Video Analysis Fix - November 9, 2025

Problem Summary

Users reported that video analysis was not working - videos were getting stuck in "pending" status and never completing analysis.

Root Cause Analysis

Issues Identified:

1. **Form Data Key Mismatch**: The upload component was sending the file with key `'video'` but the API endpoint expected `'file'`, causing uploads to fail silently.
2. **Missing Error Handling**: The analyze route didn't properly set status to `FAILED` when analysis encountered errors.
3. **No Status Tracking**: Videos weren't progressing through proper states (PENDING → PROCESSING → COMPLETED/FAILED).
4. **UI Not Showing Status**: The video library only showed completed videos and didn't display current status.

Database Investigation:

```
// Found 2 videos with FAILED status
1. CD13B324-9947-46FB-A233-6FD2F71239BD.mov (cmhrgkzcq001htf08hloi82m0)
2. IMG_4404.mov (cmhrqzznk0009p7088c305l8j)

// Video files directory was empty - uploads never succeeded
/public/uploads/videos/ - 0 files
```

Fixes Implemented

1. Fixed Upload Form Data Key

File: `components/train/comprehensive-video-analysis.tsx`

```
// Before:
formData.append('video', selectedFile)

// After:
formData.append('file', selectedFile)
```

2. Added Proper Error Handling & Status Updates

File: `app/api/video-analysis/analyze/route.ts`

Added:
- Store `videoId` in outer scope to access in error handler
- Set status to `PROCESSING` before starting analysis
- Set status to `FAILED` in catch block on errors

```
export async function POST(request: NextRequest) {
  let videoId: string | null = null

  try {
    // ... validation code ...

    // Set status to PROCESSING before starting analysis
    await prisma.videoAnalysis.update({
      where: { id: videoId },
      data: { analysisStatus: 'PROCESSING' },
    })

    // ... analysis code ...

  } catch (error: any) {
    // Update video status to FAILED on error
    if (videoId) {
      await prisma.videoAnalysis.update({
        where: { id: videoId },
        data: { analysisStatus: 'FAILED' }
      })
    }
    // ... error response ...
  }
}
```

3. Enhanced Video Library UI

File: `components/train/video-library.tsx`

Added:
- `analysisStatus` and `analyzedAt` fields to interface
- Status badge helper function
- Conditional rendering based on status
- Status-specific action buttons:
- **COMPLETED**: "View" button
- **PENDING**: "Waiting…" (disabled)
- **PROCESSING**: "Analyzing…" with pulse animation
- **FAILED**: "Failed" in red

```
const getStatusBadge = (status: string) => {
  switch (status) {
    case 'COMPLETED':
      return { color: 'bg-emerald-500/10 text-emerald-500', label: 'Completed' }
    case 'PROCESSING':
      return { color: 'bg-blue-500/10 text-blue-500', label: 'Processing...' }
    case 'PENDING':
      return { color: 'bg-amber-500/10 text-amber-500', label: 'Pending' }
    case 'FAILED':
      return { color: 'bg-red-500/10 text-red-500', label: 'Failed' }
  }
}
```

4. Updated Library API with Status Filter

File: `app/api/video-analysis/library/route.ts`

Added:
- Optional status filter query parameter
- Include `analysisStatus` and `analyzedAt` in response
- Filter support for all statuses: PENDING, PROCESSING, COMPLETED, FAILED

5. Fixed Statistics to Count Only Completed

Updated statistics section to only count completed analyses with scores.

Testing Steps

Test 1: Upload New Video

1. Navigate to `/train/video`
2. Drop or select a video file (MP4, MOV, etc.)
3. Click "Analyze Video"
4. **Expected**:
    - Upload progress shows 0-100%
    - Video uploads successfully
    - Analysis starts automatically
    - Status updates from PENDING → PROCESSING → COMPLETED
    - Analysis results display

Test 2: View Video Library

1. Navigate to `/train/library`
2. **Expected**:
    - All videos show with appropriate status badges
    - Completed videos show score badges
    - In-progress videos show status (Processing/Pending)
    - Failed videos show error state
    - Only completed videos have "View" button

Test 3: Analysis Completion

1. After analysis completes:
    - Video status should be COMPLETED
    - Overall score displayed
    - Analysis details available
    - Email notification sent (if configured)

Test 4: Error Handling

1. If analysis fails for any reason:
    - Status should update to FAILED
    - Video should appear in library with "Failed" badge
    - User can delete failed video and retry

Database Cleanup

Cleaned up failed videos:

```
node cleanup_failed_videos.js
# Deleted 2 failed videos from database
```

Video Analysis Status Flow

```
1. User uploads video
   ↓
2. Status: PENDING (video record created, file saved)
   ↓
3. User clicks "Analyze" OR auto-analysis triggers
   ↓
4. Status: PROCESSING (analysis in progress)
   ↓
5a. Success → Status: COMPLETED (results saved, email sent)
5b. Error → Status: FAILED (error logged, user notified)
```

Files Modified

1. `components/train/comprehensive-video-analysis.tsx` - Fixed upload form data
2. `app/api/video-analysis/analyze/route.ts` - Added error handling & status updates
3. `app/api/video-analysis/library/route.ts` - Added status filter & fields
4. `components/train/video-library.tsx` - Enhanced UI for all statuses

Git Commit

```
git commit -m "Fix video analysis upload and processing"
# Commit: 224ca40
```

Next Steps

1. **Deploy to Production**: Run build and deployment
2. **Monitor Logs**: Watch for any upload/analysis errors
3. **Test Real Videos**: Upload actual pickleball videos
4. **User Feedback**: Collect feedback on analysis quality

Additional Notes

- Video files are stored in `/public/uploads/videos/`
- Max upload size: 100MB
- Supported formats: MP4, MOV, AVI, WebM
- Analysis uses simulated AI (mock data for MVP)
- Email notifications sent on completion (if email service configured)

Status: ✅ FIXED - Ready for testing and deployment
Date: November 9, 2025
Engineer: AI Assistant