Video Storage & Shot Detection Investigation Report

Date: November 29, 2025
Investigated By: DeepAgent
Project: Mindful Champion

🎯 Executive Summary

This investigation was triggered by multiple videos showing loading errors on analysis pages and concerns about shot-by-shot detection functionality. The investigation revealed **critical issues** affecting video accessibility and shot detection across the platform.

Key Findings:

1. ❌ **Videos are inaccessible** - Both S3 and local storage have issues
2. ❌ **Shot detection is not working** - 0% of videos have shot detection data
3. ⚠️ **Two specific analysis IDs not found** in database
4. ⚠️ **Architecture mismatch** between S3 storage and local file processing

---

📊 Investigation Results

1. Database Analysis

Specific Video IDs (Reported by User):

- `cmikrn5oa000r1080fsm20mu` - **NOT FOUND** in database
- `cmikvanik000dib09s62kb2x` - **NOT FOUND** in database

These records either:
- Were deleted from the database
- Never existed with these IDs
- Were from a different environment/database

Storage Statistics (10 Total Videos Analyzed):

```
Total video analyses: 10
☁️ S3 storage:        3 videos (30.0%) - Recent uploads
💾 Local storage:     7 videos (70.0%) - Older uploads

Shot Detection Coverage: 0 videos (0.0%) ❌ CRITICAL ISSUE
```

Recent Video Records:

S3-Stored Videos (Most Recent):

1. `cmikvzx9z001alb097tx...` - Nov 29, 22:52 UTC
- User: deansnow59@gmail.com
- S3 Path: `6482/uploads/1764456759635-IMG_4404.mov`

- Status: COMPLETED
- Shot Detection: ❌ No data

1. `cmikvib4j000ulb098c3...` - Nov 29, 22:38 UTC
   - User: deansnow59@gmail.com
   - S3 Path: `6482/uploads/1764455937720-CD13B324-9947-46FB-A233-6FD2F71239BD.mov`
   - Status: COMPLETED
   - Shot Detection: ❌ No data

2. `cmikvanik000dlb09ks6...` - Nov 29, 22:33 UTC ⚠️ Similar ID to reported!
   - User: deansnow59@gmail.com
   - S3 Path: `6482/uploads/1764455580467-IMG_4404.mov`
   - Status: COMPLETED
   - Shot Detection: ❌ No data

Local-Stored Videos:
- 7 videos with paths like `/uploads/videos/[timestamp]-[filename]`
- Upload dates: Nov 9 - Nov 28, 2025
- All missing shot detection data

---

2. Local File System Check

Directory: `/home/ubuntu/mindful_champion/nextjs_space/public/uploads/videos/`

Files Found: 4 video files (66MB total)

```
✅ 1762899446328-IMG_4404.mov (17MB)
✅ 1762905185345-IMG_4404.mov (17MB)
✅ 1762913541285-IMG_4404.mov (17MB)
✅ 1762913779523-IMG_4404.mov (17MB)
```

Missing Files: At least 3 videos referenced in database are **NOT** on disk:

```
❌ 1764341752685-IMG_4404__3_.mov
❌ 1763438930791-VID_20251117_230158.mp4
❌ 1763438676055-VID_20251117_230158.mp4
```

Impact: Users viewing these analysis pages will see loading errors because the video files don't exist.

---

3. S3 Storage Configuration

Configuration Found in `.env`:

```
AWS_PROFILE=hosted_storage
AWS_REGION=us-west-2
AWS_BUCKET_NAME=abacusai-apps-c23443d20cd3d54c25905c2c-us-west-2
AWS_FOLDER_PREFIX=6482/
```

✅ S3 is properly configured
✅ Recent uploads are storing S3 keys in database
✅ Signed URL generation works

BUT:

❌ **All S3 videos return 403 Forbidden when accessed**

S3 Accessibility Test Results:

```
Video: 6482/uploads/1764456759635-IMG_4404.mov
├─ Signed URL: ✅ Generated
└─ Access Test: ❌ 403 Forbidden

Video: 6482/uploads/1764455937720-CD13B324-9947-46FB-A233-6FD2F71239BD.mov
├─ Signed URL: ✅ Generated
└─ Access Test: ❌ 403 Forbidden

Video: 6482/uploads/1764455580467-IMG_4404.mov
├─ Signed URL: ✅ Generated
└─ Access Test: ❌ 403 Forbidden
```

Root Cause Analysis:

The 403 Forbidden errors indicate one of the following:
1. **Files were never uploaded to S3** - Database has the path, but upload failed
2. **AWS credentials lack permissions** - Can't read objects even with signed URLs
3. **S3 bucket policy issue** - Bucket doesn't allow access
4. **Wrong AWS profile** - Using `hosted_storage` profile which may not have access

---

4. Shot Detection Implementation Analysis

Current Architecture:

Shot Detection Flow:

```
1. Video Upload → Database record created
2. User triggers analysis → /api/video-analysis/analyze
3. AdvancedAnalysisEngine.analyze() → Basic analysis only
4. Shot detection NEVER called automatically ❌
```

Shot Detection Endpoints:
- `/api/video-analysis/detect-shots` - Standalone endpoint (not integrated)
- `/api/video-analysis/analyze-enhanced` - Includes shot detection (not used by frontend)

Critical Issue:

The shot detection system (`LLMShotDetector`) is implemented but **NOT INTEGRATED** into the standard analysis flow:

```
// From analyze-enhanced/route.ts (NOT USED)
const detector = new LLMShotDetector(videoId, progressCallback);
const shotDetectionResult = await detector.detectShots(videoPath);
```

vs.

```
// From analyze/route.ts (ACTUALLY USED)
const engine = new AdvancedAnalysisEngine()
const analysisResult = await engine.analyze({...}) // No shot detection!
```

Frontend Integration:

```
// components/train/video-analysis-hub.tsx
const res = await fetch('/api/video-analysis/analyze', {...})
// ☝️ Calls the route WITHOUT shot detection
```

---

5. Architecture Mismatch: S3 Storage vs Local Processing

The Fatal Flaw:

Shot detection requires local file access:

```
// analyze-enhanced/route.ts
const videoPath = path.join(process.cwd(), 'public', videoUrl);
//                                                   ^^^^^^^^^^^^
//                                          Assumes local file path!
```

But S3 videos don't exist locally:

```
Database videoUrl: "6482/uploads/1764456759635-IMG_4404.mov"
Constructed path:  "/home/ubuntu/.../public/6482/uploads/1764456759635-IMG_4404.mov"
Reality:           ❌ File doesn't exist locally, it's in S3!
```

The `LLMShotDetector` uses FFmpeg to extract frames:

```
// llm-shot-detector.ts
async extractKeyFrames(videoPath: string) {
  // Uses exec('ffmpeg -i videoPath ...')
  // ☝️ Requires actual file on disk
}
```

Result: Shot detection fails silently for S3 videos and falls back to empty array.

🚨 Critical Issues Summary

Issue	Severity	Impact	Status
Videos not found in S3 (403 errors)	🔴 Critical	Users can't watch videos	Blocking
Local videos missing from disk	🔴 Critical	Users can't watch videos	Blocking
Shot detection not integrated	🔴 Critical	Feature completely broken	Blocking
S3/Local architecture mismatch	🟠 High	Shot detection can't work with S3	Blocking
Specific analysis IDs missing	🟡 Medium	Specific users affected	Information

---

🔧 Recommended Solutions

Solution 1: Fix S3 Video Accessibility (URGENT)

Option A: Verify S3 Upload Actually Works

```
// In upload/route.ts, verify upload success:
const uploadResult = await uploadFile(buffer, fileName, isPublic, contentType);
// Then immediately test accessibility:
const testUrl = await getFileUrl(uploadResult, isPublic);
const response = await fetch(testUrl, { method: 'HEAD' });
if (!response.ok) throw new Error('Upload verification failed');
```

Option B: Check AWS Credentials

```
# Verify the AWS profile has proper permissions:
aws s3 ls s3://abacusai-apps-c23443d20cd3d54c25905c2c-us-west-2/6482/uploads/ --profile hosted_storage
```

Option C: S3 Bucket Policy

Ensure bucket allows GetObject with signed URLs:

```
{
  "Effect": "Allow",
  "Action": ["s3:GetObject"],
  "Resource": "arn:aws:s3:::bucket-name/6482/*"
}
```

Solution 2: Integrate Shot Detection

Option A: Integrate into Main Analysis Flow

```
// In app/api/video-analysis/analyze/route.ts
import { LLMShotDetector } from '@/lib/video-analysis/llm-shot-detector';

// After basic analysis:
if (videoPath && fs.existsSync(videoPath)) {
  const detector = new LLMShotDetector(videoId, progressCallback);
  const shotResult = await detector.detectShots(videoPath);
  // Merge shotResult into analysisResult
}
```

Option B: Make Frontend Call Shot Detection

```
// After video analysis completes:
await fetch('/api/video-analysis/detect-shots', {
  method: 'POST',
  body: JSON.stringify({ videoId, videoUrl })
});
```

Solution 3: Fix S3/Local Processing Mismatch

Option A: Download S3 Videos Temporarily

```
async function getLocalVideoPath(videoUrl: string, cloudStoragePath: string | null) {
  if (cloudStoragePath) {
    // Video is in S3, download to temp location
    const tempPath = path.join('/tmp', `temp-${Date.now()}-${path.basename(cloudStorag
ePath)}`);
    await downloadFromS3(cloudStoragePath, tempPath);
    return tempPath;
  } else {
    // Local file
    return path.join(process.cwd(), 'public', videoUrl);
  }
}

// Use in analyze-enhanced:
const videoPath = await getLocalVideoPath(videoUrl, existingAnalys-
is.cloud_storage_path);
const shotResult = await detector.detectShots(videoPath);
// Clean up temp file if needed
```

Option B: Make Shot Detector Work with URLs

Modify `LLMShotDetector` to accept S3 signed URLs and stream/download internally.

Solution 4: Migration Strategy for Existing Videos

For videos with missing local files:

```
// Mark as inaccessible in database:
await prisma.videoAnalysis.updateMany({
  where: {
    AND: [
      { cloud_storage_path: null },
      { videoUrl: { startsWith: '/uploads/videos/' } }
    ]
  },
  data: {
    analysisStatus: 'FAILED',
    areasForImprovement: ['Video file no longer available']
  }
});
```

📋 Implementation Priority

Phase 1: Immediate (Fix Video Access) ⚠️

- [ ] Debug why S3 files return 403 Forbidden
- [ ] Verify AWS credentials and S3 bucket permissions
- [ ] Test S3 upload and retrieval end-to-end
- [ ] Add logging to upload process to catch failures

Phase 2: Feature Integration (Enable Shot Detection)

- [ ] Create S3 video download utility
- [ ] Integrate shot detection into main analysis flow
- [ ] Update frontend to show shot detection progress
- [ ] Test with both S3 and local videos

Phase 3: Data Migration (Clean Up)

- [ ] Identify all videos with missing files
- [ ] Mark inaccessible videos in database
- [ ] Add admin tool to re-upload or delete broken records
- [ ] Add file existence checks before analysis

🧪 Testing Scripts Created

Two diagnostic scripts have been created in `/scripts/`:

1. `check-video-analysis.ts`

- Queries database for specific analysis IDs
- Lists recent uploads with storage type
- Shows shot detection coverage statistics
- Usage: `npx tsx scripts/check-video-analysis.ts`

2. `test-video-accessibility.ts`

- Tests S3 signed URL generation

- Verifies S3 file accessibility (HEAD request)
- Checks local file existence
- Provides storage summary
- Usage: `npx tsx scripts/test-video-accessibility.ts`

💡 Additional Recommendations

1. **Add Monitoring:**
   - Log all video upload attempts with success/failure
   - Add S3 upload verification step
   - Alert on 403 errors when generating signed URLs

2. **Improve Error Handling:**
   - Show clear error messages to users when videos are inaccessible
   - Provide option to re-upload failed videos
   - Don't mark analysis as COMPLETED if video is inaccessible

3. **Architecture Improvement:**
   - Standardize on S3 storage for all new uploads
   - Add video proxy API that handles both S3 and local seamlessly
   - Consider CDN for better video delivery

4. **Database Cleanup:**
   - Add `videoAccessible` boolean field to VideoAnalysis model
   - Periodically check video accessibility
   - Archive or delete records with missing videos

📞 Next Steps

Please prioritize:
1. **Immediately** - Fix S3 video accessibility (users can't view their videos)
2. **High Priority** - Integrate shot detection into analysis flow
3. **Medium Priority** - Clean up database records with missing files

Would you like me to implement any of these solutions?

📎 Appendix: Related Files

- Database Schema: `prisma/schema.prisma`
- S3 Library: `lib/s3.ts`
- Shot Detector: `lib/video-analysis/llm-shot-detector.ts`
- Main Analysis API: `app/api/video-analysis/analyze/route.ts`
- Enhanced Analysis API: `app/api/video-analysis/analyze-enhanced/route.ts`
- Upload API: `app/api/video-analysis/upload/route.ts`
- Frontend Component: `components/train/video-analysis-hub.tsx`

Report generated on November 29, 2025