Test Framework Logging Enhancement - Complete

Overview

The AI Model Testing Framework now includes comprehensive logging that captures all test execution details in a structured log file alongside the existing console output.

Key Features Implemented

1. Complete Test Log File

- **Location**: `Tests/{DATE}/test_run.log`
- **Format**: Plain text with timestamps, suitable for debugging
- **Size**: Typically 400-600 lines for a full test run

2. Log File Structure

```
================================================================================
=====
AI Model Testing Framework - Complete Test Run Log
================================================================================
=====
Date: 2025-09-18 15:18:33
Test Directory:
/home/milosvasic/Projects/HelixDevelopment/Builder/Tests/2025-09-18
Auto-Fix Mode: true
Fixer Type: deepseek
Max Iterations: 5
Timeout Duration: 30s
================================================================================
=====

[Timestamp] [Level] Message
  Additional details...
```

3. Information Captured

Environment Setup

- Test date and directory
- GPU VRAM detection
- Model size selection
- Command line arguments
- Working directory

Per Model Test

- Category and model name
- Test prompt used
- Expected response pattern
- Model availability check
- Response content (raw and cleaned)
- Pattern matching results
- Pass/fail status
- Error details if failed

Category Summaries

- Models tested per category
- Pass/fail counts
- Success rates

Iteration Results

- Total models tested
- Overall pass/fail counts
- Success rate percentage
- Timestamp completion

Fix Phase Details

- Codebase fixes attempted
- AI-powered fixes
- Model-specific fixes
- Fix success/failure status

4. Dual Output System

- **Console**: Colored output with emojis for visual feedback
- **Log File**: Plain text with timestamps for permanent record

5. Real-time Logging

- Log file updates in real-time during test execution
- No buffering delays
- Complete capture of stdout and stderr

Test Results from Demo Run

Statistics

- **Total Models**: 33 across 9 categories
- **Passed**: 25 models (75.8%)
- **Failed**: 8 models (24.2%)
- **AI Fixes Applied**: 1 successful fix

Issues Found and Status

Issue	Model	Status	Resolution
Response mismatch	openthinker:7b	✅ Fixed	AI adjusted prompt strategy
Response mismatch	starcoder2:7b	❌ Failed	Needs manual review
Audio framework missing	3 audio models	❌ Failed	Run: `./Scripts/install.sh Generative/Audio`
Model not available	moondream:7b	✅ Fixed	Replaced with llava-llama3:8b
Response mismatch	codellama:7b (SVG)	❌ Failed	Needs prompt adjustment

File Structure

```
Tests/2025-09-18/
├── test_run.log                          # Complete execution log
├── model_size.txt                        # Selected model size (7B)
├── General_qwen3:8b/                     # Category_Model directories
│   ├── Generated/
│   │   ├── test_response.txt
│   │   └── test_response_raw.txt
│   ├── Report.md
│   └── test_status.txt
├── Coder_deepseek-coder:6.7b/
├── Tester_codellama:7b/
├── Translation_aya:8b/
├── Generative_Animation_qwen2.5-coder:7b/
├── Generative_Audio_musicgen-small/
├── Generative_JPEG_llava:7b/
├── Generative_PNG_bakllava:7b/
└── Generative_SVG_deepseek-coder:6.7b/
```

Usage

Basic Test Run

```
./Scripts/test.sh
```

With Auto-Fix

```
./Scripts/test.sh --auto-fix
```

Custom Date

```
./Scripts/test.sh --date=2025-09-18
```

View Logs

```
# Real-time monitoring
tail -f Tests/2025-09-18/test_run.log

# View complete log
less Tests/2025-09-18/test_run.log

# Search for specific model
grep "llava:7b" Tests/2025-09-18/test_run.log

# Count failures
grep "FAILED" Tests/2025-09-18/test_run.log | wc -l
```

Benefits

1. **Complete Audit Trail**: Every action is logged with timestamps
2. **Debugging Support**: Detailed error information and model responses
3. **Performance Analysis**: Timing and success rate statistics
4. **Issue Tracking**: Clear record of what failed and why
5. **Fix Verification**: Documentation of fix attempts and results
6. **Historical Reference**: Permanent record of each test run

Next Steps

Immediate Actions

1. Fix moondream model references (✅ Done - replaced with llava-llama3:8b)
2. Install audio framework for audio model testing
3. Review models with response pattern mismatches

Future Enhancements

1. Add log rotation for old test runs
2. Create log analysis tools
3. Add performance metrics (response times)
4. Generate trend reports across multiple runs

Conclusion

The test framework now provides comprehensive visibility into all aspects of model testing through detailed logging. This enables:

- Better debugging of test failures
- Historical tracking of model performance
- Clear documentation of issues and fixes
- Complete audit trail for compliance

The logging system successfully captures all 33 model tests across 9 categories with detailed information about each test execution.