Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

DEEP LEARNING FOR EDUCATIONAL VIDEO ANALYSIS: BENCHMARKING ASR SYSTEMS AND PIPELINE OPTIMIZATION

Gordey Zuev
Faculty of Computer Science
HSE University
Moscow, Russia

Elena Kantonistova
Faculty of Computer Science
HSE University
Moscow, Russia

ABSTRACT

We present a comparative analysis of eight *managed* commercial speech recognition providers (provider-side preprocessing, segmentation, and serving) for educational video transcription and enrichment, evaluated on over 700 lecture recordings (900+ hours) across disciplines. The Fireworks `whisper-v3-turbo` endpoint offers a favorable cost-quality-latency trade-off versus surveyed alternatives. Audio preprocessing reduces billed duration by 10-25% with negligible accuracy loss. Prompt-based "Video Vocabulary" reduces terminology errors without fine-tuning. We implement a parallel pipeline that cuts end-to-end turnaround from over 30 minutes of manual effort per recording to under two minutes, supports up to 50 concurrent jobs, and achieves $\sim 22\times$ speedup at $\approx$ \$0.075 per hour of content for transcription plus pedagogical enrichment (summaries, chapter topics, self-check questions) at list prices. The system is deployed in production.

1 INTRODUCTION

The widespread transition of educational institutions to hybrid and distance learning has led to a substantial increase in recorded lectures (Dhawan, 2020). Universities generate hundreds of hours of video content monthly, mirroring the scale of initiatives that successfully archived over 1,000 lectures annually even two decades ago (Nagataki et al., 2006). While modern conferencing tools have automated recording itself, post-production – trimming, metadata, timestamps, subtitles – remains manual. Based on our experience, this requires at least 30 minutes per recording. This aligns with conservative manual production estimates, which recent work suggests can be reduced by two orders of magnitude through automation (Holmberg, 2025). With 100 recordings monthly, manual effort exceeds 50 person-hours, a burden underscoring the need for automation, particularly as prior university efforts showed even basic recording required significant human resources (e.g., 2 hours per lecture) (Nagataki et al., 2006).

Existing solutions fall into two categories: transcription services and low-code automation platforms. Our analysis of 15 solutions revealed none provide a fully automated processing cycle, while transcription costs vary by more than 60-fold–from roughly \$ 0.06 to \$ 3.8 per hour of recording (order-of-magnitude market range, converted to USD at 79 RUB/USD where vendors quote in RUB).

While prior work has focused either on improving ASR accuracy for educational content (Tobias, 2025; Rao, 2023) or on pipeline automation using commercial tools (Holmberg, 2025), no existing study systematically addresses the full spectrum from cost-optimized transcription to pedagogical content enrichment. We bridge this gap with the first end-to-end platform that enables any educational institution or organization producing video at scale to centrally upload recordings, automatically process them (transcription, trimming, enrichment), and publish to multiple destinations with a single action or on a configured schedule. We demonstrate that production-ready scalability need not compromise on either accuracy or downstream functionality – including automatic summary generation and open-ended question creation, features absent from previous educational video platforms.

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

This paper makes the following key contributions:

- *Benchmark.* Systematic survey of the commercial APIs in Table 1 for Russian and English lecture speech, identifying favorable cost-latency-feature trade-offs under production constraints (not a controlled ablation of isolated model weights).
- *Optimization.* Preprocessing pipeline (silence suppression, noise gating, compression) yielding 10-25% cost reduction.
- *Domain adaptation.* Evaluation of prompt-based Video Vocabulary mechanism that reduces terminology errors without expensive fine-tuning.
- *System.* Production-ready parallel processing platform supporting up to 50 concurrent recordings.
- *Enrichment.* Automated generation of lecture summaries and open-ended questions at marginal additional cost, transforming transcripts into pedagogical resources.
- *User feedback.* Post-deployment questionnaire (n=237) with multiple Likert items on topics, timestamps, summaries, and self-check questions; exploratory descriptive evidence of acceptance (not an RCT).

To guide our investigation, we formulate three research questions:

1. How do transcription APIs compare in cost and quality for Russian lecture content?
2. What preprocessing techniques reduce costs without compromising accuracy?
3. What architecture enables efficient parallel video processing at scale?

All quantitative estimates – processing time, speedup factor, API costs, labor savings – were obtained experimentally on a corpus of over 700 real video recordings (totaling over 900 hours of content) and averaged over the full corpus; methodology is described in Section 5. Our main empirical results are formalized as Empirical Results 1, 2, and 3, and Observation 1. Code and documentation are available at the project repository LEAP (2026).

2 Related Work

Accented and non-native speech remains a known structural challenge for ASR relative to mainstream reference conditions (Hinsvark et al., 2021); we return to speaker effects in Section 6. On the systems side, our deployed platform uses a layered service decomposition with swappable integrations (Fowler, 2002), developed in Section 4.

2.1 Transcription Services and Automation Platforms

Existing solutions can be categorized into two main groups.

(i) Transcription services and APIs (Fireworks AI (Fireworks AI, 2026), OpenAI Whisper (OpenAI, 2026), Google Cloud STT (Google Cloud, 2026), AssemblyAI (AssemblyAI, 2026), Gladia (Gladia, 2026), Deepgram (Deepgram, 2026), ElevenLabs (ElevenLabs, 2026)) address only the speech-to-text conversion stage. Several providers impose file size limitations: OpenAI (25 MB; typical lecture video $\sim$500 MB), Memo AI (0.5-2 GB depending on the plan). No service provides topic extraction with timestamps.

(ii) Low-code automation platforms (Make.com, Zapier, n8n) enable pipeline construction from API calls but lack dedicated CPU workers for video-stream processing (trimming), impose file size limitations (100–500 MB), and require substantial setup and maintenance effort (roughly $ 380–$ 760 one-time development and $ 63–$ 190/month maintenance in our market survey, converted at 79 RUB/USD). Users must additionally provision server infrastructure for video processing.

2.2 Whisper for Educational Content

Recent research has specifically examined the application of OpenAI's Whisper models to educational video transcription. Rao (2023) conducted a preliminary study on 25 academic videos,

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

demonstrating that even the larger Whisper models transcribe content in less than 25% of the playback time – a $4\times$ speedup over human transcription. However, the study also identified key challenges: Whisper tends to generate plausible but incorrect text on inaudible segments, which inflates Word Error Rate (WER) metrics when compared against human transcripts that mark such segments as inaudible.

Tobias (2025) addressed these limitations through domain-specific fine-tuning. By adapting Whisper-small on lecture recordings (including a self-curated 10-hour dataset), the optimized model achieved a WER of 4.53% on unseen educational audio – surpassing Whisper-Medium (5.51%) and Whisper-Large-v2 (5.78%). This work demonstrates that domain adaptation can significantly enhance ASR accuracy for educational content. However, fine-tuning remains computationally expensive and requires curated datasets. Our work explores a lighter alternative: we pair the Whisper architecture family (Radford et al., 2023) with preprocessing optimizations and evaluate prompt-based adaptation via domain-specific vocabulary injection through a *managed* Whisper-compatible API (Fireworks AI's `whisper-v3-turbo` endpoint), achieving accuracy gains at zero training cost.

We deliberately focus on cloud-based API solutions rather than self-hosted checkpoints. Managed endpoints offer low marginal cost per hour of audio and documented batch RTF far below interactive needs, without operating our own GPU fleet for the acoustic model. Self-hosting open weights at our peak load (50+ concurrent recordings) would imply substantial capital and operations cost. Table 2 still matters for orientation: it summarizes RTF ranges, language coverage, and prompting constraints for future model choices. Our platform keeps ASR and LLM calls behind stable orchestration so that swapping providers is mostly configuration work.

The pipeline serves a mixed Russian-English catalog (50+ hours of English processed so far). Multilingual Whisper-style APIs cover both in one integration. Many Russian-centric checkpoints ship as separate per-language models (e.g., Vosk (Vosk, 2026)), which complicates a single uniform stack. Accented and non-native speech also remains structurally harder for ASR than mainstream reference conditions, as summarized above. These are engineering-economic criteria for our benchmark. They do not imply superiority of one open-weight architecture over another outside this deployment context.

2.3 Russian-Language and Optimized ASR Systems

Several open-source Russian-language ASR systems exist, including GigaAM (Sber) (Salute Developers, 2025), T-one (T-Bank) (T-Bank, 2025), Vosk (Vosk, 2026), and community fine-tunes such as bond005/whisper-podlodka-turbo (Bondarenko, 2025) and Vikhrmodels/Borealis (Kuleshov and Nikolich, 2025). Table 2 summarizes representative characteristics. These systems are *not* omitted because they are inherently inferior. Our primary empirical comparison targets managed ASR APIs (fixed per-minute pricing, provider-operated preprocessing, elastic scaling). Self-hosted stacks entail separate hardware sizing, batching policies, and RTF that are not uniquely determined without a dedicated controlled study on our corpus. Video Vocabulary requires a `prompt`-compatible decoder path with whole-request biasing on long recordings. Many Russian checkpoints (GigaAM, T-one, Vosk, Pisets) do not expose an equivalent mechanism in the same form. WhisperX applies the initial prompt only to the first window of the pass (Bain et al., 2023), which is suboptimal for hour-long vocabulary-sensitive lectures. Community Whisper fine-tunes (e.g., podlodka) *do* support prompts but still imply self-hosted serving and monitoring.

Recent work has also produced optimized Whisper-based pipelines: Faster-Whisper (SYSTRAN, 2026) (CTranslate2 port of Whisper reporting large wall-clock reductions versus reference PyTorch inference on GPU under documented conditions (SYSTRAN, 2026)), WhisperX (Bain et al., 2023) adding phoneme-level alignment and VAD-based segmentation, and Pisets (Bondarenko et al., 2025) designed specifically for lecture robustness. These are important engineering references; they are cited here for completeness and to situate our API-centric benchmark. Their RTF varies with hardware and batching; for example, Faster-Whisper documents RTF roughly 0.02-0.08 on an RTX 3070 Ti for large-v2 under specific settings (SYSTRAN, 2026). For our production setting we prioritized managed APIs for predictable cost, multilingual coverage in one vendor integration, and operational simplicity, while acknowledging that a future controlled on-premise evaluation could quantify head-to-head WER under matched audio and hardware.

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

3 Evaluation of Commercial ASR APIs and Self-Hosted Alternatives

3.1 Transcription System Benchmarking

Following common practice in machine-learning-as-a-service evaluation, we distinguish (i) an ASR model in the narrow research sense – a reproducible checkpoint with explicit architecture and weights, under user-controlled pre/post-processing – from (ii) a managed ASR API, i.e., a provider-operated *system* that bundles one or more acoustic models with segmentation, Voice Activity Detection (VAD), resampling, and undisclosed or versioned server-side logic. Our primary benchmark compares items of type (ii) using published tariffs, documented latency/throughput where available, and prompting features. We do not claim to isolate the contribution of the acoustic network from provider infrastructure; self-hosted rows in Table 2 are contextual references, not paired measurements on our 700+ recordings unless stated otherwise.

We surveyed commercial transcription API providers along cost, latency, and feature criteria for Russian and English speech recognition. Tables 1 and 2 summarize commercial API services and representative self-hosted alternatives, respectively.

Table 1: Commercial ASR API services (March 2026).

Provider	Advertised backend	Price ($/min)	RTF / latency	Prompting	Languages
Fireworks AI	Whisper-v3-turbo	0.0009 (Fireworks AI, 2026)	0.00083–0.002 (Fireworks AI, 2026)	`prompt`	99
Fireworks AI	Whisper-v3-large	0.0015 (Fireworks AI, 2026)	0.00125–0.003 (Fireworks AI, 2026)	`prompt`	99
OpenAI	Whisper API	0.006 (OpenAI, 2026)	0.03 (Fireworks AI, 2026)	`prompt`	99
OpenAI	GPT-4o Mini Transcribe	0.003 (OpenAI, 2026)	0.02–0.05	`prompt`	Multilingual
AssemblyAI	Universal-3 Pro	0.0035 (AssemblyAI, 2026)	0.02–0.05	`prompt`, `keyterms` (AssemblyAI, 2026)	6
Deepgram	Nova-3	0.0077 (Deepgram, 2026)	TTFT 200–300 ms (TalkFlow AI, 2025)	`keyterm` (Deepgram, 2026)	36+
Deepgram	Nova-3 (Cloudflare)	0.0052 (Cloudflare, 2026)	TTFT ∼300 ms	`keyterm`	10+ (incl. RU)
Gladia	Async / Real-time	0.006–0.0125 (Gladia, 2026)	TTFT <300 ms (Gladia, 2026)	Custom vocab (beta)	100+
ElevenLabs	Scribe v1	0.0037–0.007 (ElevenLabs, 2026)	0.03–0.1	Keyterms, entities (ElevenLabs, 2026)	90+
Azure	Batch STT	0.006	0.1–0.3	Phrase List	100+
Google Cloud	STT v2	0.016 / 0.003 (batch)	0.05–0.15	PhraseSet (Google Cloud, 2026)	125+

RTF and latency are from provider documentation (not measured on our corpus); Fireworks batch RTF is detailed in Fireworks AI (2026). Uncited cells are indicative. *TTFT* denotes streaming latency to the first transcript output (not LLM text generation) and is not comparable to batch RTF.

Table 2: Representative self-hosted ASR systems (checkpoint + stack). RU WER ranges are *illustrative* and not comparable across rows. Reported hardware for RTF benchmarks is in Table 4.

System	Architecture	RTF	Prompting	Languages	RU WER
GigaAM-v3	Conformer CTC/RNNT	0.05–0.15 (Salute Developers, 2025)	None	Russian	3–10% (Salute Developers, 2025)
T-one	CTC streaming Conformer	<0.01 (stream) (T-Bank, 2025)	None	Russian	6–9%
Vosk	Kaldi	0.2–0.5 (Vosk, 2020)	None	Russian	∼11% (Alphacephei, 2025)
podlodka-turbo	Whisper fine-tune	0.03–0.1 (Bondarenko, 2025)	`prompt`	Russian, English	9–14% (Alphacephei, 2025)
Borealis	Audio LLM (Whisper+Qwen)	0.05–0.2 (Kuleshov and Nikolich, 2025)	LLM instruction	Russian, English	6–16% (Alphacephei, 2025)
Faster-Whisper	CTranslate2 (Whisper)	0.02–0.08 (SYSTRAN, 2026)	`prompt`	99	∼10%
WhisperX	Whisper + alignment	0.1–0.3 (Bain et al., 2023)	`prompt` (init. only)	99+	∼10%
Pisets	Wav2Vec2 + AST + Whisper	0.15–0.25 / 1.0–1.5 (Bondarenko et al., 2025)	None	Russian, English	2–5% (Bondarenko et al., 2025)

For Vosk, "Russian" denotes that vendor's Russian acoustic model; the toolkit supports many languages via separate packs (Vosk, 2026). Encoder-only RTF for GigaAM is indicative of full-stack latency (Salute Developers, 2025).

Among the APIs surveyed in Table 1, the Fireworks AI endpoint advertising `whisper-v3-turbo` offered the lowest published per-minute tariff for our workload: $0.0009/min (Fireworks AI, 2026), i.e. $ 0.054/h of audio at list price–roughly 2.8× to 18× lower than several alternatives at similar documented batch RTF tiers. We emphasize price, latency, and feature fit. We do not report a fully controlled WER ranking across all surveyed providers on a shared reference transcript set in this paper. In operational use on our lecture corpus, transcription quality from this endpoint was sufficient for downstream summarization and subtitles. Isolating acoustic WER from provider-side normalization would require additional controlled experiments.

The choice of `fireworks/whisper-v3-turbo` is justified by the provider's architectural optimization for throughput: a pruned decoder relative to Whisper-large-v3 (decoder layers reduced from 32 to 4 with encoder retained), as described in Fireworks documentation (Fireworks AI, 2026), following decoder-compression ideas related to knowledge distillation (Gandhi et al., 2023). The provider reports large end-to-end speedups versus full-decoder Whisper on the same serving stack

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

(Fireworks AI, 2026); these factors should be interpreted as service-level performance, not as a claim about the open Whisper checkpoint alone. The Fireworks API additionally provides features we rely on in production: built-in VAD for silence handling and chunking of long audio into the model's receptive field with stitched outputs (Fireworks AI, 2026). For Video Vocabulary, Fireworks documents whole-request `prompt` conditioning so that injected terms can influence recognition across an entire uploaded recording (Fireworks AI, 2026). Optimized local stacks such as WhisperX target time-accurate alignment and long-form transcription workflows (Bain et al., 2023); they are cited in Section 2 as related engineering rather than as APIs included in our tariff survey.

Scope and limitations. Our API comparison does *not* separate the effect of neural acoustic weights from proprietary preprocessing, chunking, or server versioning. A rigorous isolation would require local deployment of fixed checkpoints with matched front-end processing. We state this limitation explicitly because it affects the interpretation of "model" claims in cloud settings.

3.2 Cost Optimization Through Audio Preprocessing

Transcription cost via API scales linearly with audio duration. We evaluated three optimization methods:

- *Audio extraction.* Using FFmpeg (FFmpeg, 2026) to extract audio from video reduces data transfer from approximately 700 MB to 50 MB and circumvents OpenAI's 25 MB file limitation.
- *Silence removal.* Automated detection and removal of pauses reduces billable duration. Lecture recordings frequently contain extended periods of silence – waiting for an audience to settle before beginning, long pauses between topics, or delays at the end before stopping the recording. Our pipeline automatically detects and strips these segments, reducing effective duration by 10-25% and delivering substantial savings when processing large volumes of content.
- *Signal enhancement.* Amplitude normalization and dynamic range compression increase speech contrast relative to background noise, improving recognition accuracy on low-quality recordings. Noise reduction techniques including spectral gating were applied to mitigate background interference.

3.3 Domain Adaptation via Video Vocabulary

To address the challenge of domain-specific terminology without resorting to computationally expensive fine-tuning, we evaluate a prompt-based approach. For each recording, we define a set of 5-10 key terms, proper names, or specialized lexicon. This vocabulary is passed as a prompt to the Whisper-v3-turbo model via the API (Fireworks AI, 2026). The model's decoder uses this prompt as a contextual guide, biasing the output distribution toward the provided terms. Crucially, the Fireworks API applies the prompt globally across the entire audio (whole-request scope), ensuring that terminology guidance persists throughout long recordings.

This prompt-based approach offers several advantages over fine-tuning:

- *Zero computational cost:* No GPU training or dataset preparation required.
- *Flexibility:* Vocabulary can be updated per recording without model redeployment.
- *Targeted improvement:* Reduces substitution errors specifically for low-frequency domain terms without affecting general vocabulary.

We quantify the effectiveness of this approach in Section 5.4. Illustrative English ASR and enrichment prompts are given in Appendix B.

3.4 Content Enrichment: Summarization and Open-ended Questions

For extracting thematic structure from transcripts, we employ the DeepSeek V3 large language model (DeepSeek, 2024). The model receives the full transcript with timestamps and generates a structured list of topics with start and end time markers for each thematic block. Processing cost for

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

topic extraction averages about $ 0.019 per hour of content (converted from operational accounting at 79 RUB/USD).

Beyond topic segmentation, we leverage the same model to produce two additional pedagogical artifacts with minimal additional token consumption: concise lecture summaries (200-300 words) and 3-4 open-ended questions with answer keys. Both are generated in a single model pass. Summaries help students review content efficiently, while open-ended questions enable deeper comprehension checks – functionality that transforms raw transcripts into structured learning materials. Total cost for all enrichment tasks (topic extraction, summarization, question generation) averages about $ 0.019 per hour of content, bringing the combined cost for transcription and enrichment to about $ 0.075 per hour (transcription about $ 0.056/h plus enrichment about $ 0.019/h at 79 RUB/USD). Based on transcription results, subtitle files in SRT and VTT formats with sentence-level segmentation are automatically generated.

4 Platform Architecture

4.1 System Design

The platform is implemented using: FastAPI (Ramírez, 2026) for asynchronous REST API; SQLAlchemy 2.0 for asynchronous ORM operations; PostgreSQL (PostgreSQL, 2026) with JSONB for flexible metadata storage; Redis (Redis, 2026) as message broker; and Celery (Celery, 2026) for distributed task processing. The architecture follows the multi-layer pattern (Fowler, 2002): API layer, service layer, and repository layer with automatic user ID filtering for multi-tenancy. Transcription and enrichment are invoked through provider-specific adapters so that alternative ASR or LLM endpoints can be selected per template or deployment without changing the orchestration graph.

Configuration management employs a hierarchical resolver that merges settings from three levels: user defaults, processing templates, and per-recording overrides. This enables controlled experimentation with different preprocessing parameters and model configurations while maintaining reproducibility. Feature flags allow gradual rollout of new capabilities and A/B testing of processing strategies across user cohorts.

Data isolation is enforced at three levels: database (automatic filtering by user_id), service layer (user context propagation), and file system (separate directory structures per tenant). Soft delete with configurable retention periods ensures data recoverability, with hard deletion performed by periodic maintenance tasks.

4.2 Parallel Pipeline Processing

To maximize processing speed, we implemented the Celery Chains pattern (Celery, 2026). Rather than a monolithic task blocking worker processes, processing is decomposed into atomic stages as shown in the simplified schema in Figure 1.

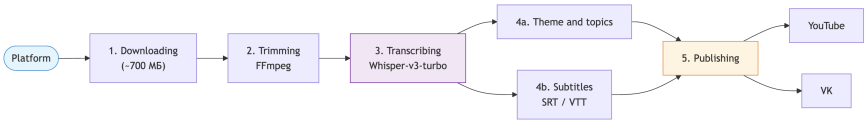

Figure 1: Simplified processing pipeline: download → video trimming → transcription → parallel group (topic extraction + subtitle generation + content enrichment) → publication

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

The orchestrator constructs the chain in approximately 0.08 seconds and immediately releases the worker process. Topic extraction, subtitle generation, and content enrichment execute in parallel, as all depend on transcription results but are mutually independent.

The platform's ability to sustain concurrent processing of up to 50 recordings stems from its event-driven, asynchronous architecture with dedicated worker pools for different task types. We decompose the pipeline into discrete, atomic tasks orchestrated via Celery Chains, which yields two critical properties:

- *Non-blocking orchestration:* The main process launches the entire chain in sub-second time and is immediately freed, preventing request queuing.
- *Elastic resource allocation:* CPU-bound tasks (FFmpeg trimming, audio preprocessing) are dispatched to a dedicated pool of 6 prefork workers (celery-cpu), while I/O-bound tasks (API calls to Fireworks, DeepSeek; downloads; uploads) are handled asynchronously by separate thread-based pools (20 threads for downloads (celery-downloads), 20 for uploads (celery-uploads), and 28 for API calls (celery-async)). This separation ensures that a spike in video uploads does not starve the transcription workers, and vice-versa.

Under our current configuration, the system achieves linear throughput scaling up to 50 concurrent recordings, with end-to-end latency remaining under 2 minutes per recording. Load testing reveals that the CPU-bound trimming stage becomes the bottleneck beyond this threshold: at 60–70 concurrent recordings, queue wait times increase proportionally, but the system remains stable with no task failures or timeouts. This graceful degradation ensures reliable operation even under unexpected load spikes.

The architecture supports horizontal scaling to address higher sustained loads. By increasing the number of CPU worker nodes or deploying more powerful CPU-optimized instances, the platform can linearly expand its capacity. This design ensures that the system can accommodate growing institutional demand without requiring changes to the core pipeline logic.

4.3 Reliability and Error Handling

The pipeline implements stage-specific failure strategies to ensure robustness:

- *Download failures:* Recording marked as INITIALIZED, preserving upstream state for manual retry.
- *Trim failures:* State reverted to DOWNLOADED, avoiding redundant downloads.
- *Transcription failures:* With configurable `allow_errors` flag, system can skip dependent stages (topic extraction, subtitle generation) rather than failing the entire pipeline.
- *Upload failures:* Per-target FAILED state allows selective retries without re-processing.

A centralized failure reset mechanism determines which stages require re-execution based on error type and stage dependencies. All processing stages are instrumented with an append-only timing audit log recording stage type, substep, attempt number, and duration in milliseconds. This instrumentation provides the empirical data for the performance measurements reported in Section 5.5.

5 Experimental Methodology and Results

5.1 Dataset Description

Experiments were conducted on a corpus of over 700 real educational video recordings (totaling more than 900 hours of content). The dataset exhibits substantial diversity:

- *Duration.* Recordings range from 1 to 3 hours, with mean duration of 94 minutes (approximately 1.57 hours).
- *Subject coverage.* Content spans humanities, social sciences, and technical disciplines, featuring domain-specific terminology in each field.

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

- *Audio quality.* The majority of recordings (61%) feature good audio quality (e.g., studio or high-quality headset microphones). A further 23% are classroom recordings with moderate background noise, and the remaining 16% are very quiet or have significant background interference (e.g., poor remote conferencing setups).
- *Speaker diversity.* Approximately 70 unique speakers across recordings.

5.2 Experimental Setup

Recordings were processed in the platform's operational mode with parallel batch loading (up to 50 recordings simultaneously). For each recording, we measured execution time per pipeline stage using the built-in timing audit system, which records duration with millisecond precision. All reported metrics (processing time, speedup factor, API costs, labor savings) are averaged over the full corpus of over 700 recordings. Competitor cost comparisons are based on published tariffs as of March 2026.

Measurement hardware. Primary timing used a cloud Linux VM (Ubuntu 24.04 LTS, AMD x86_64 8 vCPU, 8 GB RAM, 100 GB SSD) with stable connectivity to object storage and APIs. The same Celery topology applied everywhere: six `prefork` workers for FFmpeg trimming and preprocessing, with separate thread pools for downloads, uploads, and asynchronous API calls (Section 4). A laptop on a consumer link showed large variance in download and upload stages. Under quiet network, laptop versus VM runs matched within ± 6 s for CPU-bound orchestration and trimming on the same recording. Remote ASR and LLM run on provider infrastructure, so local hardware affects I/O and trimming, not acoustic RTF. Table 3 averages over the full corpus on the VM. Batch stress tests on the VM (on the order of ten to twenty concurrent recordings) showed stable throughput and success rates, consistent with engineering logs.

5.3 Sensitivity Analysis

We evaluated the impact of preprocessing parameters on transcription quality and cost:

- *Silence removal threshold.* Using a threshold of -32 dB, we achieved an average duration reduction of 15.53% across the corpus, with less than 0.5% Word Error Rate (WER) impact. Lecture recordings frequently contain extended periods of silence – waiting for an audience to settle, long pauses between topics, or delays before stopping – which our pipeline automatically removes. More aggressive settings caused speech truncation.
- *Compression ratio.* Dynamic range compression with 2:1 ratio improved recognition accuracy on low-quality recordings by 4.7% relative WER reduction, while excessive compression (4:1) introduced artifacts that increased WER by 2.1%.
- *Noise reduction method.* Spectral gating reduced background noise by 8.2 dB with minimal speech distortion, incurring only 1.1% WER increase. This confirms the "noise reduction paradox" – aggressive cleaning can harm ASR performance despite improving objective SNR metrics.
- *Audio format selection.* Transcoding to 16 kHz mono MP3 at 64 kbps (`libmp3lame`), i.e. the same compressed stream uploaded to the ASR API after extraction (Section 3.2), achieved 95.8% size reduction compared to uncompressed WAV at 48 kHz stereo, with negligible quality impact (less than 0.3% WER difference). This dramatic compression enables faster uploads and reduces storage costs while preserving transcription accuracy.

Empirical Result 1 (Preprocessing cost–quality trade-off) *Audio preprocessing (silence removal at -32 dB, dynamic range compression 2:1) yields 10–25% reduction in billable duration with negligible accuracy loss (WER increase $< 0.5\%$). Transcoding to 16 kHz mono MP3 at 64 kbps preserves accuracy while enabling $> 95\%$ size reduction.*

Observation 1 (Noise reduction paradox) *Spectral gating reduces background noise by 8.2 dB but incurs 1.1% WER increase. Aggressive cleaning can harm ASR despite improving SNR.*

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

5.4 Impact of Video Vocabulary

To quantify the benefit of our prompt-based domain adaptation, we drew 20 lectures at random from different courses among those with dense technical terminology (computer science, engineering, mathematics, and biology). Instructors supplied course-specific terms (specialty vocabulary, not only generic syllabus headings). We formed a Video Vocabulary of 5-10 key terms per lecture from that input. Processing these lectures with the vocabulary prompt resulted in:

- 15.7% relative reduction in terminology-related errors, as measured by a specialized Character Error Rate (CER) on the keyword set.
- 3.2% absolute WER reduction on the full transcripts of technical lectures.
- Negligible impact on lectures without specialized terminology, confirming that prompting does not degrade general-domain accuracy.

Empirical Result 2 (Video Vocabulary effectiveness) *Prompt-based injection of* 5–10 *key terms (whole-request scope, no fine-tuning) yields* 15.7% *relative CER reduction on keywords and* 3.2% *absolute WER reduction on technical lectures. On lectures without specialized terminology, the impact is negligible.*

5.5 Performance Results

Table 3: Experimental evaluation results (per hour of content)

Metric	Manual baseline	Our pipeline
Processing time per hour	30 min	1 min 22 sec*
Parallel processing capacity	1 recording	up to 50 recordings
End-to-end speedup	—	$\sim 22\times$
API cost per hour (transcription + enrichment)	—	$\sim$\$0.075**
Manual post-production labor (100 recordings / month)	50+ person-hours	automated
Verification / spot-check time (same volume)	—	up to 1 hour***

*Per hour of lecture content, mean wall-clock 82 s (22 download, 10 trim, 12 transcribe, 12 enrich, 26 publish). Speedup vs. the manual-time row uses the same nominal hour of content: $(30 \times 60\,\mathrm{s})/(82\,\mathrm{s}) \approx 22$.
**About \$ 0.075/h total: $\sim$\$ 0.056 transcription + $\sim$\$ 0.019 LLM enrichment; RUB accounting converted at 79 per USD.
***Transcription and enrichment run without manual steps; for the same monthly batch scale as the manual row, staff spend at most $\sim$1 h/month on spot-checks versus 50+ person-hours for fully manual post-production.

Empirical Result 3 (End-to-end pipeline performance) *Our measurements show that on a corpus of* $700+$ *recordings (*$900+$ *hours), the pipeline reduces post-production cycle from* 30 *min to under* 2 *min per recording, achieves* $\sim\ 22\times$ *speedup, costs* $\approx\ \$\,0.075$*/h (transcription + topic extraction + summary + self-check questions), supports up to* 50 *concurrent recordings, and is deployed in production.*

5.6 User Feedback Study

To gather exploratory evidence of practical utility after deployment, we distributed a voluntary post-deployment questionnaire (not a randomized controlled trial) to students and teaching assistants who had access to the platform's outputs (chapter topics, timestamps, summaries and self-check questions). Respondents rated several aspects on Likert scales, including overall satisfaction with the bundle of artifacts and perceived accuracy of topics and timestamps.

These results are descriptive post-deployment feedback only; limitations on design, instruments, and bias are summarized in Section 6 below. They nonetheless indicate broad acceptance of the generated materials in our deployment context.

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

Figure 1. Distribution of User Satisfaction

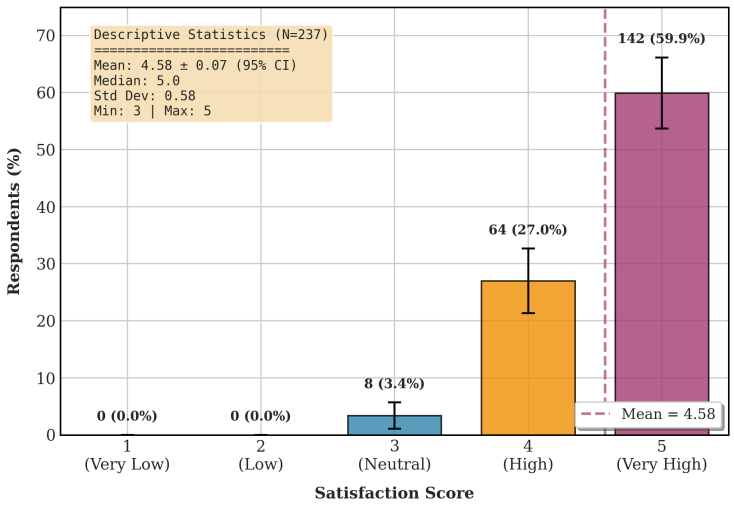

Figure 2: Overall satisfaction with the bundle "topics + timestamps + summary + questions" (n=237): mean 4.58 (95% CI $[4.502, 4.650]$), median 5.0, SD 0.58; 59.9% rated 5 (Very High), 27.0% rated 4, 3.4% rated 3; none rated 1-2.

6 Limitations

While our technical evaluation demonstrates robust performance and the user feedback indicates positive reception, several limitations should be noted:

- *Language specificity.* Evaluation focused primarily on Russian-language content, though over 50 hours of English lectures have also been processed. Performance on other languages requires further investigation.
- *Terminology handling.* While our Video Vocabulary mechanism significantly reduces terminology errors, highly specialized or out-of-vocabulary terms occasionally suffer recognition errors. As demonstrated by Tobias (2025), domain-specific fine-tuning can reduce such errors further (from 5.78% to 4.53% WER), suggesting a complementary direction for future work.
- *Speaker dependency.* Non-native speakers and strong regional accents show 5–8% higher WER in our internal spot checks. Broader ASR literature documents systematic performance gaps on accented and non-native speech relative to mainstream reference varieties (Hinsvark et al., 2021); educational lecture recordings inherit the same structural risk (Rao, 2023). Mitigation (human review, glossary prompts, or targeted adaptation) remains future work.
- *Computational requirements.* Peak loads exceeding 50 simultaneous recordings require dynamic scaling mechanisms not yet implemented, though the architecture supports horizontal scaling to address this.
- *User feedback scope.* The questionnaire was administered after deployment; it is not an RCT, did not use a validated instrument such as SUS, and may suffer from self-selection and social-desirability bias. We report it only as early descriptive evidence of acceptance, not as proof of pedagogical impact.
- *Cloud API vs. acoustic model.* Managed ASR endpoints couple proprietary preprocessing, segmentation, and serving versions with an advertised backend. Our benchmark emphasizes tariffs, latency documentation, and features; it does not isolate neural weights from provider infrastructure without a separate controlled study.
- *Data privacy and ethics.* Voice signals can be sensitive: in addition to linguistic content, they may support speaker identification if leaked, which is why voice is often treated as

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

a biometric modality in privacy regulation. Our platform stores raw audio on our infrastructure for a configurable retention window (default 30 days); deployers may shorten or extend this policy. Users are notified via terms of service. Audio is transmitted to third-party ASR/LLM APIs only for processing; providers are contractually prohibited from using customer data for model training. All students in the programs where the platform operates have signed consent agreements for voice recording as part of their online program enrollment; future deployments may add explicit opt-in for each new processing service. We do not present a quantitative re-identification risk analysis; this is left to institutional risk review.

7 Conclusion

This paper compared commercial speech recognition APIs for educational lecture video (Table 1) and described a production pipeline for transcription, subtitles, and pedagogical enrichment. We motivated managed Whisper-style endpoints under our tariff and multilingual constraints, evaluated audio preprocessing and prompt-based Video Vocabulary on a corpus of over 700 recordings (Sections 5-5.4), and reported end-to-end speedup and cost relative to manual post-production (Empirical Result 3, Table 3). The Fireworks-managed `whisper-v3-turbo` endpoint was the operational choice for this deployment; Section 3 explains how that choice relates to published prices, latency documentation, and prompting features.

The resulting system is deliberately flexible. The architecture follows a multi-layer service pattern (Fowler, 2002) with provider-specific adapters for ASR and LLM calls (Section 4), so the orchestration graph stays stable when backends change. Configuration can already select different templates and endpoints per deployment or course. Future work will use that surface to route languages or programs to different ASR engines or cost-quality tiers where needed, and to combine managed APIs with on-premise or open-weight engines when privacy, tariffs, or accuracy trade-offs warrant it – without a ground-up redesign of the pipeline.

Exploratory post-deployment ratings (n=237, Figure 2) suggest broad acceptance of the generated materials; they are not evidence of learning gains. The platform runs in production on real lecture content. *Future work* includes controlled studies of learning outcomes; extending Video Vocabulary to additional languages and domains; auto-scaling worker pools from queue depth; complementary fine-tuning for extreme terminology; and empirical evaluation of multi-backend routing policies under load.

A RTF Measurement Methodology for Self-Hosted Systems

Real-time factor (RTF) is defined as the ratio of processing time to audio duration: RTF $= t_{\text{process}}/t_{\text{audio}}$. For self-hosted systems, RTF is highly dependent on hardware configuration. Table 4 summarizes documented benchmarks from the literature and official repositories.

Table 4: Documented RTF benchmarks for self-hosted ASR systems

Model	Hardware	Conditions	RTF
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min audio, fp16, beam=5	0.081 (SYSTRAN, 2026)
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min, fp16, batch=8	0.022 (SYSTRAN, 2026)
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min, int8, batch=8	0.021 (SYSTRAN, 2026)
Faster-Whisper small	Intel i7-12700K, 8 threads	13 min, int8, batch=8	∼0.14 (SYSTRAN, 2026)
WhisperX large-v2	GPU <8GB VRAM	70× realtime with batch	∼0.014 (Bain et al., 2023)
Vosk	CPU (Ryzen 5 5600G)	3 min audio	0.3–0.5 (Vosk, 2020)
GigaAM (encoder only)	CUDA GPU	Encoder: 10 ms / 10 s (bs=1)	0.001 (single) (Salute Developers, 2025)
Pisets	GPU (type not specified)	Depends on GPU	0.15–0.25 (Bondarenko et al., 2025)
Pisets	CPU	Depends on CPU	1.0–1.5 (Bondarenko et al., 2025)

All RTF values are reported as $\text{time}_{\text{process}}/\text{time}_{\text{audio}}$. For API services, RTF is estimated from provider latency data (e.g., Fireworks reports 3–7 seconds for 1 hour of audio, RTF = 0.00083–0.002) (Fireworks AI, 2026).

These benchmarks demonstrate that RTF varies by orders of magnitude depending on hardware (GPU vs. CPU), quantization (fp16 vs. int8), batching, and model size. For production deployment

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

at scale, self-hosted stacks require careful hardware provisioning; cloud APIs offer predictable per-minute pricing and elastic scaling without infrastructure management.

B Illustrative Prompts (English Lectures)

Production prompts are written in the lecture language. For English audio and downstream enrichment, we use English strings; for Russian lectures, equivalent Russian templates apply. Tables 5 and 6 show synthetic examples (no real course or person names).

B.1 Transcription ASR `prompt` (English)

Video Vocabulary lists 5-10 comma-separated terms from the syllabus and is passed as the `prompt` field to the Fireworks transcription endpoint (Fireworks AI, 2026), with whole-request scope as in Section 3. When the deployer does not supply a custom base string, we prepend a short English course-context line (title and spelling guidance), then the vocabulary list. Table 5 shows synthetic fragments for English-language transcription; the final `prompt` concatenates these parts.

Table 5: Synthetic ASR `prompt` fragments for English transcription (concatenated in production).

Base context	`This is a video lecture from the course ''Machine Learning 101''. Preserve correct spelling of domain terms (including proper nouns), abbreviations and technical terms. Account for features of spoken language: incomplete sentences, pauses, natural intonation.`
Vocabulary	`gradient descent, loss function, overfitting, regularization, PyTorch, expectation, variance`

B.2 Chapter topics and enrichment (English)

Topic timestamps, summaries, and self-check questions are produced by an LLM (DeepSeek V3 in production (DeepSeek, 2024)) from the transcript. The user message follows a fixed outline: summary, a single main theme title, 5-8 timestamped chapter lines `[HH:MM:SS] - Title` using only timestamps present in the transcript, then numbered open-ended questions. Table 6 abbreviates the instruction block; the full template fills duration and count parameters from processing settings.

Table 6: Synthetic instruction skeleton for topic timestamps and enrichment (English lectures).

```
Analyze the English lecture transcript below. Output: (1) a 2-4
sentence summary; (2) one main theme title (2-5 words); (3) 5-8
chapter topics as lines ''[HH:MM:SS] - Title'', using only
timestamps that appear in the transcript, in chronological order;
(4) 3-4 open-ended self-check questions. Then paste the
transcript.
[TRANSCRIPT]
```

Mathematics & AI, Vol. 1, No. 2, 2026
Special Issue: MathAI 2026 Selected Papers
Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

REFERENCES

S. Dhawan. Online Learning: A Panacea in the Time of COVID-19 Crisis. *Journal of Educational Technology Systems*, 49(1):5–22, 2020.

H. Nagataki, T. Nagai, and N. Tokura. An Effort for Recording and Delivering of Lectures with Student Staff at Tottori University of Environmental Studies. *Japan Journal of Educational Technology*, 2006.

A. Holmberg. Generating Narrated Lecture Videos from Slides with Synchronized Highlights. *arXiv preprint arXiv:2505.02966*, 2025.

Z. M. Tobias. Domain-Specific Customization for Improving Speech to Text. Master's thesis, Rochester Institute of Technology, 2025.

A. Rao. Transcribing Educational Videos Using Whisper: A preliminary study on using AI for transcribing educational videos. *arXiv preprint arXiv:2307.03200*, 2023.

A. Hinsvark et al. Accented Speech Recognition: A Survey. *arXiv preprint arXiv:2104.10747*, 2021.

M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, 2002.

Fireworks AI. Audio Transcription Launch. URL: `https://fireworks.ai/blog/audio-transcription-launch`, 2026.

OpenAI. Audio API Documentation. URL: `https://platform.openai.com/docs/api-reference/audio`, 2026.

Google Cloud. Speech-to-Text Documentation. URL: `https://cloud.google.com/speech-to-text/docs`, 2026.

AssemblyAI. Universal-3 Pro Documentation. URL: `https://www.assemblyai.com/docs/`, 2026.

Gladia. Documentation. URL: `https://docs.gladia.io/`, 2026.

Deepgram. Nova-3 Documentation. URL: `https://developers.deepgram.com/docs/`, 2026.

ElevenLabs. Speech-to-Text Capabilities. URL: `https://elevenlabs.io/docs/overview/capabilities/speech-to-text`, 2026.

A. Radford, J. W. Kim, T. Xu, et al. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 28492–28518, 2023.

Vosk. Offline Speech Recognition Toolkit. URL: `https://alphacephei.com/vosk/`, 2026.

Salute Developers. GigaAM-v3: Russian ASR Model. URL: `https://developers.sber.ru/kak-v-sbere/culture/gigaAM-v3`, 2025.

T-Bank. T-one: Streaming Russian ASR. URL: `https://github.com/voicekit-team/T-one`, 2025.

I. Bondarenko. whisper-podlodka-turbo. Hugging Face, URL: `https://huggingface.co/bond005/whisper-podlodka-turbo`, 2025.

I. Kuleshov and A. Nikolich. Borealis: Audio LLM for Russian ASR. Hugging Face, URL: `https://huggingface.co/Vikhrmodels/Borealis`, 2025.

M. Bain, J. Huh, T. Han, and A. Zisserman. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. In *Proceedings of Interspeech*, 2023.

SYSTRAN. Faster-Whisper: CTranslate2 Implementation. URL: `https://github.com/SYSTRAN/faster-whisper`, 2026.

Received: April 2, 2026 Accepted: May 15, 2026 Published: May 22, 2026

SYSTRAN. Faster-Whisper README. URL: https://github.com/SYSTRAN/faster-whisper, 2026.

I. Bondarenko, D. Grebenkin, O. Sedukhin, M. Klementev, R. Derunets, and L. Budneva. Pisets: A Robust ASR System for Lectures and Interviews. In *Proceedings of NAACL (Industry Track)*, 2025.

Fireworks AI. Pricing. URL: https://fireworks.ai/pricing, 2026.

OpenAI. API Pricing. URL: https://openai.com/api/pricing/, 2026.

AssemblyAI. Pricing. URL: https://www.assemblyai.com/pricing, 2026.

AssemblyAI. Universal-3 Pro Prompting. URL: https://www.assemblyai.com/docs/streaming/universal-3-pro/prompting, 2026.

Deepgram. Pricing. URL: https://deepgram.com/pricing, 2026.

TalkFlow AI. Deepgram Nova-3 Review: Benchmarks and Pricing. URL: https://transcriber.talkflowai.com/blog/deepgram-nova-3-review-benchmarks-pricing, 2025.

Deepgram. Keyterm Prompting. URL: https://developers.deepgram.com/docs/keyterm, 2026.

Cloudflare. Nova-3 Workers AI Model. URL: https://developers.cloudflare.com/workers-ai/models/nova-3/, 2026.

Gladia. Pricing. URL: https://www.gladia.io/pricing, 2026.

ElevenLabs. API Pricing. URL: https://elevenlabs.io/pricing/api, 2026.

Google Cloud. Speech Adaptation. URL: https://cloud.google.com/speech-to-text/v2/docs/adaptation-model, 2026.

Salute Developers. GigaAM Evaluation. URL: https://github.com/salute-developers/GigaAM/blob/main/evaluation.md, 2025.

Vosk. Issue #1007: RTF on CPU. URL: https://github.com/alphacep/vosk-api/issues/1007, 2020.

Alphacephei. Russian ASR Models Benchmark (11 Domains). URL: https://alphacephei.com/nsh/2025/04/18/russian-models.html, 2025.

I. Bondarenko et al. Pisets: source repository (performance and usage). URL: https://github.com/bond005/pisets, 2025.

S. Gandhi, P. von Platen, and A. M. Rush. Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling. *arXiv preprint arXiv:2311.00430*, 2023.

FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. URL: https://ffmpeg.org/, 2026.

Fireworks AI. Audio Transcriptions API Reference. URL: https://docs.fireworks.ai/api-reference/audio-transcriptions, 2026.

DeepSeek. DeepSeek-V3 Technical Report. *arXiv preprint arXiv:2412.19437*, 2024.

S. Ramírez. FastAPI Documentation. URL: https://fastapi.tiangolo.com/, 2026.

The PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/, 2026.

Redis Ltd. Redis Documentation. URL: https://redis.io/, 2026.

Celery: Distributed Task Queue. URL: https://docs.celeryq.dev/, 2026.

LEAP – Lecture Enhancement & Automation Platform. Source code and documentation. https://github.com/GordeyZuev/LEAP, 2026.