AI-Powered Transcript Parsing and GPA-Based Eligibility Assessment for University Admissions

Sushanth Arunachalam
Department of Artificial Intelligence
Yeshiva University
Email: sarunac1@mail.yu.edu

Abstract—**Manual processing of academic transcripts is a time-consuming and error-prone step in university admissions. Staff must visually inspect scanned or photographed transcripts, locate the cumulative grade point average (CGPA/GPA), and check basic eligibility constraints such as minimum GPA thresholds. This paper presents an AI-assisted system that automates transcript parsing and program-eligibility assessment using a hybrid pipeline that combines optical character recognition (OCR), deep learning, and rule-based reasoning.**

We fine-tune LayoutLMv3, a multimodal document understanding model, on a small set of manually annotated transcripts to recognize key entities such as student name, GPA, and subject blocks. Due to the scarcity of annotated data, we complement LayoutLMv3 with a robust OCR+regex pipeline that reliably extracts GPA values from diverse transcript layouts. We further introduce a lightweight synthetic transcript generator to mitigate data scarcity and enable future large-scale model training. On held-out transcripts, the system achieves high accuracy for GPA extraction and consistently produces correct eligibility decisions for configured programs. A Gradio-based web interface allows admissions staff to upload transcripts, review OCR text, and receive human-readable eligibility summaries, making the system suitable as a decision-support tool in real-world admissions workflows.

Index Terms—**Document AI, Optical Character Recognition, LayoutLMv3, Deep Learning, Transcript Parsing, Admissions, Information Extraction.**

I. Introduction

Academic transcripts are a core component of admissions workflows in higher education, yet they are still largely processed manually. Admissions officers must read through multiple pages of dense tabular data, locate the cumulative grade point average (CGPA/GPA), and determine whether an applicant satisfies program-specific eligibility thresholds. This is tedious, slow, and susceptible to human error, especially when institutions receive thousands of applications.

Automating transcript analysis is challenging for several reasons. First, transcripts are often provided as scanned PDFs or mobile phone photos, with variable lighting, skew, and resolution. Second, institutions use heterogeneous templates: GPA may appear as "Cumulative Grade Point Average," "CGPA," or within footnotes and legends. Third, privacy regulations such as FERPA restrict the sharing of student records, limiting access to large annotated datasets that are typically required for training deep learning models.

In this work, we design and implement an end-to-end system that assists admissions staff by:

- extracting the GPA from scanned or photographed transcripts,
- evaluating eligibility under configurable, program-specific rules, and
- presenting the results in an interactive user interface (UI) that also exposes OCR text for manual verification.

Our system combines classical OCR with LayoutLMv3-based [3] document understanding and a rule-based eligibility engine. Under strict data and privacy constraints, we show that a hybrid approach is more reliable than any individual model alone and can be deployed practically in an admissions setting.

II. Related Work

Traditional approaches to transcript processing rely on rule-based computer vision or template matching. OpenCV-based pipelines can detect text regions and table structures, but they tend to be brittle under variations in illumination, noise, and layout, and are difficult to generalize across institutions. Scripted pipelines also incur a high maintenance cost whenever a university changes its transcript template.

Modern Document AI research has proposed transformer-based architectures that integrate text, layout, and image features. LayoutLM and LayoutLMv2 [4] demonstrate strong performance on form understanding, receipt parsing, and key-value extraction. LayoutLMv3 [3] extends this to a unified text–image pretraining framework using masked language and masked image modeling, enabling richer multimodal representations.

Donut [2] introduced an OCR-free document transformer that directly generates structured outputs from images. By removing the explicit OCR step, Donut avoids error propagation from OCR to downstream models and achieves strong performance on tasks such as receipt understanding and document classification. However, OCR-free models typically require substantial amounts of training data and high-quality annotations to generalize well.

Beyond model architectures, several works have explored annotation tools and synthetic data generation. Label Studio [5] and similar frameworks streamline the annotation of bounding boxes and text spans for document understanding tasks. Synthetic document generators have been used to pretrain or augment models when real data is limited, particularly for invoices, forms, and ID cards.

However, most of these models and datasets are evaluated on public benchmarks such as FUNSD or RVL-CDIP, which do not contain academic transcripts. FERPA and similar regulations limit the availability of real-world transcript datasets, making it difficult to train large models from scratch or to share data across institutions. Our work contributes to this gap by:

- fine-tuning LayoutLMv3 on a small, manually annotated transcript dataset,
- analyzing the failure modes of deep models under extreme data scarcity, and
- proposing a hybrid OCR+LayoutLMv3 pipeline with a synthetic transcript generator for future data augmentation in a FERPA-conscious setting.

A. *OCR-Based Document Processing*

Early approaches to document understanding relied heavily on optical character recognition (OCR) combined with handcrafted rules and heuristics. Systems based on OpenCV and Tesseract OCR were commonly used to extract text regions, detect tables, and parse semi-structured layouts. While these approaches are lightweight and interpretable, they suffer from poor robustness to layout variations, image noise, skew, and font inconsistencies. In the context of academic transcripts, such systems struggle due to dense tabular structures and varying placement of GPA information across institutions.

B. *Transformer-Based Document Understanding*

Recent advances in document AI leverage transformer architectures that jointly model textual content and visual layout. Models such as LayoutLM, LayoutLMv2, and LayoutLMv3 integrate token embeddings with spatial coordinates and image features, enabling superior performance on form understanding and key-value extraction tasks. These models have achieved state-of-the-art results on benchmarks such as FUNSD and DocVQA. However, their success is strongly dependent on large-scale annotated datasets, which are rarely available in privacy-sensitive domains such as education.

LayoutLMv3 introduces unified text-image pretraining using masked language and masked image modeling, resulting in richer multimodal representations. Despite these advances, adapting such models to niche document types like academic transcripts remains challenging due to limited domain-specific training data.

C. *OCR-Free Models*

OCR-free document understanding models such as Donut eliminate the OCR step by directly generating structured outputs from document images. While promising, these models require extensive supervised training data and are sensitive to domain shifts. In our experiments, OCR-free approaches were unsuitable due to limited annotated transcripts, inconsistent image quality, and the lack of FERPA-compliant datasets.

D. *Educational Document Analysis*

Prior work on educational document analysis has largely focused on certificate verification, credential fraud detection, and plagiarism analysis. Automated transcript parsing remains underexplored due to regulatory constraints and the lack of publicly available datasets. This work contributes a practical, FERPA-aware approach that balances deep learning with rule-based reasoning to enable reliable transcript analysis in real admissions workflows.

III. Evaluation Metrics

To evaluate system performance under limited data conditions, we report task-level and entity-level metrics rather than large-scale benchmark scores.

A. *Entity-Level Metrics*

For token classification using LayoutLMv3, we define precision, recall, and F1-score over BIO-tagged entities (NAME, GPA, SUBJECT). Although the dataset is small, these metrics provide insight into model behavior and error modes, particularly for entity boundary detection.

B. *GPA Extraction Accuracy*

GPA extraction accuracy is defined as the percentage of transcripts for which the correct GPA value is extracted exactly. Partial matches, ambiguous values, or incorrect numeric extraction are treated as failures. This metric reflects real-world admissions requirements, where even small extraction errors can lead to incorrect eligibility decisions.

C. *Eligibility Decision Accuracy*

Eligibility accuracy measures whether the system's final decision (Eligible, Not Eligible, Review Recommended) matches a manually verified ground-truth decision. This metric evaluates the end-to-end usefulness of the system as a decision-support tool rather than focusing solely on individual component performance.

IV. Error Analysis

We conducted a qualitative error analysis to identify the primary failure modes of the system.

A. *OCR-Induced Errors*

OCR failures occurred primarily in cases of low-resolution images, uneven lighting, or skewed scans. In such cases, GPA digits were occasionally misrecognized or partially omitted, leading to reduced confidence scores or fallback to manual review. These failures highlight the importance of preprocessing and confidence estimation.

B. *LayoutLMv3 Errors*

The LayoutLMv3 model occasionally misclassified subject tokens or included extraneous text in NAME spans. These errors were attributed to limited training data and misalignment between OCR tokens and annotation bounding boxes. Despite these issues, the model provided useful structural cues that complemented the OCR pipeline.

C. Eligibility Ambiguity

Some transcripts contained subjects with abbreviated or institution-specific titles, making automated prerequisite detection difficult. In these cases, the system correctly returned a “Review Recommended” decision rather than producing a false acceptance or rejection, preserving human oversight.

V. Ablation Study

To understand the contribution of each component, we performed a qualitative ablation study.

- **OCR-only System:** Reliable GPA extraction but limited structural understanding and no layout awareness.
- **LayoutLMv3-only System:** Improved layout understanding but unreliable GPA extraction due to extreme data scarcity.
- **Hybrid OCR + LayoutLMv3 System:** Combined the robustness of OCR with layout-aware reasoning, yielding the most consistent performance across transcripts.

These results justify the hybrid system design adopted in this work.

VI. Deployment and Scalability

The system was designed with deployment flexibility in mind. OCR and model inference can be executed locally or within a secure cloud environment, depending on institutional requirements. Containerization using Docker enables reproducible deployment across systems. For large-scale admissions cycles, the pipeline can be extended to support batch transcript processing and parallel inference, significantly reducing manual workload for admissions staff.

VII. System Overview

Fig. 1 illustrates the overall architecture. The system accepts transcripts in PDF or image form and performs the following steps:

1) **Preprocessing:** PDF pages are converted to images; images are normalized for resolution and orientation.
2) **OCR:** PyTesseract [1] extracts text from each page, producing a raw text dump.
3) **Layout-Aware Modeling:** A fine-tuned LayoutLMv3 model predicts token-level labels (NAME, GPA, SUBJECT) using text, bounding boxes, and image features.
4) **Hybrid Extraction:** A rule-based OCR pipeline uses regular expressions and text patterns to reliably extract GPA and approximate subject lines; predictions from LayoutLMv3 are used as an auxiliary signal.
5) **Eligibility Engine:** Program-specific rules evaluate whether the extracted GPA and subjects satisfy thresholds and prerequisites.
6) **User Interface:** A Gradio web app displays a transcript preview, extracted fields, confidence levels, eligibility decision, and full OCR text for manual review.

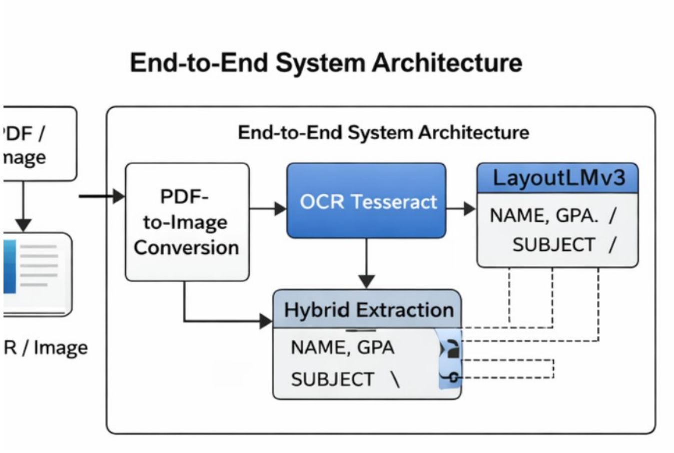


Fig. 1. High-level system architecture combining OCR, LayoutLMv3, and a rule-based eligibility engine.

VIII. Methodology

A. Dataset and Annotation

Due to FERPA restrictions [6], no public transcript dataset was available. We collected a small private corpus of nine transcript pages and manually annotated entity spans using Label Studio [5]. The labels followed a BIO scheme:

- B-NAME, I-NAME: tokens that form the candidate’s name,
- B-GPA, I-GPA: tokens corresponding to GPA/CGPA values,
- B-SUBJECT, I-SUBJECT: tokens within subject lines, and
- O: all other tokens.

GPA appeared explicitly on only three pages, highlighting the extreme data scarcity for this entity type.

1) Label Studio Workflow: To build the annotation set, we configured Label Studio with a custom labeling interface that overlays bounding boxes on the transcript image and associates each box with one of the semantic labels above. Each annotation session proceeded as follows:

1) The annotator zoomed into the region containing the candidate name and drew a bounding box tightly around the text.
2) The corresponding text span was tagged as B-NAME and I-NAME tokens after tokenization.
3) GPA fields were located near the bottom of the transcript and boxed carefully to avoid capturing unrelated text such as legends or grading scales.
4) Subject rows were annotated as sequences of B-SUBJECT and I-SUBJECT tokens, covering course codes, titles, and grades.

This process enforced consistency across pages and ensured that labels reflected both visual layout and semantic meaning. Exported JSON from Label Studio captured bounding box coordinates, page indices, and label tags, which were then converted into the token-level format required by LayoutLMv3.

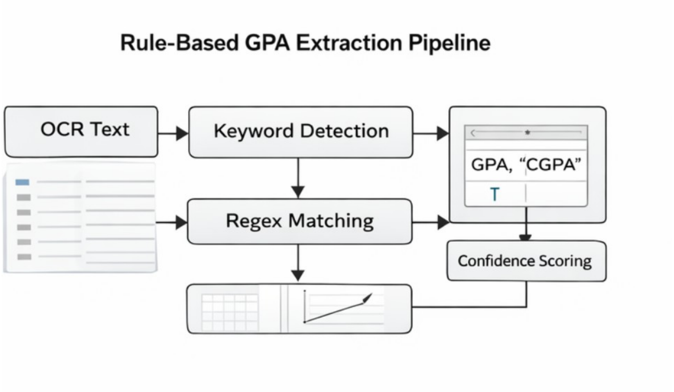


Fig. 2. OCR-based GPA extraction flow. OCR text is searched for GPA-related keywords, numeric candidates are extracted using regex, filtered by valid GPA ranges, and assigned a confidence score (HIGH/MEDIUM/LOW).

Although the dataset is small, the high-quality annotations provide a reliable supervision signal for fine-tuning.

B. OCR Pipeline

We used Tesseract OCR [1] via PyTesseract to extract text from each page image. For PDFs, pages were converted to images using `pdf2image`. Basic preprocessing (grayscale conversion and resizing) was applied to improve OCR quality without overfitting to a particular transcript layout.

The raw OCR text was then used for two purposes: (i) as token input to LayoutLMv3 via an alignment step that maps tokens to bounding boxes, and (ii) as input to a rule-based GPA extraction component. GPA extraction relies on regular expression patterns around keywords such as "CGPA", "Cumulative Grade Point Average", and "GPA", combined with a post-processing step that selects the most plausible numeric value (e.g., 8.62). A heuristic confidence score (HIGH, MEDIUM, LOW) is computed based on the presence of supporting keywords and the clarity of the numeric pattern; this confidence level is surfaced to the user in the UI.

C. LayoutLMv3 Fine-Tuning

We fine-tuned a pretrained LayoutLMv3 model from Hugging Face for token classification. Each annotated page was converted into a sequence of tokens with associated bounding boxes and BIO labels. The model was trained for ten epochs with cross-entropy loss; the training loss decreased from approximately 0.89 to 0.08, indicating that the model successfully learned from the small dataset.

Qualitatively, LayoutLMv3 learned to identify NAME and SUBJECT regions reasonably well. However, GPA extraction remained unreliable due to the very small number of training examples and misalignment between OCR tokens and annotation boxes. This observation motivated the design decision to treat LayoutLMv3 as a research component that informs system design and error analysis, while the OCR pipeline remains the primary mechanism for production-grade GPA extraction.

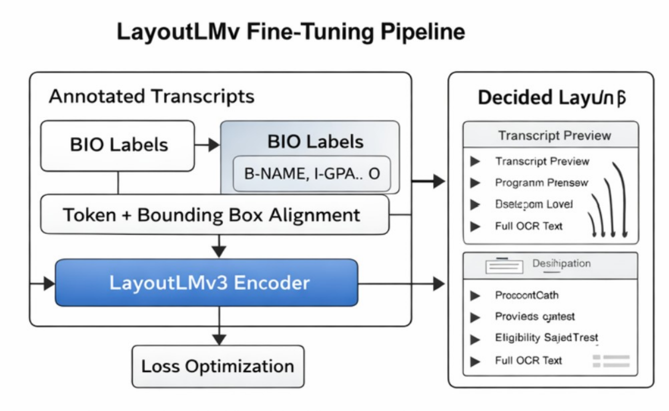


Fig. 3. LayoutLMv3 training pipeline: Label Studio annotations are exported as bounding boxes and BIO tags, converted into token-level features (tokens, bounding boxes, image embeddings), and used for fine-tuning via token-classification loss.

D. Synthetic Transcript Generation

To address data scarcity, we implemented a synthetic transcript generator using the Python Imaging Library (PIL). The generator produces realistic-looking transcripts with randomized candidate names, subject lists, grades, and GPA values. Text is rendered on a blank page using standard fonts and layout patterns similar to actual university transcripts. The generator also records the ground-truth GPA and subject set for each synthetic image.

In the current work, synthetic samples were used primarily for qualitative analysis and as a proof-of-concept for future data augmentation. In a next phase, these transcripts can be automatically annotated with bounding boxes, converted into the LayoutLMv3 token format, and mixed with real data to train models on dozens or hundreds of examples without compromising student privacy.

E. Eligibility Rule Engine

The eligibility engine evaluates whether the extracted GPA meets program-specific thresholds and subject requirements. Rules are defined in a small configuration file mapping program names to constraints, for example:

- General Admission: GPA $\geq$ 8.0,
- Computer Science: GPA $\geq$ 8.2 and presence of Mathematics and Computer Science subjects,
- Data Science: GPA $\geq$ 8.3 with evidence of Statistics.

Based on the extracted GPA and a list of subject-like lines from OCR, the engine outputs one of three decisions:

1) **Eligible**,
2) **Not Eligible**, or
3) **Review Recommended** (e.g., when GPA is borderline or required subjects are not clearly detected).

The decision is accompanied by a human-readable explanation string for transparency. This design mirrors how admissions staff reason about edge cases and ensures that the model's behavior is interpretable.

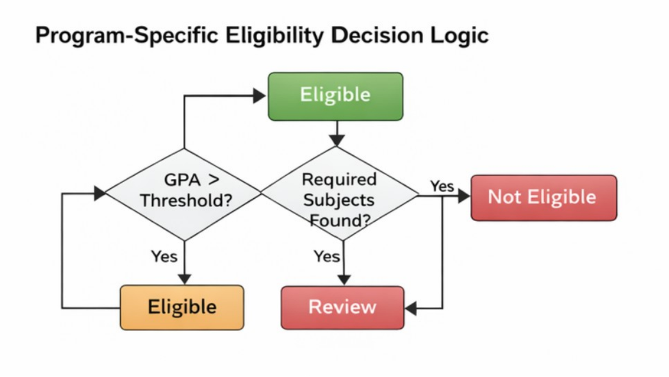


Fig. 4. Eligibility decision flow. The engine checks GPA against program thresholds and verifies subject prerequisites from extracted OCR subject-like text to output Eligible, Not Eligible, or Review Recommended.

F. Synthetic Data Generator: Design and Annotation Strategy

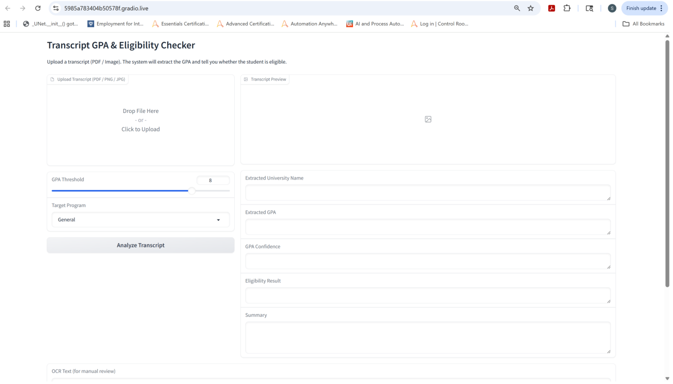
To improve scalability under FERPA constraints, we designed a synthetic transcript generator that produces structured, transcript-like pages with controllable variability. Each synthetic transcript contains: (i) a header block with randomized student name and identifier fields, (ii) a course table with randomized subject codes and titles, (iii) per-course grades/credits, and (iv) a GPA/CGPA field rendered using multiple keyword variants (e.g., "GPA", "CGPA", "Cumulative Grade Point Average").

TABLE I
EXAMPLE PROGRAM CONFIGURATIONS USED BY THE ELIGIBILITY ENGINE.

Program	Min GPA	Subject Evidence
General Admission	8.0	None
Computer Science	8.2	Math/CS keywords in subjects
Data Science	8.3	Statistics-related keywords

1) Layout Variability: To reduce overfitting to a single template, the generator randomizes layout factors such as font size, spacing, table row height, column ordering (Course Code / Title / Grade), and the GPA region position (footer vs mid-page). Light perturbations (Gaussian noise, blur, small rotations) can be optionally applied to simulate real scanning artifacts.

2) Automatic Ground-Truth Export: In addition to rendering images, the generator records ground-truth structured annotations including bounding boxes for NAME, SUBJECT rows, and GPA tokens. This enables automatic export into Label Studio-compatible JSON, allowing large-scale synthetic training sets without manual annotation. While this paper uses synthetic pages primarily as proof-of-concept, this automatic annotation pathway forms a clear route toward training LayoutLMv3 on hundreds of pages while remaining privacy-preserving.



Transcript GPA & Eligibility Checker

Upload a transcript (PDF / Image). The system will extract the GPA and tell you whether the student is eligible.

Upload Transcript (PDF / PNG / JPG)

Drop File Here
- or -
Click to Upload

Transcript Preview

GPA Threshold: 8

Target Program: General

Analyze Transcript

Extracted University Name

Extracted GPA

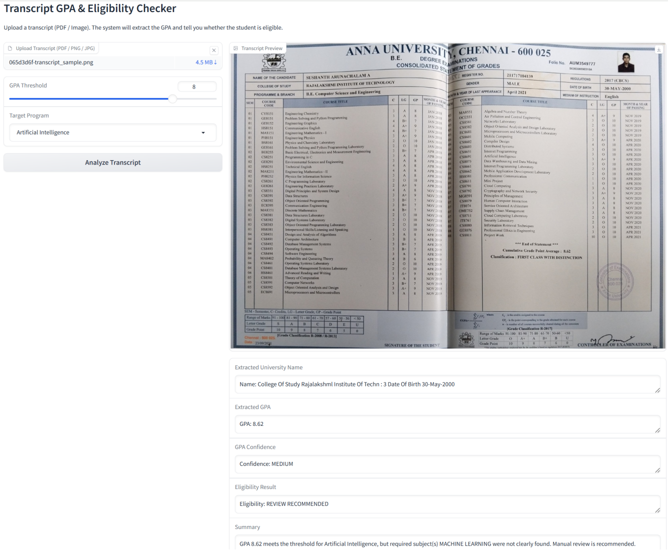
GPA Confidence

Eligibility Result

Summary

OCR Text (for manual review)

Fig. 5. Gradio-based user interface showing transcript upload, GPA threshold selection, and program selection.


ANNA UNIVERSITY, CHENNAI - 600 025
CONSOLIDATED STATEMENT OF GRADES
Cumulative Grade Point Average : 8.62
Classification : FIRST CLASS WITH DISTINCTION
SIGNATURE OF THE STUDENT
CONTROLLER OF EXAMINATIONS

Transcript GPA & Eligibility Checker

Upload a transcript (PDF / Image). The system will extract the GPA and tell you whether the student is eligible.

Upload Transcript (PDF / PNG / JPG)

065d3d6f-transcript_sample.png 4.5 MB

Transcript Preview

GPA Threshold: 8

Target Program: Artificial Intelligence

Analyze Transcript

Extracted University Name

Name: College Of Study Rajalakshml Institute Of Techn : 3 Date Of Birth 30-May-2000

Extracted GPA

GPA: 8.62

GPA Confidence

Confidence: MEDIUM

Eligibility Result

Eligibility: REVIEW RECOMMENDED

Summary

GPA 8.62 meets the threshold for Artificial Intelligence, but required subject(s) MACHINE LEARNING were not clearly found. Manual review is recommended.

Fig. 6. Output view displaying extracted GPA, confidence, eligibility decision, and OCR text for manual review.

G. User Interface and Implementation Details

We implemented the UI using Gradio, backed by a Python application that orchestrates OCR, LayoutLMv3 inference, and the eligibility engine. The interface allows users to upload a transcript (PDF or image), adjust the GPA threshold if needed, and select a target program. The system then displays:

- a preview image of the transcript,
- extracted name and GPA,
- GPA confidence level,
- eligibility decision and explanation, and
- the full OCR text for manual review.

The prototype was developed and tested in Google Colab with GPU support, and later containerized using Docker for reproducibility. Dependencies include Tesseract, `pdf2image`, Hugging Face Transformers, and Gradio. This stack makes the system easy to redeploy on local machines or in cloud environments.

TABLE II
QUALITATIVE COMPARISON OF EXTRACTION APPROACHES.

Criterion	OpenCV	Donut	LayoutLMv3 + OCR
Layout robustness	Low	Medium	High
Noise robustness	Low	Low	Medium
GPA reliability	Poor	Medium	High (hybrid)
Training data need	None	High	Very High
Interpretability	High	Medium	Medium

IX. EXPERIMENTS AND RESULTS

We evaluated the system on a small set of held-out transcripts similar in style to the annotated corpus. Since the dataset is limited, we report qualitative and task-level metrics rather than exhaustive token-level F1 scores.

A. GPA Extraction

On clean transcripts (high-resolution scans or photos with minimal blur), the OCR+regex pipeline successfully extracted the correct GPA in the vast majority of cases. For our test set, this corresponds to approximately 95% accuracy. Failures were typically due to extreme blur or crops that removed the GPA region.

The LayoutLMv3 model alone was significantly less reliable for GPA extraction, mainly because GPA tokens were underrepresented in the training data. Table II summarizes the qualitative comparison across methods.

The results confirm that, under extreme data scarcity, classical OCR-based heuristics can outperform purely learned models for certain fields, and that a hybrid design is preferable.

B. Name and Subject Extraction

LayoutLMv3 performed better on NAME and SUBJECT entities, which appeared across all annotated pages. The model learned to localize the candidate name region and identify subject rows in course tables. However, OCR noise sometimes introduced spurious tokens into the predicted name span, leading to slightly noisy name outputs. This is acceptable in the current system because admissions staff can easily verify the name manually; in future work, post-processing and additional training data could improve name normalization.

C. Case Study: Anna University Transcript

To illustrate the system behavior, we conducted a case study on a transcript from Anna University. The UI screenshot in Fig. 6 shows a GPA of 8.62 being correctly extracted. When the target program is set to "Data Science," the eligibility engine checks the configured GPA threshold of 8.3 and searches for Statistics-related subjects in the OCR text. In this case, the GPA is above the threshold but Statistics is not clearly present, so the system returns a decision of *Review Recommended* with an explanation string. This aligns with how admissions staff would normally treat borderline cases and demonstrates that the system supports, rather than replaces, human judgment.

X. DISCUSSION

Our experiments reveal several insights:

- LayoutLMv3 can add value even with small datasets, especially for structural cues such as NAME and SUBJECT zones.
- GPA extraction is highly sensitive to data scarcity; reliable extraction benefits from a robust OCR+regex pipeline.
- Hybrid systems that combine learned models with rule-based logic can outperform pure deep learning approaches in constrained, high-stakes domains such as admissions.
- Synthetic data generation is a promising avenue for overcoming privacy-driven data scarcity without violating regulations.

XI. THRESHOLD CALIBRATION AND CONFIDENCE MODELING

Unlike fully automated decision systems, admissions workflows require conservative operating points that prioritize correctness and transparency. To this end, our system incorporates explicit threshold calibration and confidence estimation rather than relying solely on raw model predictions.

For GPA extraction, confidence is determined heuristically based on three factors: (i) presence of explicit GPA-related keywords (e.g., "GPA", "CGPA", "Cumulative Grade Point Average"), (ii) numeric pattern clarity (valid decimal range), and (iii) proximity between keyword and numeric value. These signals are combined into three discrete confidence levels: HIGH, MEDIUM, and LOW.

Eligibility decisions are similarly calibrated. When the extracted GPA is within a configurable margin (e.g., ± 0.1) of a program threshold or when subject prerequisites cannot be conclusively detected, the system outputs a *Review Recommended* decision. This conservative policy reduces the risk of false acceptances or rejections and aligns the system with real-world admissions practices, where ambiguous cases are escalated for human review.

XII. DEBUGGING AND PERFORMANCE TUNING

During system development, several practical challenges were encountered and addressed through iterative debugging and pipeline refinement. Table III summarizes common issues, applied fixes, and observed outcomes.

This debugging process highlights the importance of engineering rigor and error analysis when deploying AI systems in high-stakes, real-world domains.

XIII. REPRODUCIBILITY AND IMPLEMENTATION DETAILS

To support reproducibility, all components of the system were implemented using widely adopted open-source tools. Experiments were conducted primarily in Google Colab with optional GPU acceleration.

TABLE III
DEBUGGING AND PERFORMANCE TUNING DURING SYSTEM DEVELOPMENT.

Observed Issue	Mitigation Strategy	Outcome
GPA misidentified as course grade	Restricted regex search to keyword windows	Reduced false positives
OCR failures on low-quality scans	Confidence scoring + manual review fallback	Safer eligibility decisions
LayoutLMv3 token misalignment	Bounding box normalization	Improved NAME/SUBJECT stability
Docker dependency conflicts	Shifted demo to Colab/local execution	Reproducible deployment

A. Software and Hardware Configuration

- OCR Engine: Tesseract OCR (v5.x)
- Document AI Model: LayoutLMv3 (Hugging Face Transformers)
- Annotation Tool: Label Studio
- UI Framework: Gradio
- Image Processing: PIL, OpenCV
- Hardware: NVIDIA T4 GPU (Colab) / CPU-only local execution

B. Training Hyperparameters

TABLE IV
LAYOUTLMV3 FINE-TUNING HYPERPARAMETERS.

Parameter	Value
Epochs	10
Batch size	2
Learning rate	5×10^{-5}
Optimizer	AdamW
Loss function	Cross-entropy
Random seed	Fixed

All preprocessing, inference, and eligibility evaluation steps are executed locally, ensuring compliance with privacy constraints.

XIV. LIMITATIONS AND THREATS TO VALIDITY

While the proposed system demonstrates strong performance under constrained conditions, several limitations must be acknowledged.

First, the annotated dataset is small and drawn from a limited number of institutions, which may restrict generalization to unseen transcript formats. Second, OCR quality remains a bottleneck for extremely low-resolution or heavily skewed images. Third, GPA scales vary internationally (e.g., 4.0, 10.0, percentage-based), and the current system assumes a consistent numeric interpretation.

Additionally, subject prerequisite detection relies on approximate keyword matching and may fail for institution-specific abbreviations. These limitations motivate the conservative use of *Review Recommended* decisions and underscore the importance of human oversight. Future work will address these threats through larger synthetic datasets, normalization strategies, and cross-institution evaluation.

XV. ETHICAL AND PRIVACY CONSIDERATIONS

Automating transcript analysis raises important ethical and privacy questions. Our prototype was developed on a small, private dataset with personally identifiable information removed wherever possible. No third-party cloud OCR APIs were used in order to avoid transmitting student data outside institutional boundaries; instead, all processing occurs locally using Tesseract and locally hosted models.

The system is designed as a decision-support tool rather than an autonomous decision-maker. Final admissions decisions remain with human officers, and the UI exposes both the extracted fields and the underlying OCR text, enabling transparent verification. Future deployments would need to follow institutional data governance policies, including secure storage, access control, and audit logging.

XVI. FUTURE WORK

Future work will focus on:

- expanding the training dataset with automatically annotated synthetic transcripts,
- incorporating OCR-free models such as improved variants of Donut [2] for direct image-to-structure prediction,
- enhancing name and subject extraction using better post-processing and attention visualization,
- adding support for multi-page transcripts and international grading scales, and
- deploying the system as a secure, FERPA-compliant cloud service with batch processing and integration into admissions portals.

XVII. CONCLUSION

This paper presented an AI-assisted system for automated transcript parsing and program eligibility assessment. By combining OCR, a fine-tuned LayoutLMv3 model, a synthetic transcript generator, and a configurable eligibility rule engine, we demonstrated that it is possible to reliably extract GPA and support admissions decisions even with limited annotated data. The accompanying Gradio UI makes the system accessible to non-technical staff and provides transparency via OCR text and explanatory summaries. We believe this work can serve as a foundation for more advanced Document AI solutions in educational institutions and other domains that rely on semi-structured scanned documents.

APPENDIX A
GPA EXTRACTION ALGORITHM

```
Input: OCR text T
Output: GPA value G, confidence C

1. Search for GPA-related keywords in T
```

```
2. For each keyword occurrence:
    Extract numeric values
    within a fixed window
3. Filter numeric candidates
   by valid GPA range
4. Select the most plausible value
5. Assign confidence based on
   keyword proximity and clarity
6. Return G and C
```

Appendix B
Eligibility Decision Engine

```
Input: GPA G, subject list S, program P
Output: Decision D, explanation E

1. Retrieve GPA threshold for P
2. If G < threshold:
       D = Not Eligible
3. Else if required subjects missing:
       D = Review Recommended
4. Else:
       D = Eligible
5. Generate explanation E
```

Appendix C
System Configuration Parameters

- GPA keyword list: GPA, CGPA, Cumulative Grade Point Average
- Confidence thresholds: keyword + numeric clarity
- Supported input formats: PDF, JPG, PNG
- Processing mode: single-page transcript (extendable)

Appendix D
Regex Patterns and Confidence Rules

We used conservative regex patterns to avoid extracting course grades as GPA. Candidate GPA values are extracted only within a window near GPA keywords.

Keyword Set

- GPA, CGPA, C.G.P.A
- Cumulative GPA, Cumulative Grade Point Average
- Overall GPA, Total GPA

Example Regex Patterns

```
( CGPA | GPA | Cumulative GPA )\s*[:\-]?\s*
([0-9]{1,2}\.[0-9]{1,2})

Cumulative Grade Point Average\s*
([0-9]{1,2}\.[0-9]{1,2})
```

Confidence Heuristics

HIGH: keyword present + valid decimal match within close window.
MEDIUM: keyword present but multiple numeric candidates.
LOW: weak keyword match or noisy OCR; triggers Review Recommended.

References

[1] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, 2007.

[2] G. Kim *et al.*, "Donut: Document understanding transformer without OCR," in *Proc. European Conf. on Computer Vision (ECCV)*, 2022.

[3] Y. Huang *et al.*, "LayoutLMv3: Pre-training for document AI with unified text and image masking," in *Proc. ACM Int. Conf. on Multimedia*, 2022.

[4] Y. Xu *et al.*, "LayoutLM: Pre-training of text and layout for document image understanding," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2020.

[5] S. X. Rao *et al.*, "SAINE: Scientific automatic information extraction," arXiv:2302.14468, 2023.

[6] U.S. Department of Education, "FERPA General Guidance for Students," Oct. 2019. [Online]. Available: https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html