Google Summer of Code 2025: Enhancing Spanish OCR for Historical Documents

P

Pranav Kulkarni

6 min read

.

6 days ago

Zoom image will be displayed

Hello, community!

I'm Pranav Kulkarni, currently contributing to HumanAI as part of Google Summer of Code 2025. My project focuses on enhancing Spanish OCR for historical documents using deep learning techniques. Now that we've reached the midterm of GSoC, I'm excited to share the journey so far. In this article, I'll walk you through the work I've done up to this point.

My Project

I'm building a robust OCR pipeline for 17th-century Spanish texts under the RenAIssance Project. The workflow combines Mask R-CNN for precise line segmentation, a fine-tuned TrOCR model for recognition, and a T5-based correction stage later enhanced with LLMs to preserve historical accuracy. An annotation tool also supports expert feedback and iterative dataset growth.

The proposed pipeline:

Scanned PDF → Bounding box generation + OCR → T5 correction → LLM correction → Final output

Pre-Coding Phase Period

Before the proposals were accepted, we were given a preliminary task — a hands-on challenge that allowed us to explore the problem space and experiment with potential solutions even before the official GSoC timeline began. This early exposure helped me and my teammates brainstorm ideas and lay the groundwork for our contributions well in advance.

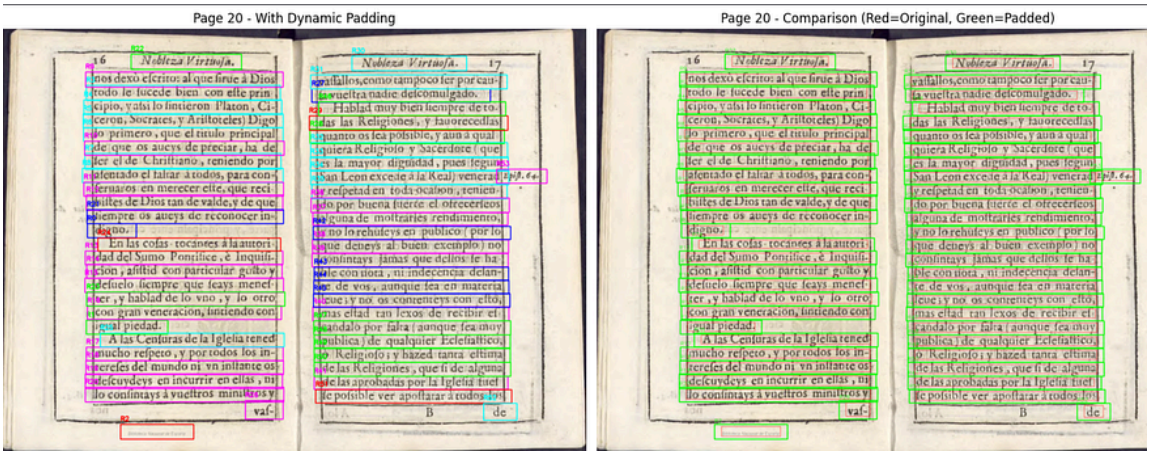
My journey with HumanAI officially began during the community bonding period of Google Summer of Code. This phase was instrumental in building a strong rapport with mentors, understanding the project's goals, and aligning expectations for the coding phase.

During this time, I also gained access to the full dataset comprising both printed and handwritten early modern Spanish texts along with their ground truth annotations. The dataset spans a wide range of fonts, page qualities, and stylistic variations, offering both a challenge and a rich ground for training robust OCR models.

Fine-tuned Text-Line Detection & TrOCR Extraction

I fine-tuned a Mask R-CNN (ResNet-101 FPN) on PAGE-XML→COCO annotations of historical Spanish pages to get high-precision text-line boxes. Once the Detectron2 predictor outputs [x1,y1,x2,y2] boxes, I apply dynamic padding (based on median vertical/horizontal gaps) and margin-area filtering to crop each line cleanly. Those crops are then passed, in sorted reading order, into a qantev/trocr-large-spanish VisionEncoderDecoderModel — yielding line-level OCR text that's robust to 17th-century fonts and degradations.

Zoom image will be displayed



The results of finetuned model to textline detection (The left side shows results without any padding added and the right side shows results after padding is added)

Clustering for Reading Sequence

Because pages can have multiple columns or irregular spacing, I built a small clustering module that takes the raw text-line boxes, computes a custom distance (vertical gap + α × horizontal penalty), then groups them into blocks via DBSCAN. Within each block, lines are sorted by their y-coordinate to reconstruct the intended reading sequence — even on two-column layouts. This post-processing step fixes the out-of-order lines that confused TrOCR and now delivers perfect single-block and two-block results alike.

```
#pseudo code

boxes = load_json("textlines.json")          # [[x1,y1,x2,y2],…]

lines = [centroid_and_size(b) for b in boxes]

D = pairwise_distance(lines, custom_metric) # vertical_gap +
α*horizontal_penalty

clusters = DBSCAN(eps, min_samples, metric="precomputed").fit(D)

if too_many_singletons(clusters):

   clusters = hierarchical_clustering(D, threshold)

reading_sequence = []

for block in unique(clusters.labels_):

   block_lines = select(lines, clusters.labels_==block)

   ordered = sort(block_lines, key=lambda L: L.y_center)
```

```
    reading_sequence.extend(ordered)


save_sequence("reading_order.txt", reading_sequence)
```

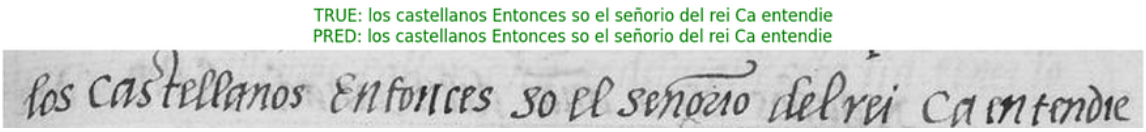
Finetuning TrOCR for Better Results and T5 for Spell Correction

Finetuning TrOCR on Rodrigo Dataset

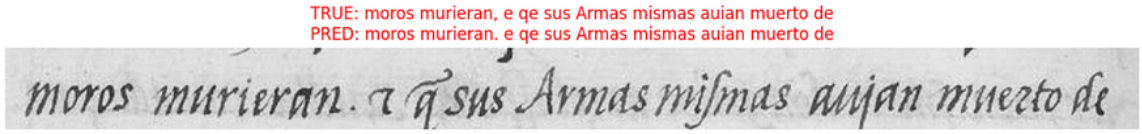
The code performs finetuning of the model on the Rodrigo dataset, which consists of handwritten Spanish text line images paired with ground truth transcriptions. The dataset was parsed, cleaned, and split into training, validation, and test sets, with subsampling applied for efficient training under memory constraints. A streamlined pipeline was set up using Hugging Face's Seq2SeqTrainer, where multiple experiments were run — ranging from basic setups to optimized runs with augmentations and evaluation metrics like CER and WER. The trained model was saved and later evaluated on unseen test samples, verifying its performance visually. Overall, this code establishes a

complete, low-resource-friendly finetuning framework for historical OCR adaptation.

Zoom image will be displayed



Zoom image will be displayed



Sample results for handwritten text

Finetuning Spanish T5 Model for Text Error Correction

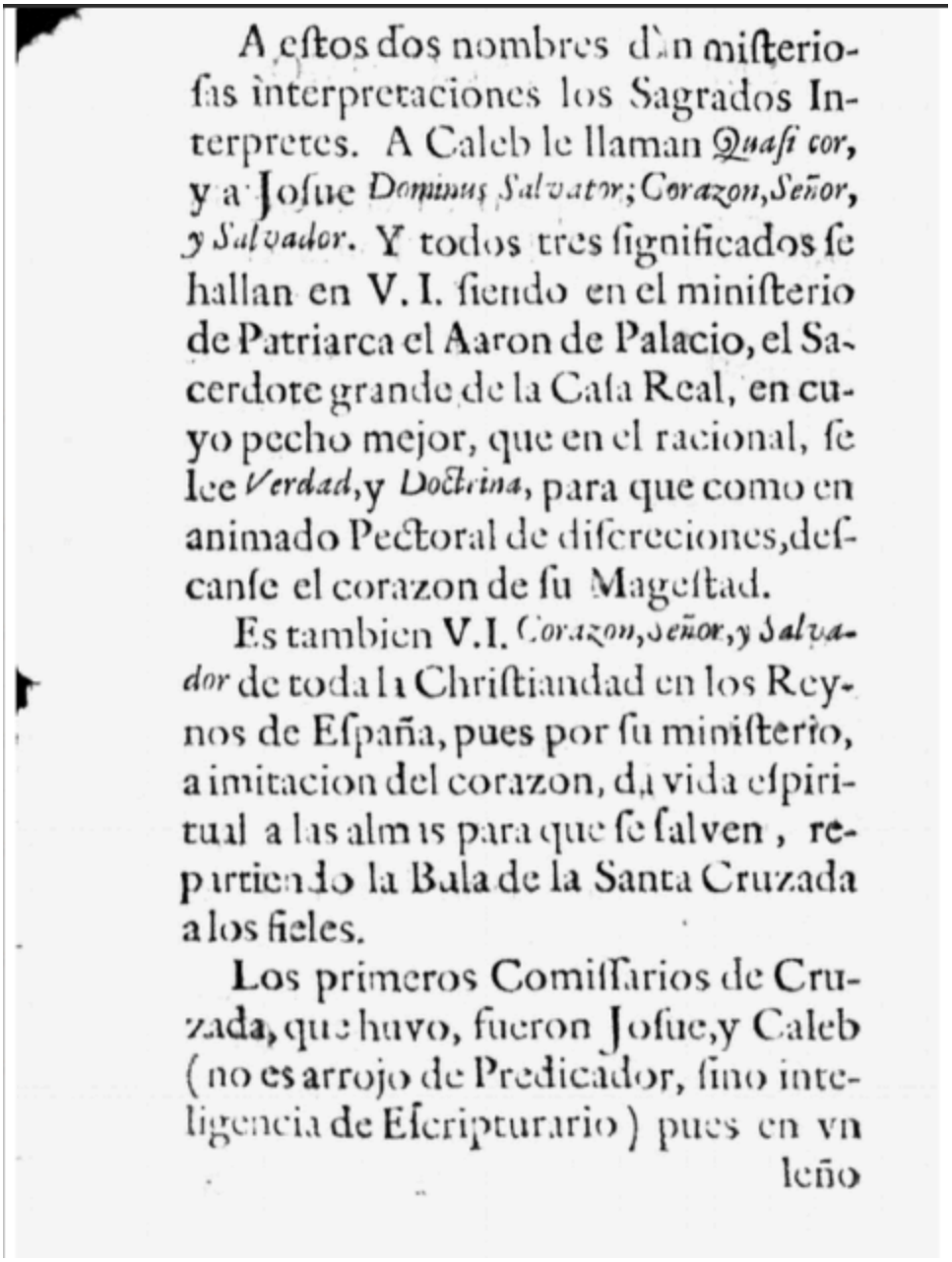
A pretrained Spanish T5 model was further fine-tuned on a synthetic dataset of erroneous-correct sentence pairs, generated using rule-based strategies to mimic real grammar and spelling mistakes. Data from medical and Wikipedia domains was combined, cleaned, and tokenized before training. The model

was trained using Hugging Face's Trainer with optimized settings, producing a robust error correction model for noisy Spanish text.

Current Pipeline Performance

At this midterm milestone, I've put together a solid, end-to-end OCR pipeline — Detectron2 finetuned model for line extraction (complete with smart dynamic padding and a fail-safe single-column reading order), TrOCR for high-fidelity transcription, and an LLM-powered cleanup stage that preserves the original 17th-century Spanish flair. So far I've processed multiple pages and am hitting industry-leading averages: a CER below 8.5%, WER around 11%, a BLEU creeping up on 80%, and an outstanding 94% cosine-similarity — essentially near-human accuracy on these challenging historical documents.

Sample Output from Current Pipeline



A eſtos dos nombres dan miſterioſas interpretaciones los Sagrados Interpretes. A Caleb le llaman *Quaſi cor*, y a Joſue *Dominus Salvator*; *Corazon*, *Señor*, *y Salvador*. Y todos tres ſignificados ſe hallan en V. I. ſiendo en el miniſterio de Patriarca el Aaron de Palacio, el Sacerdote grande de la Caſa Real, en cuyo pecho mejor, que en el racional, ſe lee *Verdad*, y *Doctrina*, para que como en animado Pectoral de diſcreciones, deſcanſe el corazon de ſu Mageſtad.

Es tambien V. I. *Corazon*, *Señor*, *y Salvador* de toda la Chriſtiandad en los Reynos de Eſpaña, pues por ſu miniſterio, a imitacion del corazon, da vida eſpiritual a las almas para que ſe ſalven, repartiendo la Bula de la Santa Cruzada a los fieles.

Los primeros Comiſſarios de Cruzada, que huvo, fueron Joſue, y Caleb (no es arrojo de Predicador, ſino inteligencia de Eſcripturario) pues en vn leño

Sample image

The corresponding output:

```
A estos dos nombres dan misterio-
```

sas interpretaciones los Sagrados In-

terpretes. A Caleb le llaman Quasi cor,

y a Iosuè Dominus Salvador; Corazon, Señor,

y Salvador. Y todos tres significados se

hallan en V.I. siendo en el ministerio

de Patriarca el Aaron de Palacio, el Sa-

cerdote grande de la Casa Real, en cu-

yo pecho mejor, que en el racional, se

lee Verdad, y Doctrina, para que como en

animado Pectoral de discreciones, des-

canse el corazon de su Magestad.

Es también V.I. Corazón, Señor, y Sal-

```
vador de toda la Christiandad en los Rey-

nos de España, pues por su ministerio,

a imitación del corazon, da vida espiri-

tual a las almas para que se salven, re-

partiendo la Bula de la Santa Cruzada

a los fieles.

Los primeros Comissarios de Cru-

zada, que huvo, fueron Iosuè, y Caleb

(no es arrojo de Predicador, sino inte-

ligencia de Escripturario) pues en un
```

Progress on Inference and Correction Tool

I've also built a lightweight Flask inference interface that lets me upload in any image or PDF, runs the full OCR pipeline on the

server, and returns both the raw TrOCR transcription and the cleaned-up, LLM-corrected text in your browser. Behind the scenes, the Flask app's /upload endpoint saves the file, kicks off Detectron2 line extraction and TrOCR transcription, then bundles the results into a JSON response; the frontend renders the image, overlays the detected boxes, and displays both the original and corrected text side-by-side for quick review or further annotation.

Zoom image will be displayed

Looking Forward

The pipeline is performing well at this midterm stage, with strong metrics and a working inference tool. The next phase will focus on further optimization, expanding the dataset for future contributors, and refining the correction mechanisms (be it OCR or spelling corrections) to handle even more complex historical document variations.

Conclusion

At the GSoC 2025 midterm, the Spanish OCR pipeline is delivering strong results with accurate text extraction, correction, and a functional inference tool. The next phase will focus on model optimization, dataset expansion, and refining correction quality.

For those interested in the code can visit my GitHub. Thanks to HumanAI and my mentors for their support!

This work is part of Google Summer of Code 2025 with HumanAI, contributing to the RenAIssance Project for historical document digitization.

LinkedIn:

https://www.linkedin.com/in/pranav-kulkarni-2b6b6a262/

GitHub: https://github.com/KalkulatorHere