ID reader

Proof of Concept

Executive Summary

The cost of hiring third-party services for DNI image processing exceeds by orders of magnitude the cost associated with moving forward with the internal development of a solution. We are convinced that developing a service that satisfies the DNI image processing for entity extraction is a task within the possibilities of our area. This document describes the results of the prototyping of a first proof of concept -PoC- and has two central objectives: 1. confirm that the development of this service is possible by demonstrating a PoC on a data sample; and 2. share performance metrics to establish a *baseline* or basic expectations of the internal client regarding the quality of the service that can be offered.

Proof of Concept

Roof

The modeling was limited to training a model for the extraction of 4 entities agreed upon in previous conversations with the potential internal client:

1. DNI number
2. Name
3. Last name
4. Date of Birth (day/month/year)

Data

The modeling was carried out from a sample of 400 images obtained from non-confidential external sources, making training on public servers possible. The validation was carried out on a sample of 1,000 own images executed locally to preserve the confidentiality of the data.

Architecture

The Architecture of the PoC Model contemplates the following steps:

1. Filtered out
    - 1.1. Verification: a first filter to discriminate images that are not the type of document to be scanned.
    - 1.2. Integrity Control: resolution, luminosity, focus, integrity of the original document, etc.
2. Pre-processing
    - 2.1. Projective Transformations:
        - 2.1.1. Image Centering
        - 2.1.2. Orientation Adjustment
        - 2.1.3. Text Adjustment
3. Entity Extraction (NER)
    - 3.1. Barcode Reading
    - 3.2. Entity Extraction with alternative algorithm
    - 3.3. Validation of Extracted Data
4. Delivery
    - 4.1. Format: adaptation of results to output format based on the client's needs. Default format: JSONL.

Demo

link

Results

Performance Metrics

Entity	**Yes**	**Acc**	**Prec**	**Recall**	**F1**
DAYS	97.5	94.7	93.2	98.0	95.5
Name	90.3	77.5	72.6	77.8	75.1
Last name	88.1	77.5	76.1	83.8	79.3
Date	94.2	92.1	91.2	94.2	95.1

Comments on the Results Obtained

The prototyping experience revealed some points worth highlighting:

1. Coverage: the results of the experience show a coverage of 96% on the sample of 1000 images for validation. Where the 4% discarded includes images of documents with old format and discarded images.
2. Latency: the average time of each iteration was around 25 seconds of local execution on the CPU.
3. Performance: as expected, the entity extraction task yields different results depending on the type of entity to be extracted, its characters (alphabetical or numerical, or combined), its stability (e.g., variability of surnames and first names), its characteristics. design (typography, font size, bold, italics, etc.) and its location in the image (there are areas of lower resolution due to proximity to areas burned by the flash, etc.).
4. Adequacy: there are currently two types of national identity documents accepted in circulation but the frequency of occurrence of the old format is marginal (<3%) compared to the current issuance format. It will be necessary to perform ad-hoc training to incorporate it into the architecture with an additional volume of images. This is a requirement for the next iteration, not a blocker.
5. New Features: there is the possibility of extracting without additional latency a feature that could be of value to the internal client for incorporation into the file: the DNI image.

Final comments

The results obtained confirm a satisfactory prototyping performance. There is room for improvement for this baseline considering the following points:

1. Data: further training with a larger number of observations.
2. Architecture: adaptation to improvements.
3. New Features: extraction of additional text entities or images (i.e., ID photo).
4. Latency: moving from local execution to distributed execution on computers with GPUs to reduce processing times.
5. Integration with BarCode reading: BarCoder reading has an almost perfect performance but a poor coverage. Calling the barcode reader whenever possible (almost half of cases) should substantially improve the final metrics.