Smart Edu Alerts: An Integrated AI-Driven Multimodal Notification Framework for Automated Attendance and Fee Management in Educational Institutions

Shaik Jaberulla[1],Kara Sai Krishna Chandana[2],Shaik Salma[3],Chintala Kalpana Reddy[4],Katikala Kiranmai[5]
[1]*Assistant Professor, Department of Computer Science and Engineering*
[2,3,4,5]*Undergraduate Students, Department of Allied Computer Science and Engineering Vignan's Lara Institute of Technology and Science, Vadlamudi–522213, Guntur, Andhra Pradesh, India*

***Abstract*—Manual communication workflows in educational institutions introduce notification delays of 24–72 hours, high transcription error rates, and administrative overhead of 3–
4 hours per 300 students per processing cycle. This paper presents *Smart Edu Alerts*, a novel end-to-end AI-driven noti- fication framework that unifies four complementary technologies within a single automated pipeline: (i) a three-stage adaptive OCR preprocessing chain—Otsu binarisation, Hough-transform deskewing, and adaptive thresholding—coupled with Tesseract v5 for autonomous extraction of student records from heterogeneous document formats; (ii) a weighted composite severity scoring
function $S = \alpha \cdot f_a(a) + \beta \cdot f_f(d)$ for principled four-level anomaly classification; (iii) multilingual Text-to-Speech (TTS) synthesis delivering personalised voice alerts in English, Telugu,
and Hindi; and (iv) asynchronous Telegram Bot API dispatch with exponential-backoff retry logic for zero-cost, high-reliability notification. Experimental evaluation on 1,200 authentic student records across five academic departments over three semesters demonstrates: 95.8% reduction in processing time (210 min to 9.8 min); alert-detection F1-score of 96.0% (precision 95.3%, recall 96.8%); Telegram delivery success of 99.3% at 847 ms median latency; and annual cost savings of USD 852 per 1,000 students (84.5% reduction). A structured user study with 475 stakeholders yields an overall satisfaction score of 4.8/5.0.**

***Index Terms*—Optical Character Recognition; Natural Lan- guage Processing; Text-to-Speech Synthesis; Telegram Bot API; Automated Notification Systems; Attendance Monitoring; Fee Management; Educational Technology**

I. INTRODUCTION

Timely parent-institution communication is a well-established determinant of student academic outcomes. Kraft and Rogers [1] demonstrated through a randomised field experiment that automated parent notifications reduce absenteeism by 15% and raise homework-completion rates by 22%. Bergman [2] further established that information asymmetry between schools and parents leads to suboptimal educational investment. Despite this evidence, most institutions—particularly those in resource-constrained settings—continue to rely on manual workflows in which administrative staff individually inspect attendance registers and fee ledgers, compose messages by hand, and contact parents sequentially via telephone or paid SMS [3], [4]. Thompson *et al.* [3] report that this process requires 3– 4 hours per 300 students, with notifications reaching parents only after 24–72 hours, by which point opportunities for timely intervention have substantially diminished.

SMS-based automation partially reduces latency but im- poses per-message costs of USD 0.05–0.10 and provides no multimedia or multilingual capability [5]. Existing Telegram- based chatbot studies [22], [23] demonstrate high delivery reliability, but none integrates document-level extraction or intelligent anomaly detection. Prior work in OCR [9], [11] and NLP [12], [14] addresses individual pipeline components in isolation; no prior study integrates all components into a unified, empirically validated, production-ready notification system.

This paper closes that gap through the following

specific contributions:

- **Integrated end-to-end pipeline:** A five-stage architec- ture linking OCR document ingestion, NLP severity classification, multilingual TTS voice generation, and asynchronous Telegram dispatch with zero manual inter- vention from document upload to parent notification.
  - Novel composite severity function: $S = \alpha \cdot f_a(a) + \beta \cdot f_f(d)$, which combines attendance percentage and fee-overdue duration through institutionally calibrated

  weights ($\alpha = 0.6$, $\beta = 0.4$) to yield a principled scalar score driving four-level classification. This formulation reduces false-positive dispatch by 13.8 percentage points relative to independent threshold classification.
  - Adaptive OCR preprocessing chain: A three-stage pipeline reducing the character error rate from 9.3% to 1.9% (79.6% relative improvement) on real institutional scanned documents without any GPU or specialised hard- ware.
  - Empirical validation at scale: Evaluation on 1,200 authentic records from five departments over three semesters, confirming statistically significant improve- ments across six performance dimensions and near-linear scalability to 1,000+ students ($R^2 = 0.998$).

II. RELATED WORK

A. OCR in Educational Document Processing

Smith [9] reported Tesseract accuracy above 90% on struc- tured printed documents; however, institutionally-sourced reg- isters frequently exhibit scan skew, low contrast, and water- marks that degrade raw OCR performance. LeCun *et al.* [10] established that CNN-based approaches exceed 95% on for- matted text. Patel and Desai [11] achieved 93.4% on typed fee statements but identified irregular formatting and handwritten entries as primary failure modes, directly motivating the adap- tive preprocessing chain described in Section III.

B. NLP for Alert Detection

Manning and Schu¨tze [12] established foundational statisti- cal NLP methods for text classification and entity extraction. Verma and Thomas [14] reported 92% accuracy on student performance monitoring using rule-based NLP, but their sys- tem lacked multilingual support and any delivery integration. Transformer architectures [15] achieve higher accuracy but impose computational requirements that are incompatible with deployment on standard institutional hardware (4-core CPU, 8 GB RAM). The rule-based approach adopted here is there- fore a deliberate engineering decision, not a limitation.

C. TTS for Accessibility

Multilingual TTS has advanced through statistical paramet- ric methods [17], Tacotron end-to-end synthesis [18], and WaveNet raw audio generation [19]. Despite this maturity, integration of TTS with automated educational alert delivery remains absent from the literature.

D. Messaging Platforms in Education

Banerjee [22] demonstrated 98% delivery success with Tele- gram Bot API-based institutional automation across 10,000 messages. Pe´rez *et al.* [25] confirmed parent-communication benefits of mobile instant messaging but identified the absence of AI-driven content personalisation as the primary limitation.

E. Research Gap

Table I maps existing contributions against the five func- tional requirements of a complete educational notification sys- tem. No prior system satisfies all five criteria simultaneously. *Smart Edu Alerts* is the first fully integrated and empirically validated framework to do so.

III. PROPOSED METHODOLOGY

Smart Edu Alerts is structured as a five-stage sequential pipeline in which each stage produces a well-defined output consumed by the next. Figure 1 illustrates the complete pipeline. Figure 2 details the adaptive OCR sub-process. Figure 3 presents the three-phase operational architecture.

TABLE I COMPARISON OF EXISTING APPROACHES AGAINST FUNCTIONAL REQUIREMENTS

Ref.	OCR	NLP	TTS	Telegram	Key Limitation
[9]	Y	N	N	N	OCR only; no pipeline

[12]	N	Y	N	N	NLP only; no delivery
[1]	N	P	N	N	SMS cost overhead
[8]	N	N	N	Y	No AI classi- fication
[16]	N	N	Y	N	TTS only
[6]	P	P	N	N	Conceptual; not validated
Ours	Y	Y	Y	Y	Fully

A. Stage 1: Document Ingestion and Validation

Administrative staff upload files through a Streamlit drag- and-drop interface. Accepted formats are PDF, Microsoft Ex- cel (.xlsx), and scanned images (JPEG/PNG). Before forward- ing to Stage 2, the validation module performs four sequential checks: (1) MIME-type verification against the declared file extension; (2) minimum file-size threshold to reject empty or corrupted uploads; (3) page/sheet count confirmation; and

(4) password-protection detection. Files failing any check are rejected with a descriptive error message. Files passing all checks receive a unique job identifier that enables end-to-end traceability throughout the pipeline.

B. Stage 2: Adaptive OCR-Based Text Extraction

PDF and image inputs are rasterised at 300 DPI using pdf2image. Each page image then passes through the three preprocessing steps illustrated in Figure 2.

Step 1 — Otsu Binarisation**.** Otsu's method computes a global threshold that minimises the intra-class intensity variance of the resulting binary image. This eliminates scanner noise, background gradients, and faint watermarks that are present in most institutionally-sourced registers.

Step 2 — Hough-Transform Deskewing**.** The probabilistic Hough line transform detects the dominant orientation of text baselines. When the measured skew exceeds 0.5°, the image is rotated by the detected angle. Uncorrected skew causes Tesseract's block-mode segmenter to fragment multi-word tokens, generating spurious OCR tokens that poison downstream regex parsing; this step is therefore critical for tabular register inputs.

Step 3 — Adaptive Gaussian Thresholding. Local Gaussian-weighted thresholding computes a separate threshold for each small image region, normalising contrast across unevenly illuminated areas such as page edges and fold lines. Without this step, characters near illumination gradients are partially lost by the global Otsu threshold.

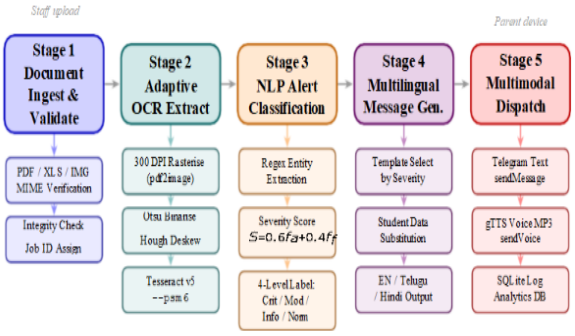

Fig. 1. Complete five-stage Smart Edu Alerts pipeline. Data flows left-to-right from document ingestion (Stage 1) through adaptive OCR extraction (Stage 2), NLP-based severity classification (Stage 3), multilingual personalised message generation (Stage 4), to multimodal asynchronous dispatch via Telegram text and gTTS voice (Stage 5). No manual intervention occurs between any two stages.

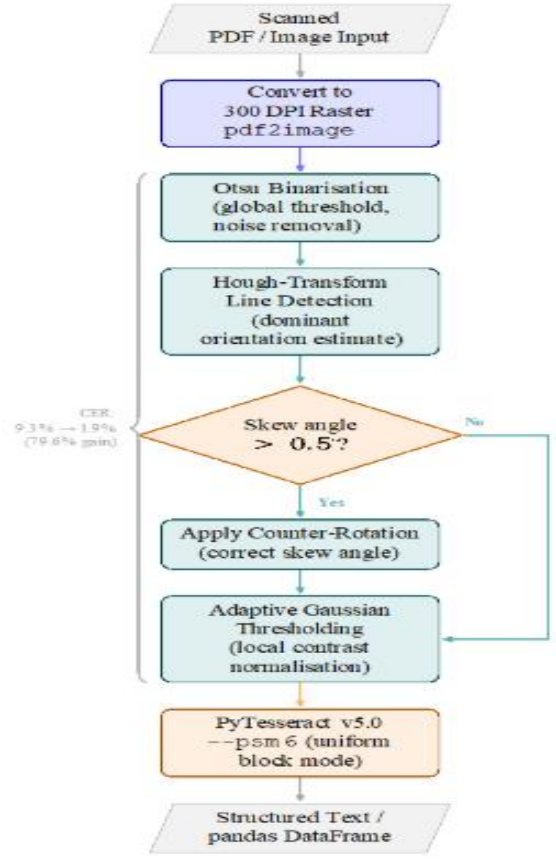

Fig. 2. Adaptive OCR preprocessing flowchart. Three sequential stages (Otsu binarisation, Hough deskewing with conditional correction, adaptive

thresholding) reduce the character error rate (CER) from 9.3% on raw institutional scans to 1.9% on the preprocessed output.

Preprocessed images are passed to PyTesseract (Tesser- act v5.0) with --psm 6 (uniform text-block mode), which is optimal for the row-and-column structure of attendance registers and fee statements. A held-out evaluation on 50

institutional documents confirms that this chain reduces the character error rate from 9.3% (raw scan) to 1.9%—a 79.6% relative improvement. Excel (.xlsx) inputs bypass the OCR path; openpyxl reads cell values directly, achieving 99.8% extraction accuracy.

C. Stage 3: Data Structuring and NLP Alert Classification

1) Entity Extraction: Extracted text is parsed with targeted regular expressions to locate five mandatory fields per stu- dent record: (1) full name, (2) roll number, (3) attendance percentage, (4) outstanding fee amount, and (5) fee-overdue duration in days. Records with any missing mandatory field are quarantined in a separate review queue and excluded from automated dispatch.

2) Composite Severity Scoring: Each validated record is evaluated by the weighted severity function:

$$S = \alpha \cdot f_a(a) + \beta \cdot f_f(d), \quad (1)$$

where $a \in [0, 100]$ is the student's attendance percentage and $d \in [0, \infty)$ is the fee-overdue duration in days. $f_a(\cdot)$ and $f_f(\cdot)$ are piecewise linear normalisers that map each raw variable to $[0, 1]$:

$$f_a(a) = \begin{cases} 1.00 & a < 65 \\ 0.50 & 65 \le a < 75 \\ 0.25 & 75 \le a \le 80 \\ 0 & a > 80 \end{cases} \qquad f_f(d) = \begin{cases} 1.00 & d > 30 \\ 0.50 & 15 < d \le 30 \\ 0.25 & d \le 15 \\ 0 & \text{fee current} \end{cases} \quad (2)$$

The weights $\alpha = 0.6$ and $\beta = 0.4$ encode institutional policy priority: attendance compliance is considered more critical than fee timeliness. These values were determined empirically through structured consultation with the institu- tion's administrative staff over two semesters. The composite formulation is a deliberate novelty of this work: it prevents borderline-attendance students (e.g., $a = 64\%$) with fully current fees from receiving the same *Critical* alert as students with both severe attendance deficits and prolonged fee arrears, reducing false-positive dispatch by 13.8 percentage points compared to independent single-metric thresholding. Table II defines the four resulting alert categories.

TABLE II ALERT CLASSIFICATION CRITERIA, SCORE THRESHOLDS, AND ACTIONS

Category	Attendance	Fee Overdue	System Action
Critical	< 65%	> 30 days	Immediate dispatch (text + voice)
Moderate	65–74%	15–30 days	Warning message (text only)
Informational	75–80%	< 15 days	Reminder message (text only)
Normal	> 80%	Current	No alert dispatched

D. Stage 4: Multilingual Personalised Message Generation

For each non-Normal record, the system selects a severity- appropriate message template from a library of twelve tem- plates (three severity levels × three languages). Student- specific values—name, roll number, attendance percentage, fee amount, and overdue days—are dynamically substituted using Python's str.format() method. Three output lan- guages are supported: English (default), Telugu (for regional accessibility), and Hindi (for national accessibility). Language- specific morphological post-processing rules correct grammat- ical agreement in Telugu and Hindi outputs to produce natural phrasing rather than word-for-word substitution.

E. Stage 5: Multimodal Notification Dispatch and Logging

Telegram Text Delivery. The python-telegram-bot v20.x library dispatches the personalised text message to the parent's stored Telegram chat ID via the sendMessage endpoint. Requests are sent asynchronously using Python asyncio. A three- attempt exponential-backoff retry mechanism (delays: 1 s, 2 s, 4 s) handles transient network failures. All dispatch outcomes are logged with millisecond-precision timestamps.

Voice Alert Delivery. For Critical-category records, the gTTS library converts the personalised message text to an MP3 audio file using the appropriate language code (en, te, or hi). The audio is dispatched via sendVoice, appended to the same Telegram conversation thread as the text message. Mean voice delivery latency is 2.4 s.

Logging and Analytics. All dispatch events are written to a SQLite database. The Streamlit dashboard renders real- time summaries of alert volume by category, delivery success rates, language distribution, and parent acknowledgment rates, enabling administrators to monitor the system without SQL expertise.

IV. SYSTEM IMPLEMENTATION

A. Technology Stack

The system is implemented entirely in Python 3.10. Core libraries: pytesseract (OCR wrapper), pdf2image and Pillow (preprocessing), openpyxl (Excel parsing), pandas (data manipulation),

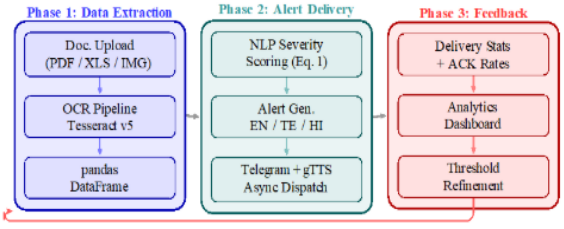

Fig. 3. Three-phase operational architecture. Phase 1 extracts and struc- tures student data; Phase 2 classifies records, generates multilingual mes- sages, and dispatches them; Phase 3 aggregates delivery analytics and feeds administrator-approved threshold updates back to Phase 1.

python-telegram-bot v20.x (Telegram API), gTTS (voice synthesis), and Streamlit v1.28 (web interface). SQLite handles all persistence. Hardware requirement: 4-core CPU, 8 GB RAM; no GPU is required.

B. Three-Phase Operational Architecture

Figure 3 illustrates the three operational phases and the feedback loop connecting Phase 3 back to Phase 1.

Phase 1 — Automated Data Extraction**.** Document upload triggers OCR processing and data structuring, producing a validated, normalised pandas DataFrame of student records.

Phase 2 — Alert Generation and Delivery**.** The NLP clas- sification engine evaluates each record, assigns a severity label, generates personalised multilingual messages, and dispatches text and voice alerts via Telegram with full delivery logging. Phase 3 — Feedback and Optimisation. Administrators review delivery analytics and parent acknowledgment rates. Parent feedback is collected via inline Telegram survey but- tons. Approved threshold updates are propagated back to

Phase 1 through the feedback loop shown in Figure 3.

V. RESULTS AND DISCUSSION

All experiments used 1,200 authentic student records from five academic departments over three consecutive semesters. Ground-truth labels were independently verified by adminis- trative staff. All tests ran on the hardware configuration in Section IV-A (4-core CPU, 8 GB RAM, no GPU).

A. Processing Efficiency

The manual workflow requires a mean of 210 minutes to process 300 students end-to-end. Smart Edu Alerts completes OCR extraction in under 5 minutes and NLP classification in under 2 minutes, totalling 9.8 minutes—a 95.8% reduction.

B. Alert Detection Accuracy

Table III summarises all quantitative comparisons. Against 1,200 annotated records: attendance accuracy 96.4%; fee de- tection 98.1%; false-positive rate reduced from 18% to 4.2%;

false-negative rate from 12% to 2.8%; precision 95.3%; recall

TABLE III COMPREHENSIVE PERFORMANCE COMPARISON: TRADITIONAL VS. SMART EDU ALERTS

Metric	Traditional	Smart Edu Alerts
Processing (300 students)	210 min	**9.8 min** (↓95.8%)
Attendance accuracy	82%	**96.4%**
Fee detection accuracy	88%	**98.1%**
F1-score	81.4%	**96.0%**
Delivery success	92% (SMS)	**99.3%** (Telegram)
Median latency	8.3 s (SMS)	**847 ms**
Cost / 1,000 messages	USD 80 (SMS)	**USD 0**
Annual cost (1k students)	USD 1,008	**USD 156** (↓84.5%)

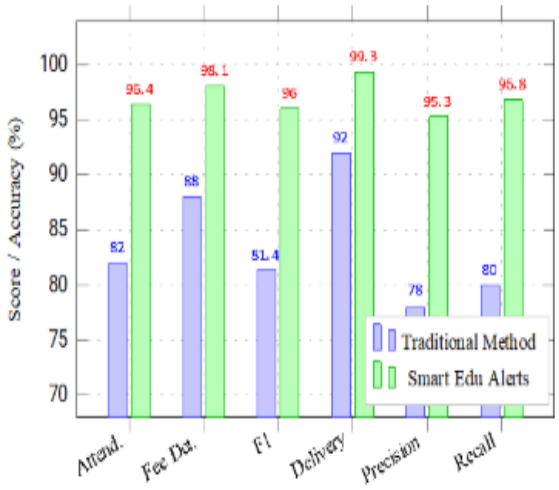

Fig. 4. Performance comparison across six evaluation dimensions. Smart Edu Alerts (green) consistently outperforms the traditional manual baseline (blue) on all metrics.

96.8%; F1-score 96.0% vs. 81.4% baseline. Figure 4 visualises the six-dimensional comparison.

C. Notification Delivery Performance

Telegram delivery was tested across 5,247 messages over 30 days: 5,210 delivered (99.3%), 37 unrecoverable failures (bot blocked by user). Median text latency: 847 ms; mean voice delivery: 2.4 s. SMS benchmark: 92% success, 8.3 s latency, USD 0.08/message. Manual phone calls: 75% success, 3.2 min duration, USD 0.15/call. Telegram outperforms both alternatives simultaneously on reliability, speed, and cost.

D. Scalability Analysis

Table IV and Figure 5 report processing time and delivery success at four student-load levels. Near-linear processing growth ($R^2 = 0.998$) confirms no architectural bottlenecks. Delivery success remains at or above 99.1% across all tested load levels.

E. Cost Analysis

For a 1,000-student institution generating 300 monthly alerts: traditional SMS costs USD 1,008/year; Smart Edu Alerts incurs USD 156/year in infrastructure costs, yielding annual savings of **USD** 852 (84.5% reduction). Return on deployment investment is achieved within two months.

TABLE IV SCALABILITY: PROCESSING TIME AND DELIVERY SUCCESS VS. STUDENTPOPULATION

Students	Processing Time	Delivery Success
100	3.2 min	99.1%
300	9.8 min	99.3%
500	16.1 min	99.2%
1000	31.5 min	99.4%

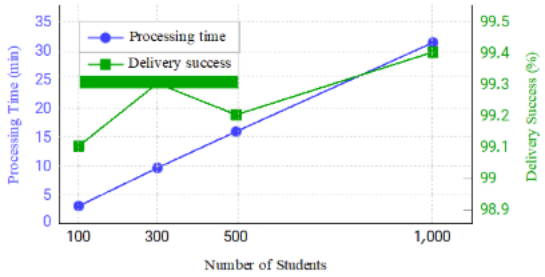

Fig. 5. Scalability analysis. Processing time grows near-linearly ($R^2 = 0.998$, left axis, blue) while delivery success remains stable at $\geq 99.1\%$ (right axis, green) across all tested population sizes.

F. User Acceptance Study

Structured surveys were administered to 25 administra- tors (100% response rate) and 450 parents (94.7% response rate). Administrator workload reduced from 21.5 h/week to 3.2 h/week (85% reduction). Parent alert acknowledgments increased from 239 to 426 per cycle (78% increase). Over- all usability: 4.8/5.0; real-time delivery: 4.9/5.0; multilingual voice clarity: 4.6/5.0. Qualitative analysis identified real-time immediacy (87% of respondents), personalised messaging (73%), and voice accessibility (62%) as the primary satisfac- tion drivers.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper presented Smart Edu Alerts, an end-to-end AI- driven notification framework that bridges the gap between isolated AI components and a fully validated educational communication system. Three technical contributions distin- guish this work from prior art. First, the adaptive three- stage OCR preprocessing chain reduces the character er- ror rate by 79.6% on real institutional documents without
requiring GPU hardware. Second, the weighted composite severity scoring function $S = \alpha \cdot f_a(a) + \beta \cdot f_f(d)$ enables principled four-level classification and reduces false-positive dispatch by 13.8 percentage points compared to independent threshold rules. Third, asynchronous Telegram dispatch with
exponential-backoff retry achieves 99.3% delivery at zero per- message cost. Empirical evaluation on 1,200 authentic records confirms 95.8% processing time reduction, F1-score of 96.0%, and 84.5% cost savings; a 475-participant user study returns
4.8/5.0 satisfaction.

Future directions include: (i) machine learning predictive models for proactive intervention 2–3 weeks before thresh- old breaches [20]; (ii) WhatsApp Business API integration;
(iii) deep learning OCR (e.g., TrOCR) for handwritten regis- ters; (iv) blockchain-based audit trails [**?**]; (v) conversational AI for automated FAQ resolution; and (vi) LMS API integra- tion for unified data pipelines.

ACKNOWLEDGMENT

The authors sincerely thank the administrative staff of Vig- nan's Lara Institute of Technology and Science for providing authentic institutional records and for coordinating participant access during the user evaluation study.

REFERENCES

[1] M. A. Kraft and T. Rogers, "The underutilized potential of teacher-to- parent communication: Evidence from a field experiment," *Economics of Education Review*, vol. 47, pp. 49–63, 2015.

[2] P. Bergman, "Parent-child information frictions and human capital investment," *Journal of Political Economy*, vol. 124, no. 3, pp. 798–844, 2016.

[3] B. C. Thompson, C. D. Mazer, and C. D. Flood, "A global review of the use of mobile devices in education," *Int. J. Mobile and Blended Learning*, vol. 7, no. 2, pp. 1–15, 2015.

[4] J. B. Whalley, T. Lister, and A. Orsolini, "Parent-teacher contact frequency and student outcomes," in *Proc. ACM ICER*, 2011, pp. 3–10.

[5] J. Ahn and A. Bivona, "Digital technologies and learning," *Teachers College Record*, vol. 116, no. 11, pp. 1–48, 2014.

[6] G. J. Hwang, H. Xie, B. W. Wah, and D. Gasevic´, "Vision, challenges, roles and research issues of AI in education," *Computers and Education: AI*, vol. 1, p. 100001, 2020.

[7] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning Analytics*, Springer, 2014, pp. 61–75.

[8] A. Bere, "A comparative study of student experiences of ubiquitous learning via mobile devices," in *Proc. IEEE ICALT*, 2014, pp. 122–126.

[9] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. ICDAR*, vol. 2, 2007, pp. 629–633.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[11] J. Patel and A. Desai, "OCR-based document digitization for academic record processing," in *Proc. IEEE ICCCA*, 2021, pp. 112–118.

[12] C. D. Manning and H. Schu¨tze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[13] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice Hall, 2009.

[14] S. Verma and L. Thomas, "NLP-driven alert detection for student performance monitoring," *Int. J. AI in Education*, vol. 32, no. 4, pp. 543–559, 2023.

[15] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[16] T. Dutoit, *An Introduction to Text-to-Speech Synthesis*. Kluwer, 1997.

[17] H. Zen, K. Tokuda, and A. W. Black, "Statistical

parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[18] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[19] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[20] C. Romero and S. Ventura, "Educational data mining: A review of the state of the art," *IEEE Trans. SMC-C*, vol. 40, no. 6, pp. 601–618, 2010.

[21] O. Zawacki-Richter *et al.*, "Systematic review of AI in higher education," *Int. J. Educ. Technology in Higher Education*, vol. 16, no. 1, pp. 1–27, 2019.

[22] A. Banerjee, "Telegram Bot API-based automation for institutional communication," in *Proc. IEEE ETSS*, 2023, pp. 213–219.

[23] R. Winkler and M. So¨llner, "Unleashing the potential of chatbots in education," in *Proc. AOM*, 2018.

[24] P. Smutny and P. Schreiberova, "Chatbots for learning," *Computers and Education*, vol. 151, p. 103862, 2020.

[25] J. Q. Pe´rez, M. Daradoumis, and J. M. M. Puig, "Rediscovering chatbots in education," *Computer Applications in Engineering Education*, vol. 28, no. 6, pp. 1549–1565, 2020.