University of New Haven

**MSDS Admission Model for Decision Making (ADM-DM)**

Empowering Smarter Decisions with Data-Driven Insights for MSDS Admissions"

“Shaik Adil”, “Pulleti Charan Sumanth” , “Bhonagiri, Sruthi”, “Vemireddy Navya”,”Talasani Snithika Patel”, “Dacharla Sai Kumar”, “Tumma Karthik Reddy” ,”Anabathula Deepa”, “Gembali Kalkesh”

# Introduction

The ADM-DM (Admission Model for Decision Making) project addresses the complexities of university admissions by automating transcript data extraction and GPA prediction. Traditional methods are often hindered by unstructured, diverse transcript formats, resulting in inefficiencies and inconsistencies in decision-making. This project leverages advanced Optical Character Recognition (OCR) tools like PaddleOCR to extract structured data from transcripts, builds a GPA prediction model, and integrates these components into a user-friendly graphical interface.

The system's goal is to streamline admissions, reduce manual errors, and enhance scalability, providing universities with a robust solution to manage high volumes of applications efficiently. Key features include real-time transcript processing, automated GPA predictions, and an adaptable framework capable of handling various transcript formats. By integrating innovative technology, the project sets a benchmark for transforming academic data management into an efficient, unbiased, and scalable process.

# Methodology

## 2.1. Pipeline

### 2.1.1. Pipeline Overview

### The ADM-DM project developed an end-to-end data pipeline for automated transcript analysis. The pipeline enables extraction, transformation, and loading (ETL) of data, designed to handle diverse transcript formats such as PDF, JPEG, and PNG files.

### 2.1.2. Pipeline Components

2.1.2.1. Data Collection

To ensure diversity and robustness in our dataset, academic transcripts in various formats were sourced, encompassing a range of layouts, structures, and levels of complexity. For the annotation process, we utilized **PPOCRLabel**, a specialized tool designed for labeling and processing optical character recognition (OCR) data. This tool was instrumental in handling noisy and intricate transcripts, enabling us to identify and annotate critical fields such as course names, grades, and CGPA with precision. By leveraging PPOCRLabel, we ensured a high level of accuracy in our dataset preparation, which was crucial for subsequent analysis and model development.

2.1.2.2. Preprocessing

To ensure consistency and facilitate further processing, the academic transcripts were standardized by converting them into structured **JSON files**. This transformation allowed for easier extraction and analysis of relevant data. Additionally, **super-resolution** techniques were applied to enhance the quality of noisy or low-resolution images, significantly improving the accuracy and performance of the OCR process. These enhancements ensured that the data extracted from the transcripts was clear, detailed, and suitable for subsequent analysis, contributing to more reliable outcomes in our project.

2.1.2.3.OCR Model Develpoment

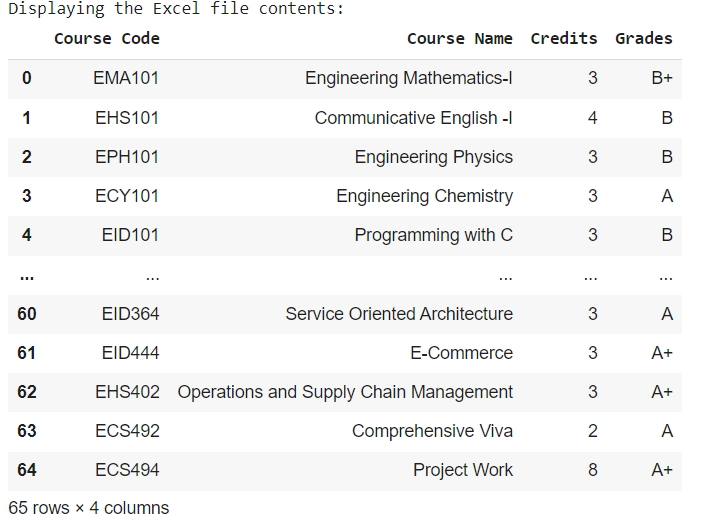
We began by downloading the pretrained model from the official **PaddleOCR GitHub repository**, which provided a robust starting point for our project. This pretrained model was already trained on large-scale datasets, allowing it to recognize a wide variety of text in images. To tailor the model to our specific needs, we fine-tuned it using our dataset, which included annotated images and labels for both text detection and recognition tasks. Using a customized **configuration file**, we adjusted hyperparameters, model architecture settings, and training strategies to optimize the performance for our specific dataset. This fine-tuning process enabled the model to better handle the complexities and nuances of the academic transcripts, improving its accuracy in identifying course names, grades, and CGPA values. By leveraging both the pretrained model and our carefully curated dataset, we were able to enhance the model's capability to perform OCR on noisy and varied transcript formats.

Model Accuracy

|  |  |  |
| --- | --- | --- |
|  | Detection | Recognition |
| Precision | 0.98 | 0.96 |
| Recall | 0.95 | 0.98 |
| F-1 Score | 0.97 | 0.95 |

2.1.2.4. Post Processing

After the text was extracted from the transcripts, we performed a series of **post-processing** steps to ensure the data was organized and structured in a meaningful way. The entire transcript, initially in **PDF format**, was first converted into a series of images, with each page of the transcript being transformed into an image to facilitate text extraction. Once the text was extracted from each image, it was systematically organized into a **tabular format**, where each piece of text was aligned according to its corresponding label (e.g., course names, grades, CGPA). This step ensured that the extracted information was not only accurate but also logically structured for further analysis. By converting the unstructured transcript data into a single, coherent tabular structure, we made it easier to process, analyze, and visualize the data for subsequent stages of the project.



2.1.2.5. GPA Prediction

The extracted data was then fed into a **Random Forest model** for GPA prediction, which utilized advanced techniques to ensure accurate and consistent results. The model employed **weighted average calculations of grades**, where each course's grade was assigned a weight based on its credit hours, reflecting the course's significance in the overall GPA. To further enhance the accuracy of the predictions, the **Random Forest algorithm** was utilized. This machine learning technique aggregates the predictions of multiple decision trees to provide a more robust and reliable output. Trained on historical data, the Random Forest model learned the underlying patterns and relationships within the grade data, allowing it to make precise and dependable GPA predictions based on the extracted academic information.

2.1.2.6. GUI Interface

The final step of the project involved integrating the **OCR model** and the **GPA prediction model** into a cohesive application that streamlines the entire process, from data extraction to GPA prediction. In this application, the **OCR model** is first utilized to extract text from academic transcripts, which may come in various formats such as scanned PDFs or images. The OCR model processes each transcript, converting it into structured text by identifying key elements like course names, grades, and CGPA. Once the data is extracted, the application performs necessary **post-processing**, organizing the text into a tabular format and ensuring that each extracted data point aligns with its corresponding label.

The tabulated data is then fed into the **GPA prediction model**, which, powered by the **Random Forest algorithm**, calculates the GPA based on weighted grade averages. The model incorporates machine learning techniques to refine the predictions, ensuring that the GPA is predicted accurately based on historical data patterns.

By integrating both models into a single application, the user can seamlessly upload a transcript, have it processed for text extraction, and receive a predicted GPA—all within one interface. This unified approach eliminates the need for manual data entry, reduces errors, and increases the overall efficiency of transcript analysis and GPA prediction. The application ensures that both OCR and machine learning algorithms work in tandem, delivering accurate results with minimal user input.

# Key Challenges and Solution

## 3.1 Challenges with data Quality

## To address noisy or inconsistent transcripts, super resolution was applied as a key preprocessing technique. Super resolution involves using algorithms to enhance the resolution of low-quality images, effectively improving the fine details that are often lost in noisy or pixelated scans. By leveraging machine learning or deep learning models, super resolution reconstructs high-resolution images from low-resolution inputs, enhancing the clarity of text and making it easier for OCR systems to recognize characters accurately. This step significantly improved the quality of the transcript images, leading to more precise text extraction.

## 3.2 Scalability

To address the challenge of processing large volumes of data, a modular architecture was implemented to support parallel processing. This approach divided the workflow into smaller, independent tasks, such as image preprocessing and OCR, which could run concurrently across multiple processors. By distributing the workload, processing times were significantly reduced, ensuring the system could scale efficiently as data volumes grew. The modular design also made it easier to update or optimize individual components without affecting the entire system.

## 3.3 Fairness

To ensure unbiased decision-making, standardized GPA calculations were implemented, ensuring consistency across all data. Additionally, fairness checks were introduced during testing to identify and correct any potential biases in the model's predictions or outcomes. These measures helped maintain fairness and equity in the decision-making process, ensuring that all students were evaluated based on the same criteria

# Deliverables

### 4.1 Phase-1

Annotated transcript data was collected and prepared for further processing. This data served as the foundation for training the initial OCR model, which was developed to recognize and extract relevant information from the transcripts. The preliminary model aimed to test the feasibility of the OCR system, allowing for early assessments of its accuracy and performance on real-world data.

### 4.2 Phase-2

The OCR model was enhanced with fine-tuning to improve its accuracy, refining its ability to correctly interpret transcript data. Additionally, super-resolution was integrated into the system to enhance the quality of low-resolution images, allowing for better handling of unclear or noisy transcript scans. These improvements led to more accurate text

### 4.3 Phase-3

A graphical user interface (GUI) was developed to allow users to easily upload transcripts and receive GPA predictions. The GUI streamlined the process, providing a user-friendly platform for interacting with the system. Users could simply upload their annotated transcript data, and the model would process it, providing a GPA prediction based on the extracted information, making the system more accessible and efficient.

# 5 Deployment

For the deployment of the integrated OCR and GPA prediction models, we utilized a pre-existing **local GUI application** to streamline the user experience. This GUI served as the interface through which users could interact with the system, providing a simple and intuitive way to upload academic transcripts, initiate the OCR process, and receive GPA predictions.

The deployment process involved integrating the **OCR model** and the **Random Forest GPA prediction model** within the local GUI framework. The application was configured to allow users to upload a transcript, typically in PDF or image format, through a file upload interface. Once the user submits the file, the application triggers the **OCR model**, which processes the transcript, extracts the relevant text, and arranges it into a structured tabular format.

After the OCR processing, the tabulated data is automatically fed into the **GPA prediction model**, which calculates the GPA using weighted average calculations and machine learning algorithms. The results are then displayed in the GUI, allowing the user to view the predicted GPA along with any other relevant data extracted from the transcript.

The **local deployment** ensured that all the processing, from OCR to GPA prediction, occurred directly on the user's machine without requiring an internet connection. This approach also offered better performance, as the models were fine-tuned to run locally, reducing potential latency and providing fast results. Additionally, the application was designed to handle multiple types of academic transcripts, ensuring flexibility and robustness in its functionality.

By using a local GUI application, we ensured that the deployment was user-friendly, accessible, and efficient, providing a seamless experience for users needing to process transcripts and predict GPAs without requiring technical expertise.

# 6 Lessons learned

One of the key lessons we learned is that robust preprocessing is crucial when dealing with noisy datasets. By applying techniques like image enhancement and organizing the data properly, we were able to clean the transcripts and ensure that the text extracted by the OCR was accurate and usable. Another important takeaway was the value of iterative testing. Since academic transcripts can vary widely in format, we needed to test the models repeatedly on different types of transcripts. This allowed us to adjust and improve the models, making them more adaptable and accurate across various formats

# 7 Future Work

## 7.1 Multilingual OCR

To increase the system’s versatility, we plan to extend the OCR functionality to support non-English transcripts. This would allow the system to process transcripts in multiple languages, making it applicable to a broader range of students and institutions.

## 7.2 Scholarship Prediction:

Another potential improvement is integrating models that can analyze student data and predict scholarship eligibility. By considering academic performance, extracurricular activities, and other relevant factors, the system could help identify top candidates for various scholarships.

## 7.3 API Integration

## To improve interoperability, we can integrate the system with existing admission platforms through seamless **API integration**. This would allow the data extracted and processed by the OCR and prediction models to be directly used by admission systems, streamlining the entire process.

## 7.4 Predictive Modelling for course planning

Add a **predictive modeling** feature to suggest personalized course plans for students based on their academic history and performance. The model could recommend courses that align with students' strengths, interests, and career goals, helping them make more informed decisions about their education path

# 8 Conclusion

The ADM-DM project showcases how automation can streamline and enhance university admissions processes. With its robust OCR model, predictive capabilities, and user-friendly interface, this system sets a benchmark in educational technology. Continuous development, including multilingual support and advanced analytics, will further solidify its impact on higher education.