

# Quick Start Guide

## Get Running in 3 Minutes!

### Step 1: Install Python Dependencies

```
bash  
pip install -r requirements.txt
```

OR install manually:

```
bash  
pip install flask flask-cors pillow numpy opencv-python scipy
```

### Step 2: Start the Server

```
bash  
python app.py
```

Wait for:

```
 Deepfake Detection System Starting...  
 Server running at: http://127.0.0.1:5000
```

### Step 3: Open Browser

Navigate to: <http://127.0.0.1:5000>

### Step 4: Upload & Analyze

1. Upload a **real image** (left side)
2. Upload a **test image** (right side)
3. Click " **Analyze Images**"
4. Review the results!

---

### What You'll See

#### Results Include:

-  **Classification:** Real / AI-Generated / Downloaded

- **Confidence Score:** 0-100%
  - **Forensic Correlation:** 0.0-1.0
  - **EXIF Status:** Present/Missing
  - **Compression Score:** Detection level
  - **AI Pattern Score:** Synthetic indicators
  - **Technical Justification:** Why this classification?
- 

## Testing Tips

### Test Case 1: Real Photos (Same Camera)

- **Upload:** Two photos from your phone
- **Expected:** High correlation (>0.75), "Real Image"

### Test Case 2: AI-Generated Image

- **Upload:** Real photo + AI-generated image (from DALL-E/Midjourney)
- **Expected:** Low correlation (<0.4), "AI-Generated"

### Test Case 3: Downloaded Image

- **Upload:** Real photo + screenshot/downloaded image
  - **Expected:** Moderate correlation, "Downloaded/Re-uploaded"
- 

## Common Issues

**Error: "Module not found"** → Run: `pip install -r requirements.txt`

**Error: "Port 5000 in use"** → Change port in `app.py` line: `app.run(debug=True, port=5001)`

**Images not uploading?** → Check file size (<10MB), use JPG/PNG format

---

## Required Files

Make sure you have:

- `app.py` (backend)
- `index.html` (frontend)
- `requirements.txt` (dependencies)

---

## For Research Demo

When presenting to review panel:

1. Show live analysis with different image types
  2. Explain the forensic correlation score
  3. Highlight the technical justification text
  4. Expand "Technical Details" to show EXIF comparison
  5. Demonstrate different classification categories
- 

Ready? Let's detect some deepfakes! 