Project Proposals by IIT Delhi and University of Alabama in 2025
GSOC 2025 • HumanAI

Proposal for GSOC 2025 Application

Aditya Raj

National Institute of Technology, Patna
hexronus@gmail.com, adityar.ug22.ec@nitp.ac.in

https://www.linkedin.com/in/hexronus/, https://hexronus.vercel.app/

+91 8797073498

TimeZone: India Standard Time • Patna, Bihar (GMT+5:30)

1 Background

My name is Aditya, I am 22 years old, currently in 3rd year of my B.Tech majoring in Electronics and Communication from the National Institute of Technology, Patna. The courses I have covered/covering related to Deep Learning are Artificial Intelligence and Machine Learning, Introduction to Deep Learning, Introduction to Large Language Models, and Applied Linear Algebra.

1.1 Competitions

I won the Annual ML Challenge of IIT-ISM AI of God 3.0, where the task was to recognize old Spanish manuscripts, where I finetuned Tr-OCR on the train-set, and used T5 on the train text after finetuning it using a custom function which was used for grammar correction achieving a WER of 0.116.

Github: Algorithm

I got 184th rank (F1 0.467) in Amazon ML challenge where the task was to extract quantity elements from product labels, I finetuned moondream VLM on the training set with prompting and a regex block for structuring the output.

1.2 Internships

I have done academic internship at IIT-Guwahati, where I learnt to developed a GAN-based shape completion system in Python, used UNet-based Generator to reconstruct overlapping geometric shapes, generated a diverse synthetic dataset to train the model effectively.

I am currently working with a startup 21Spheres where I developed a multimodal AI search engine using Ollama and CLIP, wrapped around a custom algorithm for personalized recommendations and search, it achieved a near-perfect human-aligned results which enhanced product discovery.

1.3 Awards

I qualified Regional Mathematical Olympiad(RMO) in 2019, where I ranked among the top 0.4% of students in India, Indian Qualifier in Astronomy in 2022, National Talent Search Exam (I) in 2020, METER where I achieved State Rank 7th among 50,000+ students.

This is my portfolio, and my linkedin profile, hexronus. Please visit here to get more information about me.

Thank you for considering my application.

2 Project proposal - End-to-end handwritten text recognition for early modern spanish documents

2.1 Task

Our goal is to build a robust model for optical text recognition tailored to digitized early modern Spanish manuscripts. We're leaning on transformer-based or self-supervised architectures, TrOCR[3] or similar, to tackle full-document recognition. This means not just extracting text, but also considering the document's structure: layout, reading order, the works. These manuscripts are tricky—handwritten, faded ink, so we need a system that's adaptable, and precise.

2.2 Working and Results

This was performed on a test dataset as a Task, This reflects the workflow of the original proposal work.

Dataset

We will be using a dataset specially curated for Old Spanish Printed/Handwritten text, the Transformer-OCR model will be trained on that particular dataset and will be further finetuned using Transfer Learning, Domain Adversarial Training or Active Learning for working well with Handwritten or to accommodate typographical styles, vocabulary, and regional idiosyncrasies.

We begin by splitting the train dataset which consisted of 12,008 old Spanish handwritten manuscript as train and 3,002 printed Spanish manuscript as test (IIT ISM AI-of-GOD 3.0 Dataset). We write a custom dataset class which reads images, converts Spanish Image Dataset to RGB, and tokenizes text transcriptions. Padding tokens are replaced with `-100` so the loss function ignores them. Next, we instantiate a TrOCRProcessor and build both training and evaluation datasets.

Model

We are choosing TrOCR because it gives top-notch performance, it uses transformer-based technology adapted for computer vision, making it highly effective at extracting text. It excels at recognizing text even when images are noisy or the text is distorted—situations where other models might struggle(Tesseract[5], EasyOCR[6] as these are trained to work with scanned neat documents), and as it is open sourced TrOCR allows us to customize or fine-tune it to fit your specific needs.

We then load a VisionEncoderDecoderModel (Microsoft's TrOCR base), set decoder and pad tokens, and specify beam search parameters. Using Seq2SeqTrainingArguments, we define batch sizes, evaluation intervals, logging steps, etc. A custom compute_metrics function calculates average Word Error Rate (`WER`) via `jiwer`. Finally, we wrap everything in a `Seq2SeqTrainer`, supply the model, processor, datasets, and data collator, and run `trainer.train()`.

Results

We achieved a WER of 0.901519 on the train set, which established our baseline for handwritten old Spanish OCR detection. For printed OCR detection, we used the Spanish T-OCR Large Model(due to it's training set being printed Spanish sentences). Although the image detection model occasionally required corrections, these were addressed by applying a T5[4] model on the text after fine-tuning it using a custom algorithm which was used for grammar correction.

Parameter	Value
CPU count	6
Logical CPU count	12
GPU count	1
GPU type	NVIDIA L4

Table 1: Overview of the hardware setup.

Parameter	Value
eval/loss	0.6435500979423523
train_loss	1.5594333657409795
train_runtime	4,466.3357
train/epoch	2
train/learning_rate	0.00000013315579227696
train/loss	0.528

Table 2: Training and Evaluation Metrics

Algorithm for Grammar Correction using T5

We are choosing T5 as compared to other alternatives because, T5 is pre-trained to reconstruct original text from corrupted versions (denoising), a task closely aligned with grammar correction—identifying and fixing errors. T5's text-to-text framework allows it to be fine-tuned easily on datasets of grammatically incorrect and correct sentence pairs, directly mapping errors to fixes. T5's Transformer architecture captures relationships across entire sentences, essential for understanding grammar in context.

This dataset was created by collecting a large corpus of Spanish sentences from the train transcriptions. In the first step, "Normalization," the algorithm systematically replaced and removed characters prone to confusion or misusage. For example, letters like v were changed to u, z to c, and y to i. Accents were removed using a unidecode-like method, and some letters were interchanged (e.g., $b \leftrightarrow v$) to introduce realistic variation.

The second step involved further modifying the corpus by removing, interchanging, or randomly splitting characters in words. The of the occurrences of key letters (a, e, o, s, i) were removed or swapped, and occurrences involved manipulating the letter u. The algorithm also introduced letter duplications and swaps (e.g., $g \leftrightarrow j$, $s \leftrightarrow z$), occasionally reintroduced uppercase letters in about 50% of words, and randomly added or removed accents in around 20% of words.

Ultimately, the algorithm was designed to produce a diverse array of artificial errors that reflect realistic mistakes seen during manual checking in Spanish text. These include missing letters, swapped letters, accent misplacements, and duplicated characters. By simulating these errors, the resulting dataset serves as a robust training and testing resource for grammar-correction models, enabling them to learn and correct common Spanish-language mistakes more effectively.

Line Segmentation

Line Segmentation comprises of both the extraction of lines from pages and marking text-based[2] regions in the document; this can be achieved by using a localization CNN or Transformers[1] based models for marking the start and end point in each horizontal line and using it to calculate the text-region to perform OCR, below is a simpler mathematical model which uses A* algorithm to perform line segmentation.

We implemented an A* algorithm to extract text lines from binary images. Our approach initializes with the binary image, start/end coordinates, and an optional mask, then uses a Manhattan heuristic and eight-directional moves with tailored weights to navigate through regions, effectively balancing path length with pixel value penalties.

We preprocessed the image by converting it to grayscale, applying adaptive thresholding, and performing morphological cleanup. A projection profile identified text line regions that were merged on the basis of gap criteria. Using the A* algorithm, we extracted optimal paths within these regions, allowing us to crop precise text line segments.

HDR

Various methods such as Transfer Learning, Active Learning can be used to fine-tune the model on handwritten dataset with a LLM based grammar correction for better accuracy. The metrices used is Word Error Rate(WER),

Conclusion

We presented the GSOC 2025 proposal which uses the large-spanish-TrOCR or TrOCR base or a Transformer based OCR model trained on the Spanish dataset after fine-tuning it on the specific handwritten dataset used with a T5 based grammar correction algorithn to make it work in real-life cases to detect handwritten documents.

The model also comes with a line segmentation algorithm which is used to detect text-based regions and only apply the line extraction and OCR in those places for better results.

TrOCR uses Cross Entroply Loss as it works as a classification problem and the metrics observed is Word Error Rate(WER).

3 Roadmap

Date Range (GSOC 2025)	Tasks
8th May, to 1st June,	• Read research paper related to OCR and understand text-recognition in depth • Finalize workflow and OCR Architecture Pipeline
2nd June, to 20th June,	• Small Object Detecton - RCNN, YOLO based architecture to classify the start and end of text in line wise input collected from A* Line Segmentation (precision, recall, and F1-score for detection accuracy) • Tuning the Model and Testing it against various inputs
21st June, to 14th July,	• Validate the perfect text region detection model and checking and correcting the flaws, if there are no flaws proceeding with using the Tr-OCR model for training.
15th July, to 25th July,	• Collecting/Curating a grammar based dataset for Training T5 as grammar correction model.
26th July, to 6th August,	• Integrating the Text area detection model with TrOCR and T5, and testing to make it work in real case.
6th August, to 10th August,	• Giving Everything a final check and rigorously testing the working of the model.
15th August, to 25th August,	• Building Everything as a pipeline and ready to use in real world case, built as a python package, writing documentation and report.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 213–229, 2020. Facebook AI. `https://arxiv.org/abs/2005.12872`.

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character Region Awareness for Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9365–9374, 2019. Clova AI Research, NAVER Corp. `https://arxiv.org/abs/1904.01941`.

[3] Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models, 2021, arXiv:2109.10282 [cs.CL].

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *Journal of Machine Learning Research*, 21(140):1–67, 2020, `http://jmlr.org/papers/v21/20-074.html`.

[5] Samuel Hoffstaetter, Juarez Bochi, Matthias Lee, Lars Kistner, Ryan Mitchell, Emilio Cecchini, and others. pytesseract: Python-tesseract, an Optical Character Recognition (OCR) Tool for Python, GitHub repository, version 0.2.3, 2025, `https://github.com/madmaze/pytesseract`.

[6] Rakpong Kittinaradorn and contributors. EasyOCR: Ready-to-use OCR with 80+ Supported Languages, GitHub repository, version 1.7.2, 2024, `https://github.com/JaidedAI/EasyOCR`.