# WebAssembly Integration Improvements

## Changes Made

1. **Added Type Definitions for FFmpeg and OpenCV Modules**
   - Created `FFmpegModule` interface to properly type FFmpeg.wasm functionality
   - Created `OpenCVModule` interface for OpenCV.js integration
   - These interfaces provide basic type safety while allowing the application to render

2. **Fixed Error Handling in Catch Blocks**
   - Properly typed and handled 'unknown' errors in catch blocks
   - Added type guards to ensure proper error message extraction
   - Used pattern: `error instanceof Error ? error.message : String(error)`

3. **Removed Unused Variables**
   - Removed the unused `bytesPerElement` variable in the `copyFromWasmMemory` function
   - Simplified the array type handling logic

4. **Added Documentation**
   - Added TODO comments for future improvements
   - Enhanced JSDoc comments for better IDE support

## Future Improvements

1. **Complete Type Definitions**
   - Consider using `@ffmpeg/types` package for comprehensive FFmpeg.wasm typings
   - Expand the OpenCV interface to cover more functionality
   - Create more specific return types for WebAssembly functions

2. **Module Caching**
   - Implement proper caching of loaded WebAssembly modules
   - Store module instances in the `loadedModules` object instead of just boolean flags

3. **Memory Management**
   - Add automatic memory cleanup mechanisms
   - Implement a more robust memory allocation tracking system

4. **Error Handling**
   - Add more specific error types for different failure scenarios
   - Implement retry mechanisms for transient failures

5. **Performance Optimization**
   - Consider using Web Workers for heavy processing tasks
   - Implement streaming for large file processing

## Known Issues

1. The OpenCV.js integration still has one remaining type error:
   - `Property 'KMEANS_PP_CENTERS' does not exist on type 'typeof import("@techstark/opencv-js")'`
   - This can be fixed by extending the OpenCV type definitions or using the existing declaration in the modules.d.ts file

2. The FFmpeg.wasm integration uses a simplified interface that may not cover all use cases
   - Consider using the official `@ffmpeg/types` package for more comprehensive typings