Document signature detection

Marek Sarvaš

November 17, 2024

1 Approach

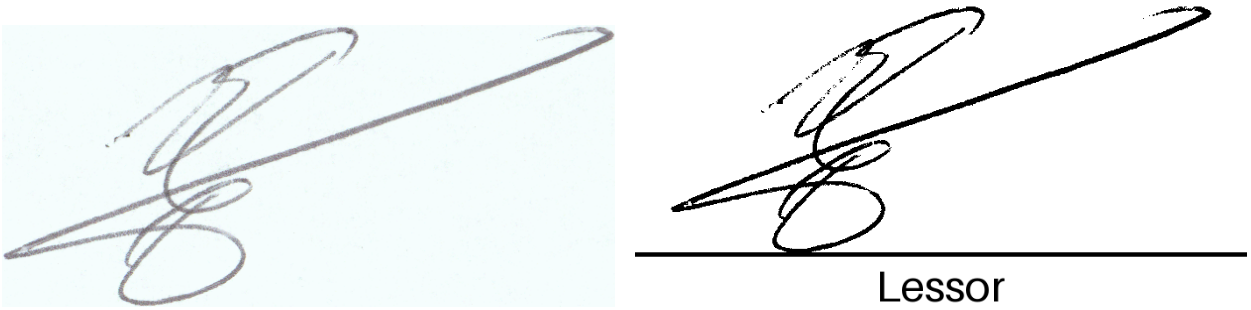
The biggest challenge for this task was the creation of a dataset that would contain documents with signatures, without signatures, and a combination of both. After the dataset creation I chose a two-step approach. First, train an object detection model to find signed and unsigned objects in the document. Second, extract text from the found bounding boxes. At the end I build a microservice in docker to accept a document png file and returns bounding boxes and classification of signatures and extracted text indicating who signed the document and who did not.

1.1 Dataset

I found two publicly available datasets, **Signature Verification Dataset**[1] and **SignverOD**[2], which I combined to create desired dataset for this task.

1. Step

Signatures from the **Signature Verification Dataset** where edited to white background and more black signature to better match the scanned documents from **SignverOD** dataset, shown in Figure 1. Next, the black line with text underneath it was added in the image under the signature. This was done to represent a person (or entity) that should be signed on the line.



(a) Original signature

(b) Edited signature

Figure 1: Main caption for the two images

2. Step

I filtered the **SignverOD** dataset to keep only documents with 1 or 2 signatures present. Using OpenCV python library, I replaced original signature with both, signed and unsigned, generated signatures. In addition, for documents containing two signatures, one signature was replaced with a signed image and the second with an unsigned image, creating a ”half” signed document. From every document, 2 (or 3) documents were created for the dataset.

1.2 Methods

For the object detection I chose the newest YOLOv11 model provided by ultralytics[3]. This model (and library) was used for the good object detection performance, fast inference time and because it was easy to train and evaluate. For the extraction of text from bounding boxes pre-trained OCR was used from the easyOCR[4] python library.

[1] https://www.kaggle.com/datasets/robinreni/signature-verification-dataset

[2] https://www.kaggle.com/datasets/victordibia/signverod/data

[3] https://docs.ultralytics.com/models/yolo11/

[4] https://github.com/jaidedai/easyocr

2 Results

The yolo11 model was trained on 1000 images of size 640 × 640 for 100 epochs. Confusion matrix on validation dataset is shown in Figure 2 and examples of signature detection are shown in Figure 3. The output of the docker microservice is beside the bounding boxes and document classification also the text extracted from the detected bounding boxes in json format, as follows:

```
{’x1’: 404, ’y1’: 2208, ’x2’: 1050, ’y2’: 2614, ’confidence’: 0.88,
’class_id’: 1, ’text’: [’Lessor’]}

{’x1’: 1662, ’y1’: 2299, ’x2’: 2602, ’y2’: 2622, ’confidence’: 0.87,
’class_id’: 1, ’text’: [’Tenant’]}
```

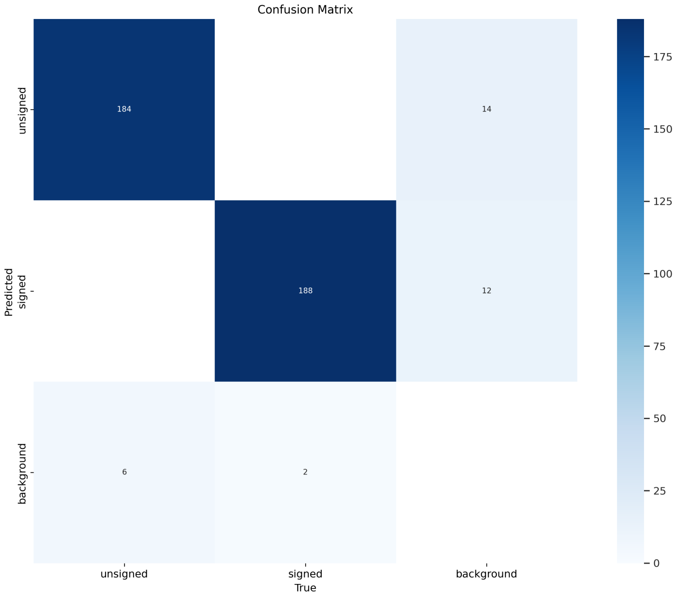


Figure 2: Confusion matrix for validation data.

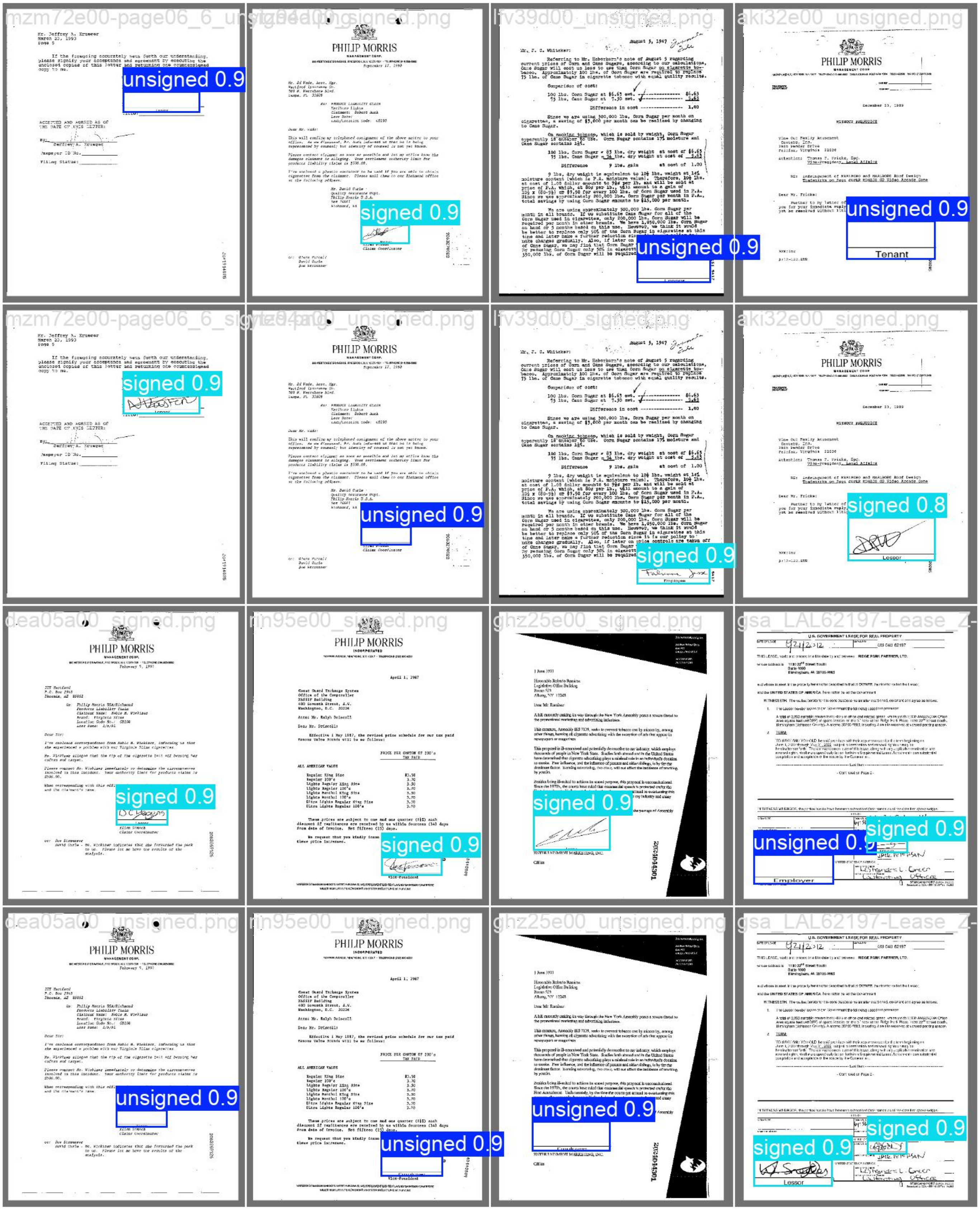


Figure 3: Examples of signature detection