Project Proposal: Improving Renaissance Spanish OCR with CNN-RNN Hybrids and Weighted Learning

Kate O'Reilly | Email: katejudithoreilly@gmail.com | Github: https://github.com/kaoreill

Personal Background

Academic Background

My name is Kate O'Reilly, I'm 21 years old and from Ireland. I study Computer Science, Linguistics, and Spanish at Trinity College Dublin. I'm currently completing an Erasmus year at the Universidad Complutense de Madrid, where I take classes in both Spanish and English in Computer Science and Linguistics.

Why This Project?

As a student passionate about language, technology, and historical texts, this project feels like a perfect match. My academic focus has been shifting more and more toward natural language processing, AI, and applied linguistics, and I've always been fascinated by the challenge of bridging past and present through technology.

Transliteration of Renaissance-era Spanish texts is a field that desperately needs better tooling. These documents are rich in linguistic and historical insight but remain largely inaccessible because modern OCR systems can't process their unique features—like obsolete symbols, archaic letterforms, and irregular layouts.

Working on this project allows me to bring together my academic interests and contribute to a real-world problem that has significant value for scholars, linguists, and digital archives. It's exciting to imagine building something that can help unlock centuries-old texts and make them searchable, readable, and usable in modern digital spaces

Why This Approach?

Tools like Adobe OCR fail on early prints due to:

- Ligatures (e.g. /ſt/, /œ/)
- Obsolete diacritics (e.g. tilde over vowels for nasalization: /ã/)
- Non-standard glyphs (e.g. long /ſ/ vs short /s/)
- Degraded ink/paper

Synopsis

Develop a free, open-source OCR system for 17th-century Spanish printed texts using weighted CNN-RNN architectures, achieving ≥80% character accuracy (CER ≤20%).

Why This Work Matters

For HumanAI and the RenAIssance Project

Most existing OCR models are optimized for modern, standardized fonts in high-quality scans. These models fail dramatically when applied to Renaissance texts, which often include degraded print, ligatures, non-standard orthography, and typographical inconsistencies.

This project contributes directly to the RenAIssance mission: enabling AI systems to understand and transcribe historically rich but digitally underserved materials. Our approach focuses on integrating **historical context** and **linguistic awareness** into the core of the model, something that modern OCR engines don't offer.

For Scholars and Digital Archives

Improving OCR for Renaissance Spanish texts means empowering researchers to analyze language change, cultural history, and literary form without needing to transcribe everything manually. It means making rare books more accessible, searchable, and analyzable, helping scholars, librarians, and educators preserve and share cultural heritage.

For digital humanists, archivists, and corpus builders, this work offers tools that go beyond generic OCR, tools that can truly understand the specific challenges of early print.

And on a personal level, it's meaningful to contribute to something that reflects my academic interests and values so closely.

Deliverables

Phase 1: Preparatory Work

- Conduct a deep review of existing OCR architectures (CNN-RNN hybrids.)
- Explore and review Renaissance-era Spanish OCR datasets
- Research weighted learning and class imbalance techniques (e.g. weighted cross-entropy, class-aware sampling)
- Document common rare symbols, ligatures, and diacritics from 17th-century texts

Phase 2: Model Implementation

- Build baseline OCR model
- Apply weighted learning to underrepresented character classes
- Train and evaluate on a historical Spanish dataset

- Metrics: Character Error Rate (CER), Word Error Rate (WER)

Phase 3: Enhancement and Fine-Tuning

- Augment data to simulate printing degradation and rare symbol appearances
- Tune weights and loss functions (e.g. Character Error Rate (CER), Word Error Rate (WER))
- Compare against baseline and evaluate improvements
- Test on unseen texts and document generalization performance

Final Deliverables

- Trained model and inference scripts
- Evaluation report and graphs
- Codebase with documentation
- Sample transcriptions on historical Spanish texts
- README and contribution notes

Technical Approach

Loss Function Design

- **Primary Loss**: **Weighted Cross-Entropy Loss** to penalize misclassifications of rare symbols (e.g., *ç*, *ſ*, *ñ*).
- **Secondary Loss**: **Focal Loss** to focus training on hard-to-classify characters

Handling Extreme Class Imbalance

- **Smoothing**: Add Laplace smoothing (α =1e - 5) to avoid infinite weights for unseen symbols.
- **Contextual Rules**: Hardcode positional logic for ambiguous symbols (e.g., replace *ſ* with *s* only in non-final positions).

Data Augmentation for Rare Symbols

- **Synthetic Oversampling**: Use generative scripts to artificially increase rare symbol frequency:

Evaluation Protocol

- **Primary Metrics**: CER and WER (computed post-training).
- **Symbol-Specific Accuracy**: Track precision/recall for rare symbols (e.g., *ç*).

Timeline

If Data is Provided:

- Weeks 1-3: Annotate and preprocess real data.
- Weeks 4-12: Train on hybrid real and synthetic data.

If No Data is Provided

- Weeks 1-3: Generate 5,000+ synthetic samples with historical fonts/degradation.
- Weeks 4-12: Train on synthetic data only, with post-processing rules (e.g., $\int \rightarrow s$ substitution)

Week	Phase	Task	Contingency
1	Data Discovery	Identify datasets Start synthetic data generation	If no real data is secured by week 3, prioritise synthetic data
2	Data Prep	Annotate sample pages (if data is available) Build a synthetic pipeline for degradation/ligatures	
3	Model Prototyping	Implement a simple CNN-RNN baseline using data Test on 10 real samples Benchmark against Tesseract	If real data is inaccessible, validate synthetic performance first
4	Baseline Training	Train baseline model on synthetic data Calculate initial CER/WER	
5	Weighted Learning	Analyze symbol frequencies in available data Integrate class weights into loss	
6	Augmentation	Add augmentations (ink bleed, noise) to synthetic data Retrain model	
7	Real Data Integration	Fine-tune the model on data. Expand synthetic diversity	Prioritize high-impact symbols (e.g., $\int$, $ç$)
8	Evaluation	Compare CER/WER against Tesseract	
9	Mentor Feedback	Dedicate time to ensuring everything thus far is to the mentor's standards	This week is a buffer allowing for any

		and suggestions.	possible pivots
10	Optimization	Prune low-impact layers (reduce CNN filters)	
11	Documentation	Write user guides README on GitHub Clean up the repository	
12	Final Testing	Deploy the model on 10 unseen pages	

Data Scarcity Mitigation

As I would be formally collaborating with U.S. universities that have access to Renaissance-era Spanish texts, this significantly mitigates data pipeline risks
However, in case of data access being delayed:

Synthetic Data

- Use TextRecognitionDataGenerator with historical fonts (e.g., *Gothic*, *Fraktur*).
- Fallback to rule-based post-processing for common errors (e.g., */ſ/ → /s/*).

Ethical Considerations

Regional/Dialectal Bias
Renaissance-era Spanish texts vary significantly by region (e.g., Castilian vs. Andalusian prints) due to historical linguistic diversity. An OCR model trained on Castilian-dominated corpora may systematically fail on texts from other regions, erasing dialectal nuances and perpetuating cultural marginalization.

Poor performance on non-Castilian texts could skew linguistic studies, reinforcing the false narrative of Castilian as the "standard" Renaissance Spanish.

Mitigation

- Curate regionally balanced training data
- Add region-specific mappings (e.g., *ss* → *s* in Andalusian, *x* → *j* in older Castilian)
- Disclaimers like, "This model performs best on Castilian prints; Andalusian texts may require manual verification."

About Me

I am a final-year Computer Science student at Trinity College Dublin, specializing in **AI, NLP, and Spanish linguistics**, with hands-on experience building OCR tools during my Erasmus year at Universidad Complutense de Madrid (UCM). My unique qualifications include:

- **AI/ML Development:** Proficient in Python and PyTorch, with coursework in neural machine translation and transformer architectures.
- **Software Engineering:** Led Agile development of a Java-based Blackjack game at UCM using Scrum and UML modeling, skills directly applicable to structuring this OCR pipeline.
- **Applied Linguistics:** B2 Spanish certification + philological training in historical orthography through UCM's literature courses.

My dual training in computational methods (TCD) and Spanish paleography (UCM) allows me to tackle both technical and linguistic challenges in this project.

Glossary

Term	Definition
OCR	Optical Character Recognition: Extracting text from images
CNN	Convolutional Neural Network: Detects spatial patterns
RNN	Recurrent Neural Network: Models sequential data (e.g. text)
LSTM	Long Short-Term Memory: RNN variant for long sequences
CER	Character Error Rate: (Insertions + Deletions + Substitutions) / Total Characters
WER	Word Error Rate: Same as CER but at word level
Ligature	Joined characters e.g. /æ/
Diacritic	Accent marks, e.g. /ċ/