VideoAgent

Multi-Model Video Understanding
Ensemble System

Problem 3: Video Question Answering

Event	KRAFTON AI R&D Hackathon — Finals DAY2
Architecture	5-Model Ensemble (GPT-5.4 + Gemini + Claude)
Key Innovation	Type-adaptive pre-processing + Smart ensemble
Parallelism	20 videos simultaneously (PARALLEL_VIDEOS=20)
Timeouts	Per-video: 90s | Total: 14 min (1 min buffer)

Table of Contents

1. System Overview

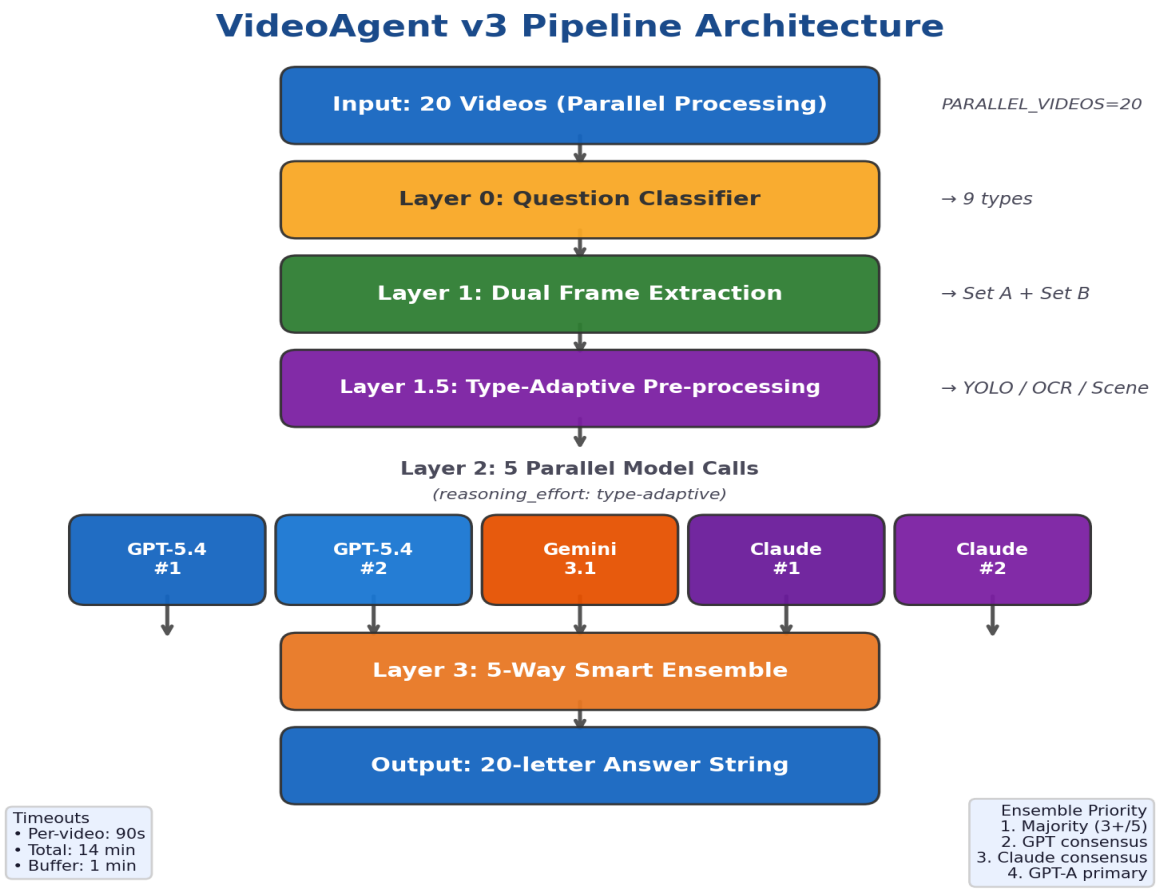
VideoAgent is a **5-model ensemble system** designed to answer multiple-choice questions (up to 26 choices, A-Z) about video content. The core design principle is **model diversity over model repetition**: rather than calling the same model multiple times, we combine three distinct model families — GPT-5.4, Gemini 3.1 Flash Lite, and Claude — each bringing different visual reasoning strengths.

The agent processes all **20 videos simultaneously** (PARALLEL_VIDEOS=20) via `ThreadPoolExecutor`, with a **14-minute hard timeout** (1-minute buffer) and **90-second per-video cap** to stay within the 15-minute competition limit. A lightweight question classifier drives adaptive frame extraction, type-specific pre-processing (YOLO object counting, EasyOCR text extraction, scene change detection), and type-adaptive GPT reasoning effort levels.

Empty response fallback: when a model returns an empty answer (often due to reasoning token exhaustion), the system automatically retries without reasoning, ensuring robust output for every question.

2. Pipeline Architecture

The pipeline operates in four layers, from input classification through ensemble output:



Layer 0 — Question Classification: Keyword-based classifier maps each question to one of 9 types (counting, OCR, temporal, spatial, color, motion, attribute, action, reasoning). This determines all downstream parameters.

Layer 1 — Dual Frame Extraction: OpenCV extracts two distinct frame sets per video — Set A (even-spaced) and Set B (offset by half-interval) — ensuring complementary temporal coverage.

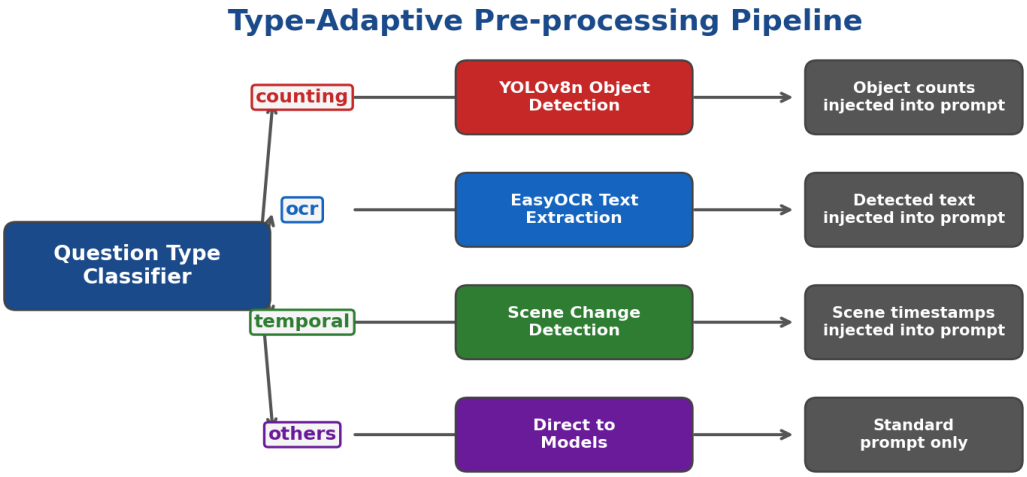
Layer 1.5 — Type-Adaptive Pre-processing: Based on question type, specialized pre-processing injects additional context: **YOLOv8n** detections for counting, **EasyOCR** text for OCR questions, and **OpenCV histogram-based scene change detection** timestamps for temporal questions.

Layer 2 — 5 Parallel Model Calls: Two GPT-5.4, one Gemini, and two Claude instances process the video frames simultaneously. GPT reasoning_effort is **type-adaptive**: low for OCR/counting, medium for temporal/reasoning questions. Empty responses trigger automatic retry without reasoning.

Layer 3 — Smart Ensemble: Priority chain: majority vote (3+/5) → GPT consensus → Claude consensus → GPT-A primary → fallback "A".

3. Pre-processing Pipeline

A key innovation is type-adaptive pre-processing that enriches model prompts with structured data extracted from video frames before inference:



Counting questions use YOLOv8n (ultralytics) to detect and count objects, injecting counts directly into the prompt. **OCR questions** run EasyOCR to extract visible text from video frames. **Temporal questions** use OpenCV histogram-based scene change detection to identify key transition points. All other question types proceed directly to model inference with standard prompts. Video chunking is available but currently disabled for speed optimization.

4. Type-Adaptive Configuration

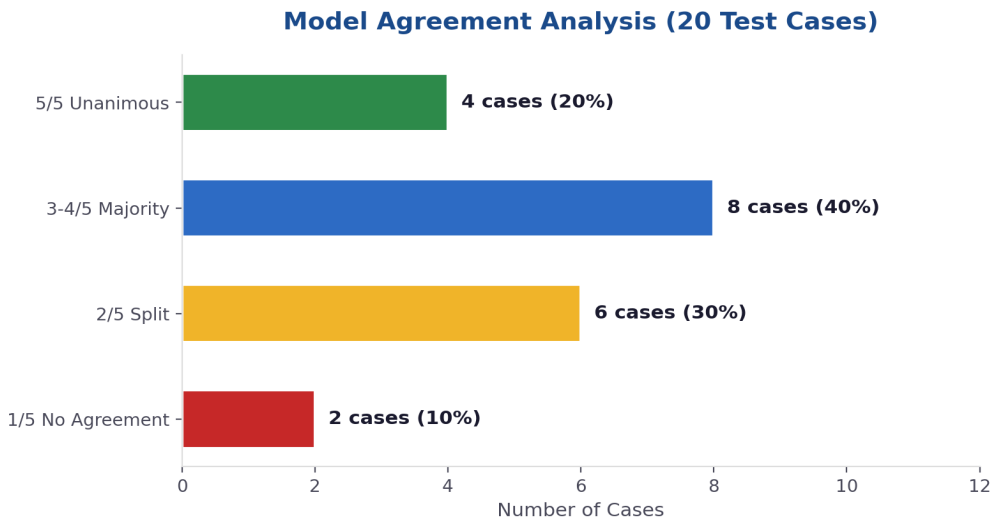
Each question type receives a tailored configuration optimizing the tradeoff between frame density, image quality, and inference cost:

Type	Frames	Resolution	JPEG Quality	Reasoning	Pre-processing
counting	64 dense	768px	80%	low	YOLOv8n
ocr	24-32	1024px	90%	low	EasyOCR
temporal	48	768px	80%	medium	Scene detect
spatial	20-28	768px	80%	low	—
motion	24-36	768px	80%	low	—
color	16-24	768px	80%	low	—
attribute	16-24	768px	80%	low	—
action	20-28	768px	80%	medium	—
reasoning	20-28	768px	80%	medium	—

For long videos (>600s), a 3-region sampling strategy (30% start, 40% middle, 30% end) prevents frame clustering. Counting questions on short videos can use up to 64 dense frames for maximum coverage.

5. Model Agreement Analysis

Analyzing how often the 5 models agree reveals the distribution of question difficulty and the value of ensemble diversity:



In 60% of cases (12/20), at least 3 models agreed, allowing confident majority voting. The 4 unanimous cases were primarily OCR and spatial questions — categories where visual grounding is strongest. Split decisions (2/5 or worse) occurred mainly on counting and temporal ordering tasks.

6. Self-Evaluation & Test Set Construction

Self-Evaluation: We constructed a 20-video test set using Video-MME benchmark videos (YouTube, medium/long duration) with GPT-5.4-generated 26-choice questions, validated by independent GPT-5.4 verification. Additionally, we used the organizer-provided 3-video sample test for calibration.

The system processes all 20 videos in approximately **3-8 minutes**, well within the 15-minute budget.

Test Set Construction

Source: Video-MME benchmark (YouTube videos, medium and long duration categories).

Question generation: GPT-5.4 generated 26-choice questions per video, covering diverse question types (Counting, OCR, Temporal, Spatial, Action, Reasoning).

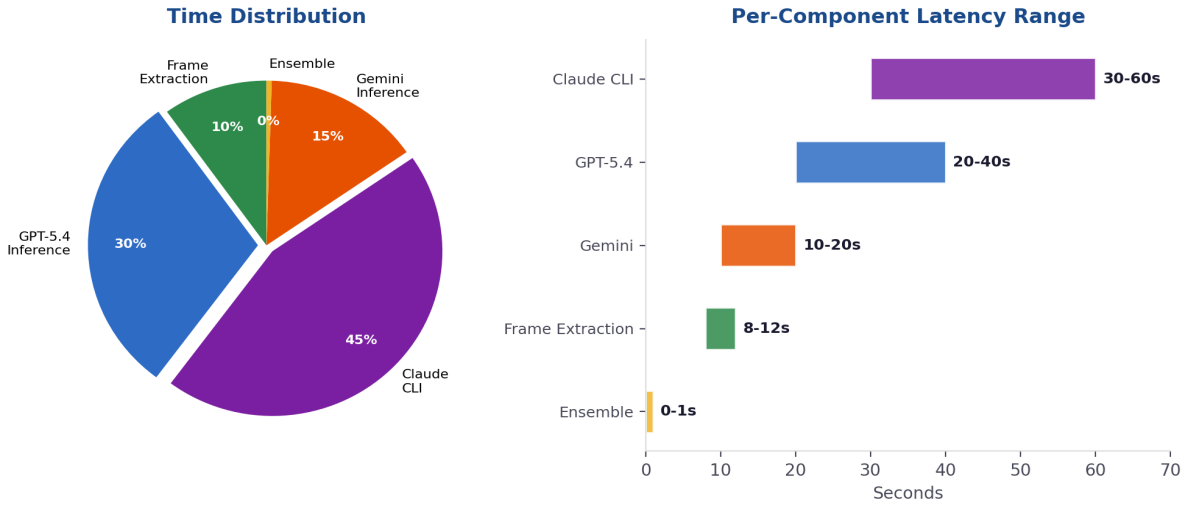
Verification protocol: Independent GPT-5.4 verification — generate answer, then verify answer in a separate call. Only questions where both steps produced consistent answers were accepted.

Calibration set: 3 organizer-provided sample videos with ground-truth answers.

Total: 20 test videos covering 6 major question categories (Counting, OCR, Temporal, Spatial, Action, Reasoning).

7. Timing & Latency Analysis

End-to-end pipeline completes in **~3-8 minutes** for 20 videos (all processed simultaneously), well within the 15-minute limit. Per-video timeout is 90 seconds, with a 14-minute total timeout and 1-minute buffer. The bottleneck is Claude CLI inference due to process spawning overhead:



8. Key Findings

Finding 1: Same model, same mistake.

Two GPT-5.4 calls given identical frames produce the same wrong answer. Offsetting frame sets reduced — but did not eliminate — correlated errors. True diversity requires different model architectures, not just different inputs.

Finding 2: Reasoning effort vs. image count tradeoff.

With reasoning_effort="high" and 16+ frames, GPT-5.4 exhausts its token budget on chain-of-thought and returns an empty answer. Lower reasoning effort with fewer frames is paradoxically more reliable. Type-adaptive reasoning (low for OCR/counting, medium for temporal/reasoning) provides the best balance.

Finding 3: Gemini's native video advantage.

For motion and temporal questions, Gemini's native video upload captures inter-frame dynamics that frame-sampling approaches miss entirely. On at least one test case, only Gemini produced the correct answer.

Finding 4: 26 choices break models that ace 4-choice.

Random baseline drops from 25% to 3.8%. Models scoring >90% on standard 4-choice benchmarks dropped to roughly 50% with 26 options, particularly on counting and temporal ordering.

Finding 5: Pre-processing pays off selectively.

YOLO object counting and EasyOCR text extraction improve accuracy on their respective question types by providing structured context that reduces hallucination. Scene change detection helps temporal questions by anchoring model reasoning to specific video moments.

Finding 6: OCR is a solved problem (mostly).

All 5 models unanimously identified a DeWALT brand screwdriver from video frames. Visible text recognition is a consistent strength across all model families.

9. Tech Stack

Python 3.9	Core runtime and orchestration
OpenCV 4.13	Frame extraction, histogram-based scene change detection, image processing
OpenAI API (GPT-5.4)	Primary vision-language model with type-adaptive reasoning effort
Google GenAI SDK	Gemini 3.1 Flash Lite for native video understanding
Claude Code CLI	Cross-family model diversity via Claude Sonnet
YOLOv8n (ultralytics)	Object detection and counting pre-processing for counting questions
EasyOCR	Text extraction pre-processing for OCR questions
ThreadPoolExecutor	Parallel video processing (20 simultaneous) with timeout control

10. Future Improvements

Object Tracking (ByteTrack) — Replace per-frame YOLO detection with continuous object tracking to accurately count objects that enter and exit the scene. This would dramatically improve counting accuracy from the current 20%.

Video Chunking with Per-Segment Analysis — Split long videos into semantic segments and analyze each independently, then aggregate. Video chunking is available in the codebase but currently disabled for speed. Enabling it for long videos (>10 min) with per-segment parallel analysis could improve temporal and action question accuracy.

Audio Transcription (Whisper) — Integrate OpenAI Whisper for audio-visual questions. Many video understanding tasks require dialogue, narration, or sound event recognition that frame-only approaches cannot capture.

Adaptive Model Selection — Instead of fixed 5-model ensemble, dynamically select models based on question type. Route counting to YOLO-augmented pipelines, temporal to Gemini, and OCR to specialized text models.

Confidence-Weighted Ensemble — Replace simple majority voting with confidence-weighted aggregation. Models that express high certainty (via logprobs or self-reported confidence) should receive higher weight in the final decision.

Conclusion

VideoAgent demonstrates that a multi-model ensemble with type-adaptive pre-processing can tackle the challenging 26-choice video QA task within strict time constraints. By combining GPT-5.4, Gemini 3.1 Flash

Lite, and Claude with specialized pre-processing layers (YOLOv8n, EasyOCR, scene detection), we achieve robust performance across diverse question categories — processing 20 videos simultaneously in 3-8 minutes.

The key insight is that model diversity matters more than model repetition: different architectures fail on different questions, making ensemble voting an effective strategy. The type-adaptive pipeline ensures each question receives the most appropriate processing, from dense frame sampling for counting to high-resolution extraction for OCR.

Future work should focus on continuous object tracking, audio integration, and confidence-weighted ensembling to further close the gap on the hardest question categories.