures (5 Types)

- Upright & Confident
- Slouching
- Tilted/Asymmetric
- Forward Head Posture
- Neutral Posture

## Context Patterns (8+ Scenarios)

- [⬆] Positive Feedback (happy + thumbs up)
- [! ] Tech Frustration (frustrated + laptop detected)
- [~ ] Tired/Stressed (sad + slouching)
- [\* ] Appears Focused (neutral + upright posture)
- [= ] Peaceful/Relaxed (happy + peace sign)
- [^ ] Excited/Enthusiastic
- [>] Presenting/Speaking
- [# ] Working/Concentrating

## 🔧 Troubleshooting

### Issue: Camera Not Found

```
# Solution 1: Try different camera index
camera_config.index = 0 # Try 0, 1, 2...

# Solution 2: List available cameras
python -c "import cv2; [print(f'Camera {i}') for i in range(10) if
cv2.VideoCapture(i).isOpened()]"
```

### Issue: Models Not Loading

```
# Re-download models
python scripts/download_models.py

# Verify models exist
python scripts/check_models.py

# Check model directory
ls -la models/
```

### Issue: Low FPS (<15)

```
# Quick fixes:
# 1. Increase emotion skip
processing_config.emotion_frame_skip = 20

# 2. Lower resolution
camera_config.width = 640
camera_config.height = 480

# 3. Disable pose detection (press 3)
# 4. Close other applications
```

### Issue: Import Errors

```
# Reinstall dependencies
pip install --upgrade -r requirements.txt

# Or install as package
pip install -e .
```

### Issue: DeepFace Errors

```
# Update DeepFace
pip install --upgrade deepface tensorflow

# Verify installation
python -c "from deepface import DeepFace; print('OK!')"
```



## Documentation

- [User Guide](#) - Comprehensive usage guide
- [API Reference](#) - Code documentation
- [Models README](#) - Model information

## Testing

```
# Run unit tests
python -m pytest tests/ -v






# Run with coverage
python -m pytest tests/ --cov=src --cov-report=html

# Test specific module
python -m pytest tests/test_detectors.py -v
```

## What is **setup.py**?

**setup.py** makes your project installable as a Python package:

### Benefits:

-  Install with **pip install .**
-  Create command-line tools (**mediapipe-detect**)
-  Clean imports without path hacks
-  Distribute to PyPI
-  Development mode: **pip install -e .**

### Usage:








```
# Install in development mode (changes reflect immediately)
pip install -e .

# Run from anywhere after installation
mediapipe-detect

# Or import in other projects
from src.detectors import EmotionAnalyzer
```

## Contributing

Contributions welcome! Areas for improvement:

-  Additional gesture recognition patterns
-  More emotion models (age, gender)
-  Activity recognition (walking, running, sitting)
-  Multi-person tracking
-  GPU acceleration (CUDA support)
-  Video file processing
-  Data export (CSV, JSON)




## License

MIT License - See [LICENSE](#) file for details

## Acknowledgments

- [MediaPipe](#) by Google - Computer vision framework
- [DeepFace](#) by Sefik Ilkin Serengil - Facial analysis
- [OpenCV](#) - Computer vision library
- **Community** - Open source contributors

## Support

-  **Issues:** [GitHub Issues](#)
-  **Email:** [ahmedaliziada@outlook.com](mailto:ahmedaliziada@outlook.com)
-  **Docs:** [Documentation](#)

## Links

- [MediaPipe Solutions](#)
- [DeepFace GitHub](#)
- [OpenCV Tutorials](#)

---

**Version 2.0.0 | Last Updated: 2024**

Made with  using MediaPipe, OpenCV, and Python

[↑ Back to Top](#)