Google Summer of Code 2025 Proposal

Automating Text Recognition and Transliteration of Historical Documents with Weighted Convolutional-Recurrent Architectures

Name and Contact Information

Full Name: Mohammad Zainuddin
Preferred Name: Zain
Email: mzainuddin51@gmail.com
GitHub: https://github.com/mzainuddin51
LinkedIn: https://www.linkedin.com/in/mohammad-zainuddin-643294104/

Synopsis

This project aims to automate text recognition and transliteration of 17th-century Spanish printed sources using a hybrid end-to-end model based on weighted convolutional-recurrent architectures (CNN-RNN). Current OCR tools, such as Adobe Acrobat and Tesseract, struggle to extract text from early printed works due to their unique characteristics, including irregular fonts, rare letterforms, diacritics, printing imperfections (e.g., ink bleed), and paper degradation. To address this, I propose developing a CNN-RNN model to recognize text in historical Spanish documents, with enhancements to improve performance on non-standard text: (1) weighted learning techniques to better handle rare letterforms, diacritics, and symbols specific to Renaissance Spanish sources, and (2) constrained beam search decoding with a Renaissance Spanish lexicon to reduce hallucinated outputs and enhance word-level accuracy. The model will achieve at least 80% text extraction accuracy on historical texts.

Building on my work in Test I (Layout Organization Recognition, detailed in `LayoutRecognition.pdf`), I have implemented a layout recognition pipeline using YOLOv5 with transfer learning to detect main text regions in 40 historical Spanish book page images (e.g., *Buendia-Instruccion*, *Mendo-Principe-perfecto*, *Excaray-Vozes*, *Constituciones-sinodales-Calahorra-1602*). I pre-trained YOLOv5 on a newspaper dataset (1200/100 train/val images), achieving an mAP of 0.65, precision of ~0.8, and recall of ~0.67, then fine-tuned it on the Spanish dataset, achieving an mAP of 0.8, precision of 0.81, and recall of 0.8 at an image size of 416 and 50 epochs. Despite computational limitations (e.g., MPS memory leakage on M2 Pro), this layout recognition step effectively preprocesses images by extracting main text regions, which will be fed into the CNN-RNN model for text recognition.

This project will develop the CNN-RNN model, expand the dataset to 200 images, implement weighted learning and constrained beam search decoding, and evaluate the model on real historical texts. The final deliverables will include a text recognition pipeline, a dataset of 200 annotated images, a trained CNN-RNN

model, comprehensive documentation, and a Jupyter Notebook summarizing the results.

Benefits to Community

This project offers significant benefits to the open source community, particularly for organizations focused on historical document analysis, digital humanities, and machine learning:

- **Advanced Text Recognition for Historical Texts**: The CNN-RNN model, enhanced with weighted learning and constrained beam search decoding, will enable accurate text recognition in 17th-century Spanish printed sources, addressing a critical gap in current OCR tools. Achieving at least 80% text extraction accuracy will make historical texts more accessible for research and digitization.
- **Cultural Preservation**: By automating text recognition and transliteration, the project supports the preservation and study of Renaissance-era Spanish literature, making works like *Mendo-Principe-perfecto* and *Constituciones-sinodales-Calahorra-1602* more accessible to scholars, educators, and the public.
- **Robust Machine Learning Models**: The weighted learning techniques will improve the model's ability to handle rare letterforms and symbols, benefiting other historical document analysis tasks (e.g., manuscript recognition, text segmentation).
- **Open Source Contribution**: The project will deliver a well-documented text recognition pipeline, a dataset of 200 annotated historical images, and a trained CNN-RNN model, all of which will be open-sourced under the target organization's repository. This will foster collaboration and further development in the open source community.

Google and the target organization will be proud to sponsor this work, as it demonstrates the power of hybrid machine learning models in preserving cultural heritage while advancing open source tools for historical research.

Deliverables

Required Deliverables

- A text recognition pipeline for 17th-century Spanish printed sources, including:
  - A layout recognition module (using YOLOv5) to detect main text regions.
  - A hybrid CNN-RNN model for end-to-end text recognition.
  - Weighted learning techniques to improve recognition of rare letterforms, diacritics, and symbols.
  - Constrained beam search decoding with a Renaissance Spanish lexicon to enhance word-level accuracy.

- A dataset of 200 annotated historical Spanish book page images with bounding boxes and transcriptions, stored in `./dataset`.
- A trained CNN-RNN model achieving at least 80% text extraction accuracy on historical texts.
- Evaluation metrics (e.g., Character Error Rate, Word Error Rate, mAP, precision, recall) for both layout recognition and text recognition.
- Comprehensive documentation of the pipeline, including setup instructions, usage guide, and evaluation results.
- A Jupyter Notebook summarizing the project, methodology, and results.

Optional Deliverables

- A white paper detailing the methodology, challenges (e.g., handling rare letterforms, computational limitations), and findings of the text recognition process.
- A blog post showcasing the model's performance on historical texts, with visual examples of recognized text regions and transcriptions.
- An expanded dataset of up to 300 annotated images if initial evaluation metrics indicate the need for more data diversity.

Project Methodology

The project builds on my existing work in Test I (Layout Organization Recognition, detailed in `LayoutRecognition.pdf`) and will implement the following steps:

1. **Dataset Preparation**:
    - Start with the dataset of 40 historical Spanish book page images (e.g., *Buendia-Instruccion_pdf_page_1_png.rf.1df35c148a953090a046fde77a5d4667.jpg*, *Mendo-Principe-perfecto_pdf_page_8_png.rf.ff199ef18d2cd1c3fbad-dbb4a8b642a6.jpg*), already collected in `./test1/yolov5/spanishdocs/train/images`.
    - Use the YOLOv5 model from Test I (mAP: 0.8, precision: 0.81, recall: 0.8) to detect main text regions in these images, cropping the regions for text recognition.
    - Expand the dataset by collecting 160 additional historical images (e.g., from public domain sources like the Internet Archive) to reach 200 images, increasing diversity in text styles, fonts, and degradation patterns.
    - Preprocess the images to improve detection accuracy, applying binarization, grayscale conversion, and denoising as recommended in `LayoutRecognition.pdf`.
    - Annotate the new images with bounding boxes for main text regions (using the YOLOv5 model) and transcriptions (manually or from existing transcriptions like `Buendia transcription.docx`).
    - Apply data augmentation techniques (e.g., rotation, color jitter, brightness adjustments) to increase the effective diversity of the dataset,

resizing images to 416x416 pixels while preserving aspect ratio (consistent with Test I training).
- Store the annotated dataset in `./dataset` with metadata (e.g., bounding box coordinates, transcriptions) saved in `metadata.csv`.

Detected Text Regions
Figure 1: Sample historical Spanish book page with main text regions detected by the YOLOv5 model (red bounding boxes).

2. **CNN-RNN Model Development**:
   - Implement a hybrid CNN-RNN model for end-to-end text recognition:
     - **CNN Backbone**: Use a pre-trained CNN (e.g., ResNet-18) to extract features from the cropped text regions.
     - **RNN Component**: Use a bidirectional LSTM to model the sequential nature of text, capturing dependencies between characters.
     - **CTC Loss**: Apply Connectionist Temporal Classification (CTC) loss for training the model to predict character sequences without explicit alignment.
   - Preprocess the cropped text regions: resize to a fixed height (e.g., 32 pixels) while maintaining aspect ratio, convert to grayscale, and normalize pixel values.
   - Train the model on the annotated dataset using PyTorch, with initial hyperparameters: learning rate 1e-4, batch size 16, and Adam optimizer, to be adjusted during iteration.
3. **Weighted Learning for Rare Letterforms**:
   - Identify rare letterforms, diacritics, and symbols specific to Renaissance Spanish sources (e.g., long s [ſ], ligatures like æ, diacritics like ñ) by analyzing the dataset and consulting historical Spanish lexicons.
   - Implement weighted learning by assigning higher weights to rare characters in the CTC loss function, ensuring the model prioritizes learning these underrepresented forms.
   - Use class-balanced loss to handle imbalanced character distributions, with weights inversely proportional to character frequency in the dataset.
4. **Constrained Beam Search Decoding**:
   - Build a Renaissance Spanish lexicon by compiling a vocabulary of common words, spellings, and abbreviations from 17th-century Spanish texts (e.g., using transcriptions from `Buendia transcription.docx` and public domain sources).
   - Implement constrained beam search decoding during inference to guide the CNN-RNN model's predictions:
     - Use the lexicon to constrain the output to valid Renaissance Spanish words, reducing hallucinated outputs (e.g., nonsensical

character sequences).
  - Set beam width to 5 to balance accuracy and computational efficiency.
- Integrate the decoding step into the text recognition pipeline, ensuring the model outputs accurate word-level transcriptions.

5. **Model Evaluation**:

- Evaluate the layout recognition module using mAP, precision, and recall, aiming to maintain or improve the mAP of 0.8 from Test I with the expanded dataset and preprocessing improvements.
- Evaluate the text recognition model using Character Error Rate (CER) and Word Error Rate (WER) on a test set of real historical texts (e.g., *Excaray-Vozes*), aiming for at least 80% text extraction accuracy (1 - CER).
- Compare the model's performance with and without weighted learning and constrained beam search to quantify their impact on accuracy for rare letterforms and overall transcription quality.

Text Recognition Output
Figure 2: Sample text recognition output from the CNN-RNN model, showing the detected text region (left) and the transcribed text (right).

Work Breakdown Structure and Timeline (175 Hours)

The GSoC coding period is 12 weeks (June 2, 2025, to August 25, 2025), and the project requires 175 hours of work. I plan to commit approximately 14.6 hours per week, ensuring all deliverables are completed by the final submission deadline on September 1, 2025. Below is the detailed work plan:

- **Week 1 (June 2 – June 8, 2025): Project Setup and Research (14.6 hours)**
  - Set up the development environment (Python, PyTorch, YOLOv5, OpenCV, python-docx) (2 hours).
  - Research related work in text recognition for historical documents (e.g., CNN-RNN models, weighted learning, beam search decoding) (4 hours).
  - Review the Test I codebase (`LayoutRecognition.pdf`) and results (mAP: 0.8, precision: 0.81, recall: 0.8) to finalize the integration plan (4 hours).
  - Discuss the project plan with the mentor and confirm deliverables (4.6 hours).
  - **Milestone**: Project setup completed, integration plan finalized.

- **Week 2 (June 9 – June 15, 2025): Dataset Expansion and Annotation (14.6 hours)**
  - Collect 160 additional historical Spanish book page images from public domain sources (e.g., Internet Archive) to expand the dataset to 200 images (5 hours).
  - Preprocess the images (binarization, grayscale conversion, denoising) and use the YOLOv5 model from Test I to detect main text regions, cropping the regions for text recognition (4 hours).
  - Annotate the new images with transcriptions (manually or using `Buendia transcription.docx`) and save metadata in `metadata.csv` (5.6 hours).
  - **Milestone**: Expanded dataset of 200 annotated images ready for training.
- **Week 3 (June 16 – June 22, 2025): CNN-RNN Model Setup (14.6 hours)**
  - Implement the CNN-RNN model: set up a ResNet-18 backbone for feature extraction and a bidirectional LSTM for sequence modeling (6 hours).
  - Preprocess the cropped text regions: resize to 32 pixels height, convert to grayscale, and normalize (4 hours).
  - Begin training the model on the annotated dataset with CTC loss (4.6 hours, continued in Week 4).
  - **Milestone**: CNN-RNN model implemented, training initiated.
- **Week 4 (June 23 – June 29, 2025): CNN-RNN Model Training and Evaluation (14.6 hours)**
  - Complete training the CNN-RNN model (6 hours).
  - Evaluate the model on a test set using CER and WER, analyzing initial performance (4 hours).
  - Document the model architecture and initial results (4.6 hours).
  - **Milestone**: Initial CNN-RNN model trained and evaluated.
- **Week 5 (June 30 – July 6, 2025): Weighted Learning Implementation (14.6 hours)**
  - Identify rare letterforms, diacritics, and symbols in the dataset (e.g., long s [f], ligatures, diacritics) (4 hours).
  - Implement weighted learning by assigning higher weights to rare characters in the CTC loss function (6 hours).

  - Begin retraining the model with weighted learning (4.6 hours, continued in Week 6).

  - **Note**: I will take a 3-day break (June 24–26, 2025) for a family event, which I will compensate for by working 16 hours in Week 6.

  - **Milestone**: Weighted learning implemented, retraining initiated.
- **Week 6 (July 7 – July 13, 2025): Weighted Learning Completion and Midterm Evaluation (16 hours)**
  - Complete retraining the model with weighted learning (6 hours).

  - Evaluate the model's performance on rare letterforms, comparing CER and WER with the baseline model (4 hours).

  - Prepare and submit the midterm evaluation, reflecting on progress and discussing next steps with the mentor (6 hours, during July 14–18 evaluation period).

  - **Milestone**: Weighted learning integrated, midterm evaluation submitted.
- **Week 7 (July 14 – July 20, 2025): Constrained Beam Search Decoding (14.6 hours)**
  - Build a Renaissance Spanish lexicon from transcriptions and public domain sources (4 hours).

  - Implement constrained beam search decoding with a beam width of 5, integrating the lexicon to guide predictions (6 hours).

  - Test the decoding step and evaluate its impact on word-level accuracy (WER) (4.6 hours).

  - **Milestone**: Constrained beam search decoding implemented and tested.
- **Week 8 (July 21 – July 27, 2025): Final Model Training and Evaluation (14.6 hours)**
  - Retrain the CNN-RNN model with both weighted learning and constrained beam search decoding (6 hours).

  - Evaluate the final model on a test set of real historical texts (e.g., *Excaray-Vozes*), aiming for at least 80% accuracy (1 - CER) (4 hours).

  - Document the final model performance and evaluation results (4.6 hours).

  - **Milestone**: Final CNN-RNN model trained and evaluated.
- **Week 9 (July 28 – August 3, 2025): Iterate and Optimize Model (14.6 hours)**
  - Iterate on the model to achieve at least 80% accuracy, adjusting hyperparameters (e.g., learning rate, beam width) or training data as needed (6 hours).
  - Evaluate the layout recognition module (mAP, precision, recall) to ensure it maintains mAP of 0.8 with preprocessing improvements (4 hours).
  - Document the optimization process and final metrics (4.6 hours).
  - **Milestone**: Model optimized, achieving at least 80% text extraction accuracy.
- **Week 10 (August 4 – August 10, 2025): Documentation and Optional Deliverables (14.6 hours)**
  - Write comprehensive documentation for the text recognition pipeline, including setup instructions and usage guide (4 hours).
  - Prepare a white paper detailing the methodology, challenges (e.g., handling rare letterforms), and findings (optional deliverable) (4 hours).
  - Create a blog post showcasing the model's performance, with visual examples (optional deliverable) (4 hours).
  - Review deliverables with the mentor and make initial adjustments (2.6 hours).
  - **Milestone**: Documentation and optional deliverables completed.
- **Week 11 (August 11 – August 17, 2025): Finalize Deliverables (14.6 hours)**
  - Finalize the Jupyter Notebook summarizing the project, methodology, and results (4 hours).
  - Ensure all code, documentation, and datasets are properly organized in the repository (4 hours).
  - Prepare the final dataset of 200 annotated images for submission (4 hours).
  - Make final adjustments based on mentor feedback (2.6 hours).
  - **Milestone**: All deliverables finalized.
- **Week 12 (August 18 – August 25, 2025): Final Submission and**

Wrap-Up (14.6 hours)

- Submit the final deliverables to the target organization and prepare the final mentor evaluation (4 hours).
- Prepare a final report for GSoC evaluation (4 hours, submitted by September 1, 2025).
- Document any lessons learned and future work opportunities (e.g., scaling to manuscripts) (4 hours).
- Ensure all repository links and documentation are accessible (2.6 hours).
- **Milestone**: Project completed and submitted by the August 25, 2025, deadline.

Related Work

Text recognition for historical documents has been explored in projects like Tesseract's historical OCR efforts and academic research on CNN-RNN models for handwritten text recognition (e.g., CRNN by Shi et al., 2017). However, these efforts often focus on modern or handwritten texts and do not address the unique challenges of 17th-century Spanish printed sources, such as rare letterforms (e.g., long s [f]), ligatures, and printing imperfections. My project fills this gap by developing a specialized CNN-RNN model with weighted learning and constrained beam search decoding, tailored for Renaissance Spanish texts.

In Test I (`LayoutRecognition.pdf`), I implemented a layout recognition pipeline using YOLOv5 with transfer learning. I pre-trained YOLOv5 on a newspaper dataset (1200/100 train/val images), achieving an mAP of 0.65, precision of ~0.8, and recall of ~0.67, then fine-tuned it on a dataset of 40 Spanish documents, achieving an mAP of 0.8, precision of 0.81, and recall of 0.8. Despite challenges like MPS memory leakage on M2 Pro, which slowed training by a factor of 4x, the model effectively detected main text regions. This project extends that work by integrating layout recognition with text recognition, leveraging weighted learning to handle rare characters and a Renaissance Spanish lexicon to improve word-level accuracy. Within the target organization, The Digital Humanities Lab, this project complements existing efforts in historical document analysis by providing a robust tool for text recognition and transliteration of early printed works.

Results

The Test I (Layout Organization Recognition) project demonstrated promising results, with the following evaluation metrics for the YOLOv5 model: - **mAP (at IoU 0.5)**: Achieved an mAP of 0.8 on the Spanish document dataset, improved from 0.65 on the newspaper dataset, indicating strong performance

in detecting main text regions. - **Precision**: Achieved 0.81, reflecting high accuracy in identifying true text regions. - **Recall**: Achieved 0.8, showing the model's ability to detect most text regions while disregarding embellishments (e.g., margins, decorative elements).

These metrics highlight that the layout recognition approach—using YOLOv5 with transfer learning—is effective for identifying main text regions in historical Spanish book pages. However, the performance was constrained by several factors, as noted in `LayoutRecognition.pdf`: - **Computational Limitations**: Training on an M2 Pro with MPS was hindered by memory leakage, slowing training by a factor of 4x. Switching to CPU training was faster but suboptimal. Using CUDA-enabled resources (e.g., Google Colab) in this project will allow longer training and potentially improve mAP beyond 0.65 on the newspaper dataset. - **Dataset Size**: The Spanish dataset of 40 images limits the model's ability to generalize across diverse layouts. Expanding to 200 images in this project will improve robustness. - **False Negatives**: Some false negatives were observed due to lighting issues and background cluttering. Preprocessing improvements (binarization, grayscale conversion, denoising) will address this. - **Further Metrics**: Additional metrics like F1 score are available on the Comet website, as noted in `LayoutRecognition.pdf`.

For the text recognition component, this project aims to achieve at least 80% accuracy (1 - CER) on real historical texts. The weighted learning and constrained beam search decoding techniques will specifically improve performance on rare letterforms and reduce hallucinated outputs, ensuring the model is effective for real-world historical scenarios.

Biographical Information

I am Mohammad Zainuddin, a Master's student in Informatics at the Technical University of Munich. I have ~3 years of professional experience in software development, system design, and am currently pursuing a master's degree focusing on machine learning and computer vision. My relevant skills include:

- **Python**: Proficient in Python, with extensive experience in libraries like PyTorch, YOLOv5, OpenCV, and python-docx, which I used to develop the layout recognition pipeline in Test I (`LayoutRecognition.pdf`).
- **Machine Learning**: Experienced in deep learning models (e.g., CNNs, RNNs) and transfer learning, as demonstrated by my work on the Test I project, where I pre-trained YOLOv5 on a newspaper dataset (mAP: 0.65) and fine-tuned it on a Spanish document dataset (mAP: 0.8, precision: 0.81, recall: 0.8).
- **Computer Vision**: Familiar with object detection (e.g., YOLOv5), image preprocessing techniques (e.g., cropping, resizing), and evaluation metrics (e.g., mAP, precision, recall), which I applied in Test I to detect main text regions in historical images.
- **Open Source Contributions**: I have actively engaged with open source

communities by studying and building upon projects like VS Code.
- **Relevant Projects**: PRNU image fingerprint detection with CNNs, using LLMs to analyze crypto sentiments in markets, and currently working on a project to custom-train an LLM for technical analysis of stocks. Additionally, my Test I project involved layout recognition for historical documents using YOLOv5 with transfer learning, achieving an mAP of 0.8.

I have completed relevant coursework in machine learning, computer vision, and deep learning at TUM. I am confident in my ability to complete this project, given my experience with deep learning models, my skills in Python and machine learning, and my passion for historical document analysis.

Time Commitments

I am available to commit 14.6 hours per week to GSoC, totaling 175 hours over the 12-week coding period from June 2, 2025, to August 25, 2025. I have no significant outside commitments during this period, as my academic semester will be over, and I am not employed part-time. I may need to take a 3-day break in Week 5 (June 24–26, 2025) for a family event, which I will compensate for by working 16 hours in Week 6 (July 7–13, 2025). I will also allocate time for the midterm evaluation during July 14–18, 2025, and the final evaluation during August 25–September 1, 2025. I have reliable internet access and will be in regular contact with my mentor via email and chat throughout the program. I am based in Madrid, Spain, and have confirmed my eligibility to work in this country during the GSoC period.

Conclusion

This project offers a unique opportunity to advance historical document analysis within the open source community. By developing a CNN-RNN model for text recognition and transliteration of 17th-century Spanish texts, enhanced with weighted learning and constrained beam search decoding, I aim to deliver a valuable tool that will benefit researchers, educators, and developers. The Test I project has already demonstrated the effectiveness of layout recognition (mAP: 0.8), and the Results section highlights areas for improvement that this project addresses. With an expanded dataset (200 images), advanced techniques for handling rare letterforms, and a focus on achieving 80% text extraction accuracy, I am confident in delivering a robust solution. The pipeline is designed to scale to other historical languages or manuscripts in the future. I am excited to contribute to The Digital Humanities Lab and look forward to collaborating with mentors to make this project a success.