**Title:**

Leveraging Process Mining Videos for AI Worker Agents: Feasibility, Implementation Guide, Challenges, and Cost Analysis

**Author:**

**Date:**

July 19, 2025

**Executive Summary**

This white paper explores the feasibility of transforming process mining videos into structured data for fine-tuning large language models (LLMs), ultimately enabling the creation of AI worker agents. By converting video content into text, markup, or JSON workflows, organizations can develop specialized AI agents that automate and optimize business processes. The approach is technically viable with current AI tools, offering 70-90% accuracy in workflow recognition. We provide a step-by-step guide, address potential challenges, and estimate costs ranging from $50 to $1,000 for initial implementation. This methodology has the potential to significantly reduce process analysis time and enhance operational efficiency.

**Introduction**

In today's data-driven business environment, process mining videos serve as valuable resources for documenting workflows, steps, and events. However, extracting actionable insights from these videos manually is time-consuming. This white paper proposes an innovative approach: converting video content into structured formats to fine-tune LLMs, thereby creating AI worker agents capable of simulating, executing, and optimizing processes. We assess the feasibility, outline implementation steps, discuss challenges, and provide cost estimates to guide organizations in adopting this technology.

**Section 1: Feasibility Assessment**

Yes, this is entirely feasible with current AI technologies. Process mining videos typically contain demonstrations of workflows, steps, and events, which can be converted into structured data (text, JSON, or markup) for fine-tuning large language models (LLMs). Fine-tuning an LLM on this data can create a specialized model that understands your company's specific processes, enabling the development of AI worker agents. These agents can automate tasks, simulate workflows, or assist in process optimization.

**Key Enablers:**

* **Video-to-Data Conversion:** Tools like automatic speech recognition (ASR) and video analysis can transcribe and extract meaningful content.
* **Workflow Extraction:** LLMs can parse transcripts into structured formats, building on AI-driven process mining tools.
* **Fine-Tuning:** Platforms support tuning LLMs on text or even video inputs, improving performance on domain-specific tasks.
* **AI Agents:** Frameworks allow building autonomous agents that use the fine-tuned LLM to execute or monitor workflows.

Success rates depend on video quality, data volume (aim for 100+ high-quality examples for fine-tuning), and computational resources. Challenges include handling visual elements (beyond audio), ensuring data privacy, and model hallucination. Overall, this approach can achieve 70-90% accuracy in workflow recognition with proper setup, based on similar applications in video summarization and process automation. Costs could range from free/open-source tools to $1,000+ for cloud-based fine-tuning, but expect iterative testing.

**Section 2: Step-by-Step Implementation Guide**

Here's a practical roadmap, divided into phases. Focus on open-source or accessible tools where possible, assuming basic programming knowledge (e.g., Python).

**Phase 1: Convert Videos to Structured Data**

Videos need to be transcribed and analyzed to extract workflows. This creates text/markup/JSON for fine-tuning.

1. **Transcribe Audio and Describe Visual:**
   * Use ASR tools like OpenAI's Whisper (open-source) or Google Cloud Speech-to-Text to convert video audio to text transcripts.
   * For visuals (e.g., diagrams, steps shown on screen), sample frames every 5-10 seconds using libraries like OpenCV or FFmpeg, then describe them with multi-modal models like GPT-4V or Gemini (e.g., "Frame at 00:30 shows a flowchart with steps: Input data — Process — Output").
   * Alternative: Use integrated tools like Dumpling AI with Make.com for automated extraction. Steps:
     + Monitor a folder (e.g., Google Drive) for video uploads.
     + Download and process the video with Dumpling AI to extract transcripts or key data (e.g., via a prompt: "Extract all process steps from this video").
     + Output to JSON: {"step1": "Receive input", "step2": "Analyze data", ...}.
2. **Extract Workflows:**
   * Feed transcripts into an LLM (e.g., GPT-4 or Llama 3) with a prompt like: "Convert this transcript into a JSON workflow: Identify sequential steps, decision points, and outputs."
   * Use process mining libraries like PM4Py (Python) to model workflows as Petri nets or BPMN markup, if videos include event logs.
   * For AI enhancement: Integrate with tools like Pega or Appian for AI-driven mining, which analyzes data to uncover hidden workflows.
   * This could yield structured output like:

text

CollapseWrap

Copy

{

"workflow": "Order Processing",

"steps": [

{"id": 1, "action": "Receive order", "input": "Customer data"},

{"id": 2, "action": "Validate inventory", "decision": "If available, proceed; else, notify"}

]

}

* + Aim for 50-100 videos to build a robust dataset.

**Phase 2: Fine-Tune the LLM**

Use the extracted data to specialize an LLM on your processes.

1. **Prepare the Dataset:**
   * Format as instruction-response pairs: Input (transcript/description), Output (workflow JSON/markup).
   * Example: Input: "Transcript: [full text from video]", Output: "[JSON workflow]".
   * Ensure 100-1,000 examples; use tools like Hugging Face Datasets for management.
   * If videos are key, use multi-modal fine-tuning (e.g., include frame descriptions).
2. **Perform Fine-Tuning:**
   * **Text-based:** Use Hugging Face Transformers with models like Mistral-7B or Llama 3. Techniques: LoRA/QLORA for efficiency (reduces compute needs).
   * **Video-based:** Use Google Vertex AI for Gemini 2.5 Flash, which supports direct video inputs. Prerequisites: Videos <100MB, 5-20 min length. Steps:
     + Upload videos to Google Cloud Storage.
     + Create JSONL dataset: {"user": {"fileData": {"fileUri": "gs://path/video.mp4"}, "text": "Extract workflow from this process mining video"}, "model": {"text": "[JSON output]"}}.
     + Tune with supervised fine-tuning; use low/medium resolution for frames.
     + Use cases: Event recognition (e.g., detect workflow steps) or summarization (condense processes).
   * Platforms: OpenAI Fine-Tuning API, AWS SageMaker, or NVIDIA NeMo for advanced setups. Train for 1-5 epochs; evaluate on held-out videos.

**Phase 3: Build AI Worker Agents**

Integrate the fine-tuned LLM into agents that act on workflows.

1. **Choose a Framework:**
   * AutoGen: For multi-agent systems. LangChain/LangGraph: For chaining workflows.
   * n8n or Botpress: No-code for automation.
2. **Implement Agents:**
   * Use the fine-tuned LLM as the core (e.g., replace base model in AutoGen).
   * Steps with AutoGen:
     + Set up environment: Install AutoGen, OpenAI, etc.; configure API keys.
     + Define agents: e.g., PlannerAgent (uses fine-tuned LLM for workflow planning), ExecutorAgent (runs steps).
     + Add tools: Web search, code execution, or custom (e.g., "simulate\_workflow" function).
     + Add memory: Short-term (conversation buffer) for context, long-term (vector store like ChromaDB) for past workflows.
     + Example: Agent prompt: "You are a process mining agent. Given a query, use the fine-tuned model to generate and execute a workflow."
     + Run: Initiate with user input (e.g., "Automate order processing"), let agents collaborate autonomously.
   * Design patterns: Chaining (sequential steps), routing (decision-based), parallel (multi-task), evaluator-optimizer (refine outputs).
   * For worker agents: Make them autonomous (e.g., monitor systems, execute processes like "If inventory low, reorder").
3. **Test and Deploy:**
   * Simulate workflows on test data.
   * Deploy via APIs (e.g., FastAPI) or cloud (AWS Lambda, Google Cloud Run).
   * Monitor: Use logging to track agent decisions.

**Section 3: Potential Challenges and Mitigations**

* **Data Quality:** Noisy transcripts — Clean with post-processing (e.g., spaCy for NLP).
* **Compute Costs:** Fine-tuning is expensive — Use efficient methods like LoRA; start with smaller models.
* **Multi-Modal Gaps:** Videos have visuals — Combine ASR with image models (e.g., CLIP).
* **Ethical/Legal:** Fine-tuning on company data — Ensure compliance; use private clouds.
* **Scalability:** For real-time, chunk long videos or use RAG instead of full fine-tuning for agents.

**Section 4: Cost Estimates**

Below is a breakdown of underlying cost estimates for the process. These are approximate figures based on current (July 2025) pricing from major providers. Costs can vary based on scale (e.g., number of videos, dataset size, training duration), region, usage volume, and discounts. Assumed a small-scale project: 100 videos (each ~10 minutes), resulting in ~1,000 minutes of audio/video processing, a fine-tuning dataset of ~1-5 million tokens, and 1-5 hours of training time. For self-hosted options, assume 4-8 GPUs for fine-tuning.

* Free tiers: Many services offer initial credits (e.g., Google Cloud $300 free, AWS up to $200).
* Variables: Transcription at ~$0.006-0.016/min; fine-tuning ~$100-500 total for small datasets; GPU rental $1-3/hour per GPU.
* Total estimate: $50-1,000 for setup and initial run (excluding ongoing inference/agent usage, which could add $0.01-0.10 per query).

| **Phase** | **Component** | **Tool/Provider Options** | **Estimated Cost (Low-End)** | **Estimated Cost (High-End)** | **Notes/Details** |
| --- | --- | --- | --- | --- | --- |
| Phase 1: Video Conversion (Transcription & Workflow Extraction) | Audio Transcription (ASR) | OpenAI Whisper API | $6 (for 1,000 min at $0.006/min) | $10 (at $0.010/min) | Priced per minute; infrastructure for self-hosted ~$0.07-0.10/hour. Free for small volumes with credits. |
|  |  | Google Cloud Speech-to-Text | Free (first 60 min/month) then ~$16 (for remaining at $0.016/min) | $20 (enhanced models at higher rates) | Billed in 15-second increments; $0.004/15s standard. Up to $300 free credits for new users. |
|  | Visual Frame Description & Workflow JSON Extraction | Google Vertex AI Gemini (multi-modal inference) | $5-10 (for 100 videos, sampling 10 frames each at ~$0.00025/1k chars equivalent) | $20-50 (video input per second, assuming $0.001-0.005/frame) | Inference pricing ~$0.00025/1k input chars; video grounding free up to limits. Tuning preview was free, but now ~$1.50/1,000 pages data upload. |
|  |  | OpenAI GPT-4V/GPT-4.1 (via API) | $10-20 (at $3-5/1M tokens for descriptions) | $50 (higher for video tokens) | Input $3/1M tokens; audio $100/1M tokens (~$0.06/min). Use for prompting transcripts to JSON. |
| Phase 2: LLM Fine-Tuning | Managed Fine-Tuning | OpenAI API (e.g., o4-mini or GPT-4.1) | $100-200 (1-2 hours training at $100/hour) | $500+ (5+ hours or larger models) | Hourly for reinforcement fine-tuning (~$100/hour); token-based for others ($3-25/1M tokens training). Azure variant ~$110/hour. |
|  |  | Google Vertex AI (Gemini tuning) | $50-100 (data upload + tuning) | $200+ (full deployment) | Data upload $1.50/1,000 pages after free tier; tuning costs not explicitly hourly but similar to inference (~$0.75/node hour deployment). Preview was free. |
|  |  | AWS SageMaker | $50-100 (ml.g5 instances at ~$1.41/hour for 10-20 hours) | $300+ (GPU-heavy, e.g., 8xA100 at $32/hour) | Per instance hour; A100 ~$3/hour. Savings plans reduce 30-50%. Storage $0.10/GB-month. |
|  | Self-Hosted Fine-Tuning (e.g., Hugging Face Transformers on GPUs) | Hugging Face Inference Endpoints (post-tuning hosting) | $10-20 ($0.033/hour for 300-600 hours uptime) | $50+ (autoscaling GPUs at $0.06-0.60/hour) | Pay-as-you-go; no direct fine-tuning cost, but run on rented GPUs. Pro plan $9/month includes $2 credits. |
|  | GPU Rental for Training (e.g., for LoRA on Mistral-7B) | AWS EC2/Google Cloud (H100/A100) | $20-50 (4 GPUs x 2 hours at $1-3/hour/GPU) | $200+ (8 GPUs x 5 hours at $8/hour/GPU peak) | H100 ~$2.49/hour (Lambda); A100 ~$3/hour. Vast.ai 3-5x cheaper (~$0.98/hour effective). Reserved instances save 30-50%. |
| Phase 3: Building & Deploying AI Agents | Frameworks & Deployment | AutoGen/LangChain (open-source) | Free (local run) | $10-50 (cloud hosting via Hugging Face or AWS Lambda) | Core frameworks free; add $0.01-0.10 per agent query for LLM calls (e.g., $3-15/1M tokens output). |
|  |  | n8n/Botpress (no-code) | Free tier | $20-100/month (pro features/deployment) | Usage-based; integrate with fine-tuned model endpoints. |
|  | Ongoing Inference (for Agents) | Hosted Endpoints (e.g., Hugging Face/OpenAI) | $5-20/month (low usage at $0.10/1M tokens) | $100+ (high-volume queries) | Per token/output; e.g., OpenAI $4-15/1M output tokens. Deployment ~$0.75/node hour on Vertex AI. |

These estimates are for one-time setup; scale up for production (e.g., multiply by 10 for 1,000 videos). Use cost calculators from providers (e.g., OpenAI, Google Cloud) for precise quotes. Opt for open-source (e.g., Whisper local, Vast.ai GPUs) to minimize costs to under $100 total.

**Conclusion**

This approach can transform your videos into intelligent agents, potentially reducing process analysis time significantly. For hands-on, start with free tools like Google Colab. If you share a sample video URL, I can demo extraction. By adopting this methodology, organizations can unlock new efficiencies in process automation and AI-driven decision-making.