# Phase 2 AI Video Analysis - Testing Guide

## Quick Start

### 1. Start the Development Server

```
cd /home/ubuntu/mindful_champion/nextjs_space
npm run dev
```

The app should start on `http://localhost:3000`

### 2. Sign In

Navigate to the app and sign in with your user account.

### 3. Navigate to Video Analysis

Go to `/train/video` in your app to access the video analysis feature.

## Testing Checklist

### ✅ Video Upload Test

1. **Prepare Test Video**
   - Use a short pickleball video (10-30 seconds recommended for first test)
   - Supported formats: MP4, MOV, AVI, WebM
   - Max size: 100MB

2. **Upload Video**
   - Click “Upload Video” button
   - Select your test video
   - Verify upload success message
   - Note the video ID

3. **Expected Results**
   - Video appears in library
   - Status shows “PENDING”
   - Video player displays the uploaded video

### ✅ Video Analysis Test

1. **Trigger Analysis**
   - Click “Analyze” button on uploaded video
   - Or navigate to video detail page

2. **Monitor Progress**
   - Status should change to “PROCESSING”
   - Processing takes 30-60 seconds for a 1-minute video

3. **View Results**
   - Status changes to “COMPLETED”

- Overall score is displayed (0-100)
- Strengths list is populated
- Areas for improvement are shown
- Recommendations are provided
- Shot breakdown is visible
- Key moments with timestamps are available

### ✅ Analysis Quality Test

#### Check Shot Detection

- [ ] Serves are identified correctly
- [ ] Forehand/backhand shots are detected
- [ ] Volleys are recognized
- [ ] Dinks are identified
- [ ] Shot counts are reasonable

#### Check Technical Scores

- [ ] Overall score is between 0-100
- [ ] Paddle angle score is calculated
- [ ] Follow-through score is present
- [ ] Body rotation score is shown
- [ ] Ready position score is displayed

#### Check Movement Metrics

- [ ] Court coverage percentage is shown
- [ ] Average speed is calculated
- [ ] Efficiency score is present
- [ ] Positioning quality is evaluated

#### Check Personalized Feedback

- [ ] At least 2-3 strengths are identified
- [ ] At least 2-3 areas for improvement are listed
- [ ] Recommendations are actionable
- [ ] Key moments have timestamps
- [ ] Feedback is relevant to video content

## Manual API Testing

### Using curl

#### 1. Upload Video

```
# Get your auth token from browser DevTools (Application > Cookies > next-
auth.session-token)
TOKEN="your_session_token_here"

curl -X POST http://localhost:3000/api/video-analysis/upload \
  -H "Cookie: next-auth.session-token=$TOKEN" \
  -F "file=@path/to/your/video.mp4"
```

Expected response:

```
{
  "success": true,
  "videoId": "clxxx...",
  "videoUrl": "/uploads/videos/1234567890-video.mp4",
  "message": "Video uploaded successfully"
}
```

### 2. Start Analysis

```
VIDEO_ID="video_id_from_upload"

curl -X POST http://localhost:3000/api/video-analysis/analyze \
  -H "Cookie: next-auth.session-token=$TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"videoId\": \"$VIDEO_ID\"}"
```

Expected response:

```
{
  "success": true,
  "analysis": {
    "overallScore": 78,
    "strengths": ["Excellent stance quality...", ...],
    "areasForImprovement": [...],
    ...
  }
}
```

### 3. Get Video Details

```
curl http://localhost:3000/api/video-analysis/$VIDEO_ID \
  -H "Cookie: next-auth.session-token=$TOKEN"
```

### 4. Get Video Library

```
curl "http://localhost:3000/api/video-analysis/library?limit=10" \
  -H "Cookie: next-auth.session-token=$TOKEN"
```

## Debugging

### Check Logs

```
# In development mode, check terminal output for:
- "Starting AI video analysis for: [videoId]"
- "Extracting frames from video..."
- "Detected poses in X frames"
- "Analysis complete!"
```

## Common Issues

### 1. “Video file not found”

**Cause**: Upload didn’t save file correctly
**Fix**: Check that `public/uploads/videos/` directory exists and is writable

### 2. “Failed to initialize pose detection model”

**Cause**: TensorFlow.js initialization failed
**Fix**:

```
# Reinstall TensorFlow dependencies
npm install @tensorflow/tfjs-node @tensorflow-models/pose-detection --legacy-peer-deps
```

### 3. “Cannot find module @mediapipe/pose”

**Cause**: Build trying to import MediaPipe (should use MoveNet)
**Fix**: This should be resolved in current implementation. If you see this:
- Clear Next.js cache: `rm -rf .next`
- Rebuild: `npm run build`

### 4. Analysis takes too long

**Cause**: Video is too long or high resolution
**Solutions**:
- Use shorter test videos (< 1 minute)
- Video will be sampled at 2 fps regardless of original fps
- For production, consider background job processing

### 5. Low pose detection scores

**Cause**: Video quality or framing issues
**Solutions**:
- Ensure person is clearly visible
- Good lighting
- Stable camera
- Person should fill 30-60% of frame
- No obstructions

## Checking Database

```
# Connect to your database
# Check video records
SELECT id, title, "analysisStatus", "overallScore", "analyzedAt"
FROM "VideoAnalysis"
ORDER BY "createdAt" DESC
LIMIT 10;

# Check analysis results
SELECT id, title, strengths, "areasForImprovement", "shotTypes"
FROM "VideoAnalysis"
WHERE "analysisStatus" = 'COMPLETED'
LIMIT 5;
```

# Performance Benchmarks

## Expected Processing Times

- 10 seconds video: ~15-20 seconds
- 30 seconds video: ~30-40 seconds
- 1 minute video: ~45-60 seconds
- 2 minutes video: ~90-120 seconds

## Frame Processing

- Extraction: 2 fps (60 frames per minute)
- Pose detection: ~0.5-1 second per frame
- Analysis: ~2-5 seconds total

# Test Videos

## Good Test Video Characteristics

- **Duration**: 10-60 seconds
- **Content**: Player actively playing (not just standing)
- **Framing**: Player takes up 30-60% of frame
- **Quality**: 720p or higher
- **Lighting**: Well-lit environment
- **Camera**: Relatively stable (not shaky)
- **Angle**: Side or diagonal view showing full body

## What to Include

- Multiple shot types (serves, groundstrokes, volleys)
- Movement around the court
- Ready position between shots
- Complete stroke motions

## What to Avoid

- Very short videos (< 5 seconds)
- Static/no movement
- Player too far away
- Poor lighting
- Obstructed view
- Extreme camera angles

# Success Criteria

## ✅ Phase 2 is working correctly if:

1. **Video Upload**:
   - Files save to disk
   - Database records created
   - Videos appear in library

2. **Video Processing**:
   - Frames extract successfully
   - Pose detection runs without errors
   - Analysis completes in reasonable time

3. **Analysis Results**:
   - Overall score is calculated (0-100)
   - Shot types are detected and classified
   - Technical scores are present
   - Movement metrics are calculated
   - Strengths and improvements are listed
   - Recommendations are actionable

4. **User Experience**:
   - Upload is intuitive
   - Progress is visible
   - Results are clear and helpful
   - Feedback is personalized

## Next Steps After Testing

1. **If tests pass**: Deploy to production!
2. **If issues found**: Check debugging section above
3. **For optimization**: Consider GPU acceleration
4. **For features**: See Phase 3 enhancements in summary document

## Support

If you encounter issues not covered in this guide:

1. Check console logs (both browser and server)
2. Verify all dependencies are installed
3. Ensure database is accessible
4. Check file permissions on upload directory
5. Review error messages carefully

## Test Reporting Template

```
### Test Report

**Date**: [Date]
**Video**: [Video name/length]
**User**: [Test user]

#### Upload
- ✅/❌ File uploaded successfully
- ✅/❌ Appears in library
- ✅/❌ Video plays correctly

#### Analysis
- ✅/❌ Analysis started
- ✅/❌ Processing completed
- ✅/❌ Results saved
- Processing time: [X seconds]

#### Results Quality
- Overall Score: [X/100]
- Shots Detected: [X]
- Shot Types: [List]
- Strengths: [Count]
- Improvements: [Count]
- Recommendations: [Count]

#### Issues Found
[List any issues]

#### Notes
[Any additional observations]
```

---

Happy Testing! 🎾🤖