Decoupling Perception from Reasoning

A Modular Framework for Construction Video Analysis

Systematic Evaluation of Hybrid CV-LLM Architectures for Automated Activity Recognition

AI Showcase Submission
December 2025

Abstract

We present a modular video analysis framework for automated construction site monitoring and systematically evaluate nine distinct strategies spanning pure computer vision (CV), pure large language model (LLM), and hybrid approaches. Our experiments reveal that while no single architecture achieves fine-grained activity recognition, architectural choices significantly impact the accuracy-speed tradeoff. The best-performing strategy (`raw_llm_sonnet`) achieved 50.0% State F1 and 47.4% Object F1 but failed to meet real-time requirements (2.89x speed ratio). Conversely, the hybrid strategy `prompt2_recheck_haiku` demonstrated that confidence-based fallback logic can allow smaller models to approach larger model performance while maintaining real-time capability (0.99x). We identify the CV component as the primary system bottleneck, with the visual-only baseline performing near random chance. Despite limitations in fine-grained classification, the system achieved 80.5% F1 for aggregate tool-use detection, demonstrating immediate viability for productivity tracking. The modular architecture enables rapid experimentation across model types and provides infrastructure for continued optimization as foundation model capabilities improve.

Contents

1 Introduction

1.1 The Productivity Challenge in Construction

Construction productivity has grown slowly relative to other industries, with measurement difficulties contributing to this gap. The industry metric "Wrench Time"—the percentage of time workers actively use tools—averages around 35% across the industry, but measuring it currently requires human observers, an approach that is costly, subject to observer bias, and difficult to scale.

Automated video analysis offers a potential alternative. Many construction sites already deploy camera networks for safety and security, and some companies now offer egocentric recording devices for workers. Repurposing this infrastructure for productivity measurement is appealing, but as we demonstrate, current AI approaches exhibit systematic limitations that must be understood before deployment.

1.2 The Promise and Limitations of AI

Two paradigms dominate video understanding:

1. **Computer Vision (CV)**: Traditional object detection and motion analysis. Fast, deterministic, but limited to recognizing patterns seen during training.
2. **Large Language Models (LLMs)**: Vision-language models that bring semantic reasoning to visual analysis. Flexible, capable of zero-shot recognition, but computationally expensive and prone to hallucination.

Hybrid approaches attempt to combine CV's efficiency with LLM's reasoning capability. Our work systematically evaluates these strategies to understand their complementary strengths and irreducible limitations.

1.3 Contributions

This paper makes two main contributions:

1. **Exploratory Evaluation**: We design and benchmark nine strategies across pure CV, pure LLM, and hybrid approaches, revealing how combining LLMs with modular "enhancements" can selectively improve accuracy and performance in a way complementary to improving LLM capabilities.
2. **Modular Architecture**: We developed an extensible video processing platform that enabled rapid experimentation. The registry-based design allows new models and decision strategies to be integrated without modifying core code, providing infrastructure for continued optimization.

1.4 Scope and Honest Limitations

We acknowledge upfront: end-to-end multimodal models will continue improving. Our contribution is pragmatic—given current constraints (cost, latency, accuracy), we quantify the value of architectural choices and establish benchmarks for comparison. The modular design ensures the system can adopt superior models as they emerge.

Our evaluation dataset is small (5 videos), limiting statistical generalization. Results should be interpreted as diagnostic rather than definitive. We present this work as an exploration of the problem space rather than a solved solution.

2 System Architecture

2.1 Design Philosophy

The architecture (accessible at https://github.com/zachmacsmith/Spatial-AI) is built on three principles:

2.1.1 Model Agnosticism

The system abstracts AI providers (Anthropic Claude, Google Gemini, OpenAI GPT) or CV models behind a common interface. This design:

- Mitigates vendor lock-in risk
- Enables cost optimization by selecting providers dynamically
- Ensures rapid adoption of new models without code changes

Switching from Claude to Gemini requires a single configuration change—no code modifications.

2.1.2 Algorithmic Modularity

Decision logic is decoupled from the processing pipeline via a **Registry Pattern**. New strategies are added via decorator:

```
@register_state_check("safety_sensitive")
def custom_state_logic(context):
    # Custom classification logic
    return predicted_state
```

This allows data scientists to inject new classification strategies, safety rules, or tool-specific detectors without modifying core infrastructure or risking system stability. This facilitates easy customization of individual decision strategies that use different combinations of CV, LLMs or other data to classify construction worker tool-use, but also the implementation of more sophisticated workflow choices about how the video should be broken down and passed to these various modular sub-models.

2.1.3 Adjustable Presets

Configurable "Presets" allow operators to rapidly test and compare different models to prototype quickly or optimize for different situations. For example:

- **Model usage:** Our presets compare Sonnet 4.5 and Haiku 4.5.
- **Granularity:** Presets can sample videos with different frequency, even adapting responsively as identifications are made.
- **Data Incorporation:** Presets can incorporate any modular form of video analysis including LLMs, CV and pixel-motion calculations.
- **Batch Analysis:** Supports rapid batch-testing and comparative benchmarking with over 80 adjustable parameters.

2.2 Pipeline Overview

The system processes video through three phases:

1. **Perception Layer**: CV model extracts motion scores, bounding boxes, and basic object candidates from keyframes.
2. **Reasoning Layer**: LLM receives frames plus CV context and applies semantic understanding to classify activities and identify tools.
3. **Decision Layer**: Pluggable decision functions combine CV and LLM outputs according to configurable logic, producing final classifications.

Importantly, each layer does not have to use LLM, CV or either. Some of our presets also incorporate the perception of motion through pixel difference calculations. The decision layer can be adjusted to incorporate any combination of perception and reasoning.

2.3 Extensibility Features

The architecture provides several extension points:

Table 1: System Extension Points

Component	Purpose	Extension Method
State Classifiers	Idle/Moving/Using Tool	`@register_state_check`
Object Detectors	Tool identification	`@register_object_check`
LLM Providers	AI model integration	Implement `LLMService` interface
CV Models	Vision preprocessing	Implement `CVService` interface
Processing Strategies	Frame selection logic	`@register_processing_strategy`
Prompt Templates	LLM instruction sets	Add to `PromptBuilder`

This modularity allowed us to develop and test more than nine strategies in days.

3 Methodology

3.1 Dataset

We evaluated on a dataset of five construction site videos:

- Duration: 3–5 minutes each
- Resolution: Standard HD (1080p)
- Activities: Movement, idle periods and diverse tool use.
- Tools present: Hammer, measuring tape, nail gun, saw, drill, pencil, wood, caulk gun

Ground truth annotations were created manually, labeling each frame with:

- **State**: idle, moving, or using tool
- **Object**: The tool being used (or "unknown" if no clear tool)

Limitation: Dataset size limits statistical generalization. We report results as diagnostic indicators rather than production benchmarks.

3.2 Evaluation Metrics

3.2.1 Macro F1 Score

Since the data is not distributed equally between states and tool use, we use Macro F1 as the primary metric. This was computed as the average of per-class F1 scores, unweighted to accentuate the importance of successfully identifying rarer states and tool use:

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \frac{2 \cdot P_c \cdot R_c}{P_c + R_c} \tag{1}$$

where P_c and R_c are precision and recall for class c. Macro F1 weights all classes equally regardless of frequency, exposing performance on minority classes. This was evaluated against the three states (idle, moving and using tool) and objects are present within the specific video being examined.

3.2.2 Confusion Matrices

We analyze recall-normalized confusion matrices to understand *where* predictions go wrong, not just *that* they are wrong. This reveals systematic biases (attractor classes) in each model, which was especially useful in highlighting specific tools that are commonly confused.

3.2.3 Speed Ratio

Processing time divided by video duration. Values below 1.0 indicate real-time capability:

$$\text{Speed Ratio} = \frac{\text{Processing Time}}{\text{Video Duration}} \tag{2}$$

3.3 Strategies Evaluated

Table 2: Evaluated Strategies

Strategy	Architecture	Key Characteristics
RAW_LLM_SONNET	LLM only	Direct frame analysis, no CV
VISUAL_HEURISTIC	CV only	Motion + basic detection, no LLM
BASIC_COMBINED	Hybrid	CV preprocessing + strict LLM reasoning
INTERVAL_AGGREGATION	Hybrid	Temporal windowing over multiple frames
PROMPT2_HAIKU	Hybrid (Haiku)	Smaller LLM with CV preprocessing
PROMPT2_SONNET	Hybrid (Sonnet)	Larger LLM with CV preprocessing
PROMPT2_RECHECK_HAIKU	Confidence hybrid	LLM first, CV fallback on low confidence
BASELINE_MAJORITY	Baseline	Always predicts majority class
BASELINE_RANDOM	Baseline	Random prediction

4 Results and Analysis

4.1 Overall Performance

Table 3: Performance Summary (Sorted by State F1)

Strategy	State F1	Object F1	Speed	Realtime?
RAW_LLM_SONNET	50.0%	47.4%	2.89x	× No
BASIC_COMBINED	45.7%	38.0%	1.74x	× No
INTERVAL_AGGREGATION	45.7%	37.3%	3.12x	× No
PROMPT2_SONNET	44.7%	46.9%	3.65x	× No
PROMPT2_RECHECK_HAIKU	40.6%	44.2%	0.99x	✓Yes
PROMPT2_HAIKU	31.8%	26.5%	0.82x	✓Yes
BASELINE_RANDOM	30.1%	4.2%	0.30x	✓Yes
BASELINE_MAJORITY	27.0%	42.0%	0.28x	✓Yes
VISUAL_HEURISTIC	24.2%	24.2%	0.27x	✓Yes

4.1.1 Key finding (1): Enhancements can improve model performance significantly

PROMPT2_RECHECK_HAIKU significantly outperforms PROMPT2_HAIKU by 27% and 66% while only decreasing speed by 20% (still running faster than realtime). While it is still less accurate than PROMPT2_SONNET, it runs nearly 4x faster. This reveals that adaptive, multi-model architectures can increase both accuracy and performance.

4.1.2 Key Finding (2): Not all enhancements are helpful

BASIC_COMBINED and INTERVAL_AGGREGATION achieved nearly identical performance (∼46% State F1, ∼37% Object F1) while the second takes 1.8x longer. Rather than undermining all possible enhancements, this reasserts the importance of a modular, scalable framework to rapidly explore and compare these possible architectures.

4.1.3 Key Finding (3): CV-model is key bottleneck

VISUAL_HEURISTIC achieved only 24.2% State F1, which is barely above the random baseline (30.1%). While the LLMs used are near state-of-the-art, our CV model was trained on a limited database of 500 images of tools. Importantly, this suggests the strength of our CV model was a key bottleneck, likely explaining why models like BASIC_COMBINED and INTERVAL_AGGREGATION did not perform better. This becomes more clear later when the attractors and blind spots of each model are identified.

4.1.4 Key Finding (4): Enhancements do not replace LLM improvements

Even with enhancements, no strategies with Haiku outperformed those using Sonnet. Adding external enhancements to multi-modal models should not replace progress on these foundation models, but complement them. However, these enhancements did allow these smaller models to approach the performance of larger models, with significantly better performance. The modular approach of this architecture means new models can be almost instantly incorporated upon release, meaning this result actually bodes well that base-model performance still exists as one direction of improvement adjacent to external architectural changes.

4.1.5 Key Finding (5): There is significant room for progress

Even the best performing models only achieve 50% for state and object accuracy. These are initial findings only, and more comprehensive testing should be undertaken.

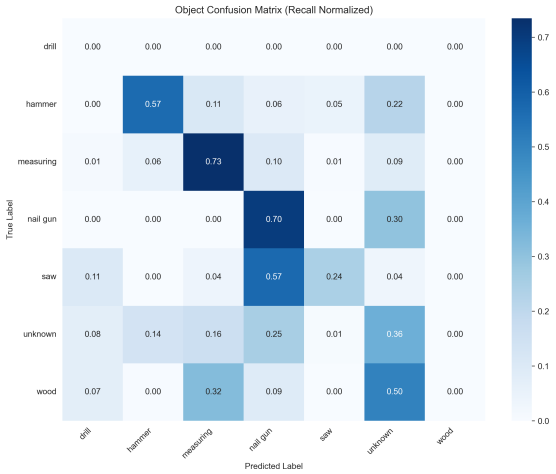

Figure 1: Confusion heat map for PROMPT2_SONNET

4.2 Isolated Failures

Significant gaps in model accuracy can be attributed to key, systemic and identifiable weaknesses.

4.2.1 Identifying Saw

As example, PROMPT2_SONNET above confuses 67% of saw images with a nailgun. This isolated but repeated confusion is exactly where CV enhancements have the potential to be so effective. Increasing the distinction between similar tools, such as nail guns or a drill (other commonly mistaken tools), can be achieved by training the underlying CV model on a more diverse dataset of the desired tools.

4.2.2 Target Modular Improvements

A modular architecture means specific weaknesses of a model can be pinpointed and addressed (as above), without having to risk disrupting the more complex internal workings of a pure LLM. Enhancing functions and processes can target the weaknesses of the current LLMS much faster than it takes to train new multi-modal models, with more precision than just finetuning.

4.3 A surprisingly short path to usefulness

While current strategies only achieve <50% on macro F-1 for identifying states or tools, this actually deflates the general usefulness of these models. This metric artificially inflates the influence of rare events or tools, which helps to identify with more granularity the more niche strengths of each strategy. However, they are significantly more viable for more immediately likely use cases, such as approximating wrench-time (the percentage of time spent using any tool) or identifying common tasks like measurement.

Table 4: Fastest "Wrench time" identification

Strategy	**Using tool F1**	**Speed**
BASIC_COMBINED	80.5 %	1.74x
INTERVAL_AGGREGATION	80.5 %	3.12x
RAW_LLM_SONNET	76.9 %	2.89x

For this use case, the combined approach of LLM and CV data is clearly best. Correctly identifying tool-use ("wrench time") 80% of the time at only 1.74x time is non-trivial, and this was achieved with Claude Haiku and a CV model with significant limitations.

Next Steps The limitations of this research mean that the potential of this modular architecture has still not be fully explored

- Dataset limitations
  - The dataset used to train the YOLO had many rudimentary objects like hands, measuring tape, which it learnt well.
  - Training a future YOLO model with more diverse photos of complex, similar-looking tool like drills, saws and nail guns.
  - Expanding into new construction domains only needs select photo sets of key new tools.
- Rapid Iteration
  - The time constraints of this research meant only nine final presets were explored.
  - Further comparisons with other model providers, such as Gemini or ChatGPT.
- Self-adjusting frameworks
  - the PROMPT2_RECHECK provided unique benefits by selectively sampling more data based on LLM confidence.
  - Further explorations of propagating and analysing model confidence could provide more insights for internal and user-level analysis.

The benefit of the modular architecture is clear in facilitating fast development, benchmarking and diagnostic clarity which means any of these discussed future extensions would require minimal additional time.

==========

5 Conclusion

We presented a modular framework for construction video analysis and systematically evaluated nine strategies spanning pure CV, pure LLM, and hybrid approaches. Our contribution is not a single solved model, but a pragmatic architecture that allows for the rapid benchmarking and integration of emerging AI capabilities.

Our experiments revealed that while no single strategy currently solves the problem of fine-grained activity recognition, the modular approach exposes the specific mechanical levers required to get there. While the best overall performance was achieved by RAW_LLM_SONNET (50.0% State F1), it failed to meet real-time requirements. Conversely, our hybrid strategy PROMPT2_RECHECK_HAIKU demonstrated that intelligent architectural choices—specifically using confidence-based fallback logic—can allow smaller, faster models to approach the utility of larger foundation models while maintaining real-time speeds (0.99x).

Critically, we identified the current Computer Vision component as the primary system bottleneck. With the VISUAL_HEURISTIC baseline performing near random chance (24.2% F1), it is clear that the hybrid models were limited by the specific training data of the CV module rather than the architecture itself. This validates the modular design: specific weaknesses, such as the consistent confusion between saws and nail guns, can be addressed by swapping or retraining the object detector without disrupting the semantic reasoning layer.

Despite the limitations in fine-grained classification, the system demonstrated immediate industrial viability for high-level productivity tracking. Strategies like BASIC_COMBINED achieved an 80.5% F1 score for identifying general tool use ("wrench time"), proving that current technology is sufficient for aggregate productivity analysis even if specific tool identification requires further refinement.