International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

CogniLearn: An Adaptive AI Learning Assistant with Real-time Multi-Signal Confusion Detection

Mr. Sandeep B, C V Vaishnav, Divya S Dalawai, G R Shalini, Karthik G K
Department of Computer Science and Engineering JNN College of Engineering, Shivamogga, Karnataka, India
sandeepb@jnnce.ac.in

***Abstract*—Learners who watch long educational videos on platforms such as YouTube often hit a concept they do not understand but keep playing the video, because stopping to look for help breaks their flow. This paper describes CogniLearn, a Chrome extension with a local backend that notices likely cognitive overload while a video is playing and explains the concept being taught at that moment. A webcam frame is sampled at 4 fps and processed entirely on the device by MediaPipe Face Landmarker, which yields facial blendshape scores and head pose. A per-user detector calibrates a personal baseline in roughly 30 seconds using an online mean and variance estimate, converts each frame into a weighted z-score of effort-related facial actions (brow lowering, eye squint, inner-brow raise, mouth press), and aggregates frames over a 10-second sliding window. Viewing behaviour (rewinds and pauses) can amplify the facial evidence but can never trigger an intervention alone. A trigger fires only when the score stays above a personal threshold for 3 seconds and a cooldown has expired. The threshold is then adapted from real user labels ("Helpful" and "I wasn't confused") and from manual help requests. On a trigger, the video is paused and a FastAPI server fetches the transcript around the current timestamp and asks Gemini 2.5 Flash for a short explanation in the speaker's language, which is also read aloud with browser speech synthesis. The same transcript and a memory of earlier explanations are used to build a three-question multiple-choice quiz and exportable Markdown study notes. Only numeric features and the video ID leave the browser; camera frames never do. The paper describes the design, the algorithms and the parameters in detail, states the limitations of the current prototype, and proposes an evaluation protocol.**

Keywords*—Cognitive Overload Detection; Adaptive Learning; Facial Blendshapes; MediaPipe; Personalised Thresholding; Large Language Models; Gemini; Chrome Extension; FastAPI; Educational Video

I. INTRODUCTION

Video is now a main way of learning programming, mathematics and other technical subjects. Unlike a classroom, a video cannot see the learner. A confused student has three poor options: rewind and hope, pause and search elsewhere, or carry on and fall behind. Cognitive load theory explains why this matters: when the demands of a task exceed working-memory capacity, learning suffers [1]. Studies of affect during complex learning also report that confusion is common and that how long it persists decides whether it helps or hurts learning [3].

Facial expression is one cheap, non-intrusive signal of such states. Action units linked to effort and concentration, such as brow lowering and eyelid tightening, are described in the Facial Action Coding System [2], and automatic recognition of engagement from faces has been studied for some time [4]. Two difficulties keep this signal from being used directly in a product. First, people differ a great deal in resting expression, camera angle and lighting, so a single global threshold gives many false alarms for some users and misses for others. Second, a webcam stream is sensitive data, so a design that uploads video is hard to justify to users.

A third difficulty is what to do once confusion is suspected. A generic tip is of little use; the help has to be about the exact concept the speaker is explaining, in the language the speaker is using.

CogniLearn addresses these three problems together. Its contributions are:

1. An **on-device, per-user detector** that calibrates a personal facial baseline and scores effort as a z-score against it, with explicit handling of blinks, looking away and eye closure.
2. An **evidence-gated trigger**: a sliding-window score with a hold time and a cooldown, in which behavioural signals only amplify facial evidence.
3. **Label-driven threshold adaptation** that learns only from real user feedback, never from the system's own decisions.
4. A **transcript-grounded LLM tutor** that returns an explanation in the transcript's language together with its BCP-47 tag so it can be spoken aloud, plus a memory-based quiz and study-note export.
5. A **privacy-preserving architecture** in which frames are analysed locally and only numeric scores, thresholds and the video ID are sent to a local server.

Section II reviews related work. Section III gives the architecture and methods. Section IV describes the implementation. Section V gives the evaluation protocol, Section VI lists limitations, and Section VII concludes.

II. LITERATURE SURVEY

Cognitive load and affect in learning. Sweller's cognitive load theory [1] holds that learning is hindered when intrinsic and extraneous load exceed working memory. D'Mello and Graesser [3] describe a sequence of affective states in complex learning, including engagement, confusion, frustration and boredom, and show that confusion can be productive if it is resolved. This motivates intervening when

confusion persists rather than at its first sign, which is the role of the hold time in our detector.

Facial analysis for engagement. The Facial Action Coding System [2] defines the action units from which expression research is built. Whitehill et al. [4] showed that engagement can be estimated from facial expression video, and that such estimates are noisy for individuals. Per-person normalisation is therefore a central design choice here, in line with that finding.

Real-time face tracking. MediaPipe [5] provides cross-platform perception pipelines, and its Face Landmarker builds on real-time facial surface geometry estimation from monocular video [6]. It outputs landmarks, 52 blendshape coefficients that resemble action units, and a facial transformation matrix from which head pose can be derived. It also runs as WebAssembly in the browser, which makes on-device inference practical.

LLMs for tutoring. Large multimodal language models such as Gemini [7] can explain a short passage of text clearly and in many languages. Their weakness is grounding: without context they produce generic advice. Supplying the speaker's own transcript around the current timestamp reduces this problem.

Retrieval practice. Roediger and Karpicke [9] showed that testing improves later retention more than re-reading. This supports the memory quiz that CogniLearn builds from the watched transcript.

Gap. Existing engagement-detection work is mostly offline or classroom-based, and existing video tutors usually wait for the learner to ask. We found few openly documented systems that combine a locally calibrated facial detector, a conservative trigger, adaptive thresholding from real feedback and transcript-grounded multilingual explanation inside the video player itself. That is the gap this work addresses.

III. PROPOSED METHODOLOGY

The system is a pipeline: **capture, feature extraction, per-user detection, trigger, transcript retrieval, LLM explanation, and speech, feedback and memory**. Each stage is described below.

A. System Architecture

CogniLearn has four runtime components.

- **Content script (React).** Injected into YouTube pages. It draws the floating widget and the draggable camera preview, owns the webcam stream, runs the sampling loop, hosts the detector and controls the video element.
- **Offscreen document.** A Manifest V3 extension page that runs MediaPipe Face Landmarker (CPU delegate, WebAssembly) in IMAGE mode. A service worker cannot hold a long-lived WASM model, so an offscreen document is used.
- **Background service worker.** Creates the offscreen document, forwards frames to it, and relays backend calls so that the content script does not make cross-origin requests itself.
- **Local backend (FastAPI).** Exposes `/analyze`, `/feedback`, `/generate_quiz`, `/stats` and `/export` on `127.0.0.1:8000`, fetches transcripts, calls Gemini and persists memory and an event log.

The flow for one sample is: the content script draws the webcam frame onto a 320x240 canvas and encodes it as JPEG (quality 0.7), the service worker passes it to the offscreen document, MediaPipe returns blendshapes and a transformation matrix, and a compact feature vector comes back to the content script. The detector consumes the vector. Only if it triggers does the content script send a numeric payload to the backend.

B. On-Device Feature Extraction

From the Face Landmarker output the following scalar features are computed per frame:

TABLE I
FACIAL FEATURES EXTRACTED PER FRAME

Feature	Definition
brow	mean of browDownLeft and browDownRight
squint	mean of eyeSquintLeft and eyeSquintRight
innerUp	browInnerUp
mouthPress	mean of mouthPressLeft and mouthPressRight
blink	mean of eyeBlinkLeft and eyeBlinkRight
gazeOff	min(1, sqrt(h^2 + v^2)); h, v from the eyeLook* blendshapes
yaw, pitch	head angles (deg) from the 4x4 facial transformation matrix
present	whether a face was found

Because the extractor returns only these numbers, the raw blendshape vector and the image never leave the offscreen document.

C. Per-User Calibration

The detector keeps a running mean and variance for seven features (brow, squint, innerUp, mouthPress, yaw, pitch, gazeOff) using Welford's online algorithm [10], so no samples are stored. Calibration collects 120 valid frames, about 30 seconds at 4 fps. A frame is valid only if a face is present, the eyes are open (blink < 0.5), gaze offset is below 0.6 and |yaw| is below 30 degrees. Calibration is saved per user in `chrome.storage.local` and is discarded after 12 hours, because lighting and seating usually change by then. The learner can also force recalibration from the widget. A floor of 0.03 on each standard deviation prevents division by a near-zero spread for people with very still faces.

D. Frame Scoring

After calibration, each frame is classified as on-screen if gaze offset is no more than the personal mean plus a margin of 0.25, and yaw and pitch are within 20 and 18 degrees of the personal means. Frames with no face or off-screen gaze count as *off*. Frames with blink >= 0.5 count as *closed* and are excluded from effort scoring, because a blink distorts brow and squint readings.

For every remaining frame the z-score of each feature against the baseline is clipped at zero from below, and the effort z-score is:

$$z_effort = 0.40\ z_brow + 0.25\ z_squint + 0.20\ z_innerUp + 0.15\ z_mouthPress$$

A frame is *effortful* if z_effort >= 1.0.

E. Window Aggregation and States

Frames from the last 10 seconds are kept, and no judgement is made on less than 5 seconds of data. Over the window let *off*, *closed* and *valid* be the fractions of each kind. The state is then decided in this priority order:

1. **distracted** if the off fraction is at least 0.5;
2. **fatigued** if the closed fraction is at least 0.3;
3. otherwise, if at least half the frames are valid, the facial score is computed and the state is **overload** or **focused**.

The facial score and final score are:

$$facial = 0.5\ clamp(mean\ z_effort / 2.0) + 0.5\ (fraction\ of\ effortful\ frames)$$

$$score = clamp(facial\ (1 + 0.5\ behaviour))$$

The behaviour term is the sum over the last 30 seconds of 0.5 for each rewind (a backward seek of more than 2 seconds) and 0.25 for each pause, clamped to [0, 1]. Pauses caused by the system itself are ignored for 1.5 seconds. Because behaviour only multiplies the facial score, a score of zero stays zero, so rewinding or pausing alone can never trigger an intervention.

While the score is below 70% of the threshold, the effort baselines drift slowly towards the current values (rate 0.002 per sample), which handles gradual changes in posture and lighting without recalibration.

F. Trigger Logic

A trigger is raised only if the state has been *overload* continuously for at least 3 seconds and at least 60 seconds have passed since the last trigger. Processing pauses and the window is reset whenever the video is paused, ended, in an advertisement, or the tab is hidden, so stale data never crosses these boundaries. If the learner has been distracted for more than 20 seconds, the video is paused with a "you looked away" message.

G. Adaptive Threshold from Real Labels

The personal threshold starts at 0.55 and is bounded to [0.30, 0.90] in steps of 0.05. It changes only on explicit evidence:

TABLE II
THRESHOLD UPDATE RULES FROM USER FEEDBACK

Event	Update
"I wasn't confused" after an automatic trigger (false positive)	+0.05
"Helpful" after an automatic trigger (true positive)	-0.025
Manual "Explain" while score < threshold (missed detection)	-0.025
Feedback after a manual request (not evidence about auto-trigger accuracy)	none

This is a simple online, asymmetric update: false alarms are penalised twice as strongly as confirmations are rewarded, which favours fewer interruptions. It is related in spirit to incremental value updates in reinforcement learning [8], but it has no state transitions or discounting and should be seen as a feedback-driven controller. The detector persists the threshold locally. The backend also stores a copy and logs every feedback event with its score, threshold and feature summary. These labelled records can later train a supervised model.

H. Transcript-Grounded Explanation

On an allowed request, the backend fetches the transcript with `youtube-transcript-api`, requesting a list of 30 language codes (English variants and major Indian and world languages). It keeps the snippets that overlap the current time or start within +/-10 seconds of it. This text is sent to Gemini 2.5 Flash with a prompt that asks the model to identify the subject, explain the core idea in two to four short sentences, answer in the transcript's language, and begin with a BCP-47 language code followed by a pipe character (for example `hi-IN|...`). The content script splits this prefix from the text, shows the text and passes the language tag to speech synthesis.

If no transcript is available, a generic study tip is generated instead. If the API fails, a fixed fallback message with the pause time is returned. Only explanations built from a real transcript are saved to memory, so generic tips and error messages never pollute it.

The backend enforces a 45-second per-user cooldown on automatic requests as a second guard; manual requests are always allowed.

I. Speech Output

Browser speech synthesis is used with a language-matched voice. A local voice is preferred as fallback. Text is split into sentence-sized chunks, which avoids the cut-off that network voices show on long utterances, and speech restarts with a local voice if the first voice fails. Sentence splitting also recognises the Devanagari danda, so Hindi text is chunked correctly.

J. Memory, Quiz and Notes

The backend keeps a persistent list of explanations that were genuinely generated from transcripts (the "concepts the learner needed help with"). For a quiz it takes the transcript from the start of the video up to the current time (the most recent 12,000 characters, cached per video) and the last five memory items. It asks Gemini for a three-question multiple-choice quiz as a JSON array in the transcript's language, using JSON response mode, and parses it. The learner sees the questions one at a time with a final score. A separate endpoint returns all remembered concepts as Markdown for download. A statistics endpoint reports the current sensitivity and the number of concepts.

K. Privacy and Data Flow

No image, video or raw blendshape vector leaves the browser. The backend receives: a random per-install user ID,

the numeric score, threshold and state, a small feature summary (mean z, effortful fraction, off fraction, closed fraction, behaviour), the video ID and the current timestamp. The backend runs on the loopback interface, and the extension's host permissions are limited to YouTube and that local address. The widget states this to the user, and the camera preview can be hidden without stopping tracking.

IV. IMPLEMENTATION

TABLE III
IMPLEMENTATION TECHNOLOGY STACK

Layer	Technology
Extension	Chrome Manifest V3, React, Vite; content script, offscreen document, service worker
Vision	MediaPipe Tasks Vision (Face Landmarker, WASM, CPU)
Detector	JavaScript class with persistence in chrome.storage.local
Backend	Python, FastAPI, Pydantic, requests, youtube-transcript-api
LLM	Gemini 2.5 Flash (REST API)
Persistence	JSON memory file; JSON-lines event log
Speech	Web Speech API (SpeechSynthesis)

The interface is a draggable floating widget that collapses to a brain icon while tracking, with panels for tracking status, intervention, away, quiz, results and analytics. A second draggable window shows the mirrored camera feed. The popup page only shows system status. The service worker also guards against duplicate offscreen documents and returns a not-ready response if the model has not loaded, in which case the content script skips the sample instead of treating it as an absent face.

The key parameters are summarised below.

TABLE IV
KEY SYSTEM PARAMETERS

Parameter	Value
Sampling rate	4 fps (250 ms)
Calibration	120 valid frames, valid 12 h
Window / minimum span	10 s / 5 s
Hold / client cooldown / server cooldown	3 s / 60 s / 45 s
Effortful z / full-scale z	1.0 / 2.0
Weights (brow, squint, innerUp, mouthPress)	0.40, 0.25, 0.20, 0.15
Distracted / fatigued fraction	>= 0.5 / >= 0.3
Away auto-pause	20 s
Threshold range / step / default	0.30-0.90 / 0.05 / 0.55

V. EVALUATION PROTOCOL

This paper describes the system design; the measurements below are planned and are not reported results.

Detector accuracy. Recruit participants to watch a mixed set of easy and hard segments. After each segment they mark confused or not confused on a short form. Compare automatic triggers against these labels using precision, recall and F1, and compare a global fixed threshold against the personalised one. Report the false-alarm rate per hour of video.

Adaptation. Track threshold trajectories over sessions and measure how many feedback events are needed before the false-alarm rate stabilises.

Explanation quality. Rate explanations on correctness, relevance to the paused moment and clarity (Likert scale), in several languages, and compare transcript-grounded explanations with the generic fallback.

Learning outcome. Run a between-group study with and without CogniLearn, using a post-video test and the generated quiz, to check whether interventions improve retention [9].

System performance. Measure end-to-end latency from trigger to spoken explanation, per-frame inference time on CPU, and CPU and memory overhead in the browser.

Ablations. Remove, in turn, per-user calibration, the hold time, the behaviour amplifier and the blink exclusion, to quantify the contribution of each.

VI. LIMITATIONS AND FUTURE WORK

- **Face-only signal.** Facial effort cues are noisy and can reflect glare, reading small text or habitual expression, not confusion. No validation against ground-truth labels is yet reported.
- **Single global memory.** The backend keeps one memory list and one threshold copy, not one per user, so it suits a single-user local setup. A per-user store is needed for multi-user deployment.
- **Transcript dependence.** Videos without captions fall back to generic tips, and auto-generated captions may contain errors. Explanations depend on text only and not on what is shown on screen.
- **Security hardening.** The prototype allows all origins in CORS and passes the Gemini key as a URL query parameter. Production use should restrict origins, send the key in a header and keep it server-side only.
- **Fixed heuristics.** Weights, window sizes and gaze margins were set by design and not learned. The logged labelled events allow future training of a small classifier.
- **Scope.** The extension targets YouTube, and the language list is fixed.

Future work includes adding a learned classifier on the logged features, multimodal context (slides and code on screen), per-user cloud-free profiles, spaced-repetition scheduling of the quiz, accessibility options and support for other video platforms.

VII. CONCLUSION

CogniLearn shows that useful, timely help for video learners can be built without uploading camera images. A per-user calibrated facial detector, a conservative hold-and-cooldown

trigger, behaviour used only as an amplifier, and a threshold that learns from real user labels together keep interruptions rare and personal. Grounding the LLM in the speaker's own transcript produces explanations that match both the moment and the language of the lesson, and the quiz and notes close the loop with retrieval practice. The prototype is modular and its labelled event log provides a path to data-driven improvement. A formal user study, as outlined in Section V, is the next step.

ACKNOWLEDGMENT

The authors thank Sandeep B, Assistant Professor, Department of Computer Science and Engineering, Jawaharlal Nehru New College of Engineering, Shivamogga. together with the Head of the Department and the Principal, for their guidance and support.

REFERENCES

[1] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257-285, 1988.

[2] P. Ekman and W. V. Friesen, *Facial Action Coding System*. Palo Alto, CA, USA: Consulting Psychologists Press, 1978.

[3] S. D'Mello and A. Graesser, "Dynamics of affective states during complex learning," *Learning and Instruction*, vol. 22, no. 2, pp. 145-157, 2012.

[4] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, "The faces of engagement: Automatic recognition of student engagement from facial expressions," *IEEE Trans. Affective Computing*, vol. 5, no. 1, pp. 86-98, 2014.

[5] C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," arXiv:1906.08172, 2019.

[6] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," arXiv:1907.06724, 2019.

[7] Gemini Team, Google, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[9] H. L. Roediger III and J. D. Karpicke, "Test-enhanced learning: Taking memory tests improves long-term retention," *Psychological Science*, vol. 17, no. 3, pp. 249-255, 2006.

[10] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419-420, 1962.