Weekly Report N6

Enhancing the Model1 pipline and the Model2 OCR

DABBA DABBA'S RESOURCE FOR
EFFICIENCY,ALIGNEMENT AND MANAGEMENT

EDIHE AHMED JEDOU

HAMMAMI FARAH

SUMMARY

1.EXECUTIVE SUMMARY

This week's progress involved three main phases: OCR performance optimization and deployment preparation (Model 1), and OCR selection and advanced document reconstruction enhancement (Model 2).

1. MODEL 1:DATASET ANNOTATION

This week, we created and annotated a new dataset of 100 passport samples. The dataset was designed to increase model robustness and includes:

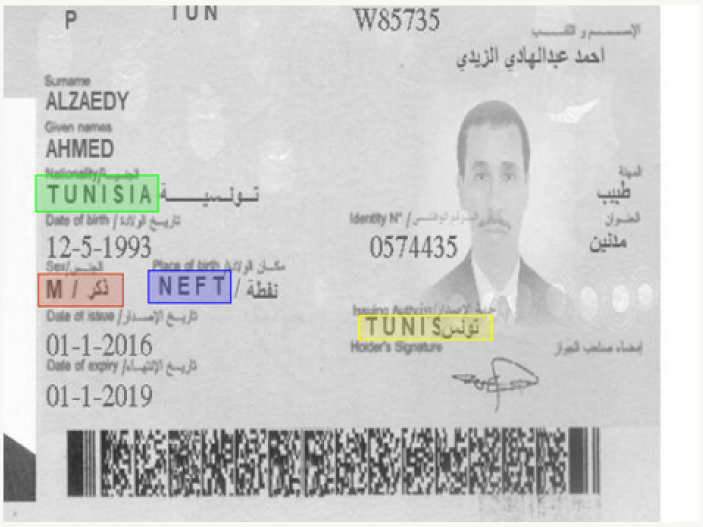
- Augmented samples to simulate variations in lighting, resolution, and orientation.
- Clear annotations of key fields such as Nationality, Sex, Place of Birth, and Issuing Authority.

Annotation was carried out using Label Studio, ensuring consistent labeling guidelines and high annotation accuracy. The final dataset is stored in YOLO-compatible format for seamless model training.

2. MODEL 1:YOLO MODEL ENHANCEMENT

Enhancements achieved:

- Improved detection accuracy on passport field recognition.
- Better bounding box precision, reducing false positives and improving field localization.
- The model now handles a wider range of passport designs and image conditions, making it more robust for real-world scenarios.



P TUN W85735

الإســـم و اللقـــب
احمد عبدالهادي الزيدي

Surname
ALZAEDY

Given names
AHMED

Nationality/الجنسية
TUNISIA تـونـسيـــة

Date of birth / تاريخ الولادة
12-5-1993

Sex/الجنس
M / ذكر

Place of birth / مكان الولادة
NEFT / نفطة

Date of issue / تاريخ الإصدار
01-1-2016

Date of expiry / تاريخ الإنتهاء
01-1-2019

المهنة
طبيب

Identity N° / المعرف الوطني
0574435

العنوان
مدنين

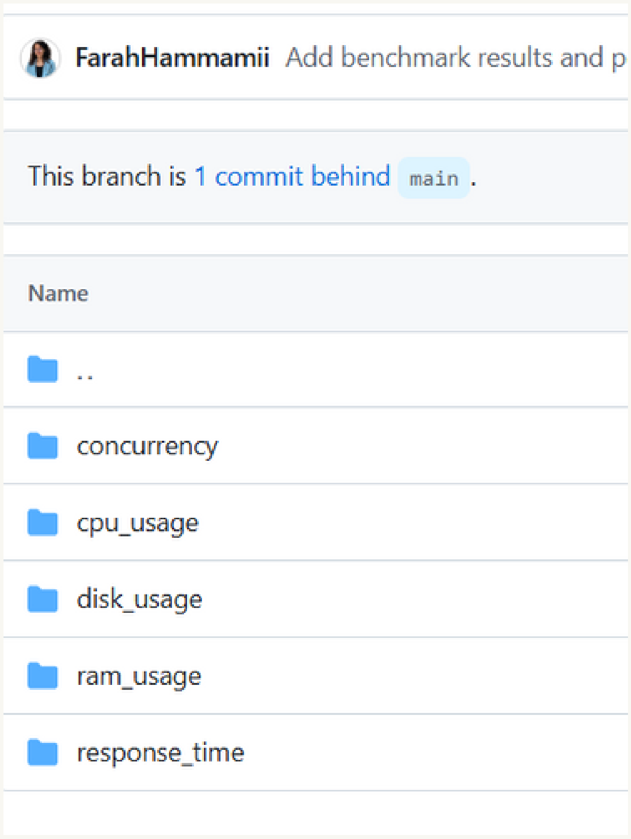
Issuing Authority / جهة الإصدار
TUNIS تونس

Holder's Signature
إمضاء صاحب الجواز

3. MODEL 1:PERFORMANCE OPTIMIZATION AND DEPLOYMENT PREPARATION

3.1 Performance Benchmarking

Comprehensive performance analysis was conducted to evaluate system resource usage and identify optimization opportunities.



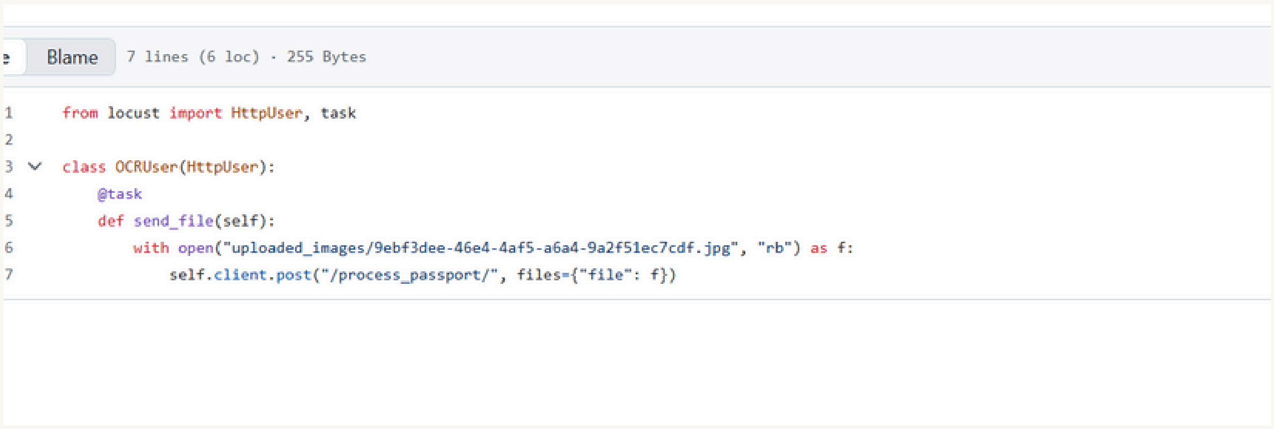
FarahHammamii Add benchmark results and p

This branch is 1 commit behind `main` .

Name

..
concurrency
cpu_usage
disk_usage
ram_usage
response_time

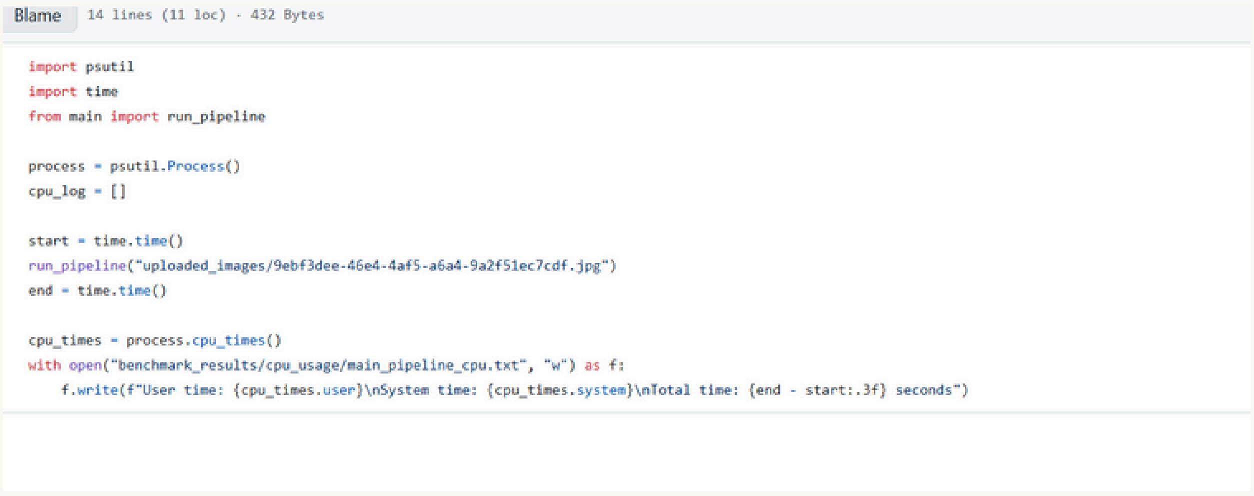
3.2 Concurrency



```python
from locust import HttpUser, task

class OCRUser(HttpUser):
    @task
    def send_file(self):
        with open("uploaded_images/9ebf3dee-46e4-4af5-a6a4-9a2f51ec7cdf.jpg", "rb") as f:
            self.client.post("/process_passport/", files={"file": f})
```

This script uses Locust to simulate an HTTP user sending an image file to the /process_passport/ endpoint.
It helps test the API's performance and behavior under simulated traffic conditions.

3.3 CPU Usage



```python
import psutil
import time
from main import run_pipeline

process = psutil.Process()
cpu_log = []

start = time.time()
run_pipeline("uploaded_images/9ebf3dee-46e4-4af5-a6a4-9a2f51ec7cdf.jpg")
end = time.time()

cpu_times = process.cpu_times()
with open("benchmark_results/cpu_usage/main_pipeline_cpu.txt", "w") as f:
    f.write(f"User time: {cpu_times.user}\nSystem time: {cpu_times.system}\nTotal time: {end - start:.3f} seconds")
```

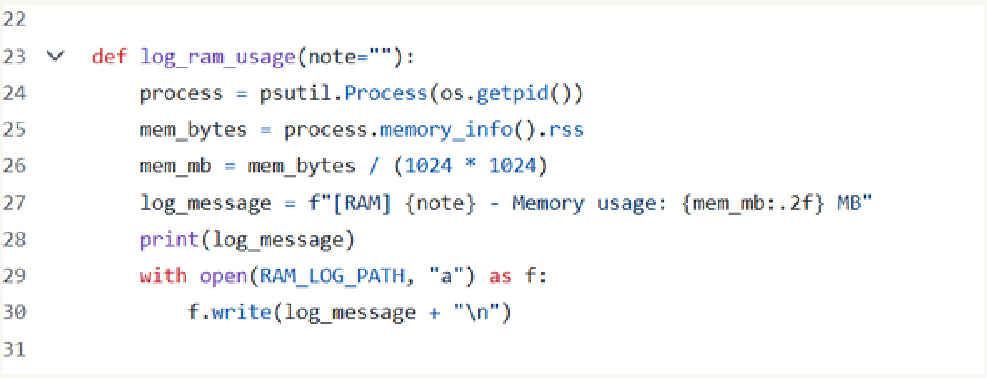
This script runs the run_pipeline function while tracking CPU usage and execution time with psutil.
It logs CPU statistics to a file to benchmark the processing performance of the main pipeline.

3.4 Disk Usage

A script was written to measure disk usage before, during, and after running the FastAPI server and making a sample request.
It helps evaluate storage consumption and the impact of file uploads and API operations on disk space.

3.5 Ram Usage :

This FastAPI application exposes the /process_passport/ endpoint, which uploads an image, processes it with run_pipeline, and returns extracted JSON data.
It also logs RAM usage at different stages to monitor memory consumption during processing.

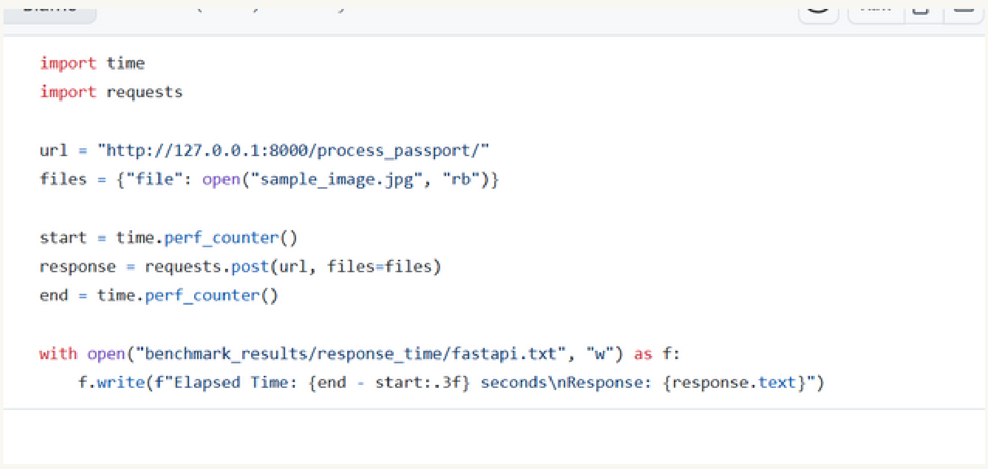


```python
def log_ram_usage(note=""):
    process = psutil.Process(os.getpid())
    mem_bytes = process.memory_info().rss
    mem_mb = mem_bytes / (1024 * 1024)
    log_message = f"[RAM] {note} - Memory usage: {mem_mb:.2f} MB"
    print(log_message)
    with open(RAM_LOG_PATH, "a") as f:
        f.write(log_message + "\n")
```

3.6 Response time Evaluation

A script was written to send a test image to the /process_passport/ endpoint and measure the total response time.

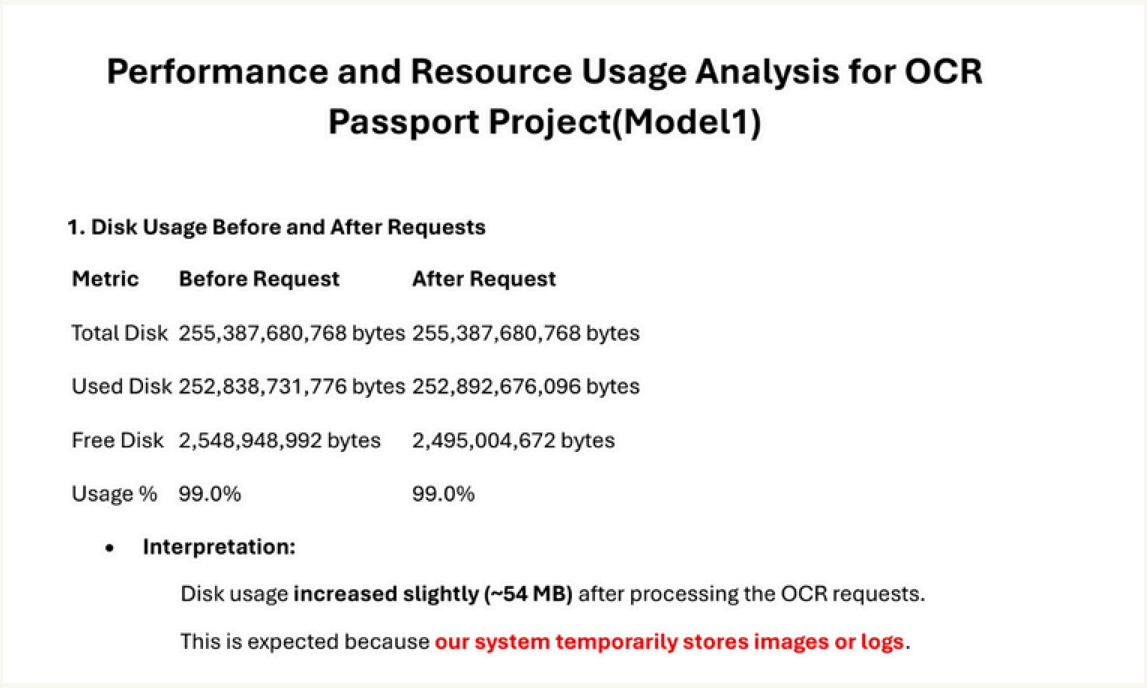
 It saves both the elapsed time and API response to a file for benchmarking purposes.



```python
import time
import requests

url = "http://127.0.0.1:8000/process_passport/"
files = {"file": open("sample_image.jpg", "rb")}

start = time.perf_counter()
response = requests.post(url, files=files)
end = time.perf_counter()

with open("benchmark_results/response_time/fastapi.txt", "w") as f:
    f.write(f"Elapsed Time: {end - start:.3f} seconds\nResponse: {response.text}")
```

3.7 Final Report

By the end of the evaluation , we generated a report with the results obtained in every single process , with reflection on the necessary steps and needs for the deployment phase.



Performance and Resource Usage Analysis for OCR Passport Project(Model1)

1. Disk Usage Before and After Requests

Metric	**Before Request**	**After Request**
Total Disk	255,387,680,768 bytes	255,387,680,768 bytes
Used Disk	252,838,731,776 bytes	252,892,676,096 bytes
Free Disk	2,548,948,992 bytes	2,495,004,672 bytes
Usage %	99.0%	99.0%

- **Interpretation:**

  Disk usage **increased slightly (~54 MB)** after processing the OCR requests.

  This is expected because **our system temporarily stores images or logs**.

4. MODEL 2: ENHANCED DOCUMENT RECONSTRUCTION SYSTEM

4.1 Smart Alignment Detection and Management

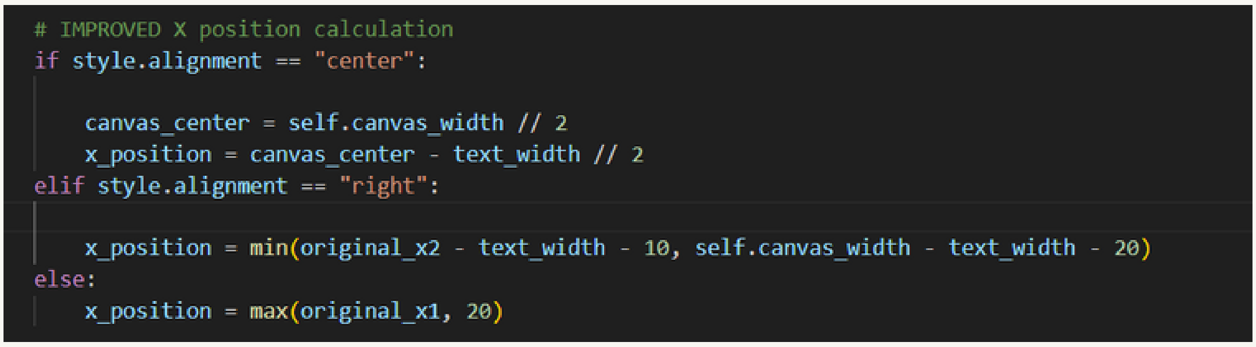
Problem Addressed:

Poor text alignment in reconstructed documents

Inconsistent positioning of text blocks

Loss of original document structure

Solution Implemented:



```
# IMPROVED X position calculation
if style.alignment == "center":

    canvas_center = self.canvas_width // 2
    x_position = canvas_center - text_width // 2
elif style.alignment == "right":

    x_position = min(original_x2 - text_width - 10, self.canvas_width - text_width - 20)
else:
    x_position = max(original_x1, 20)
```

Key Improvements:

Intelligent alignment detection based on document structure

Preservation of original document layout relationships

Responsive positioning system with boundary protection

4.2 Advanced Overlap Prevention System

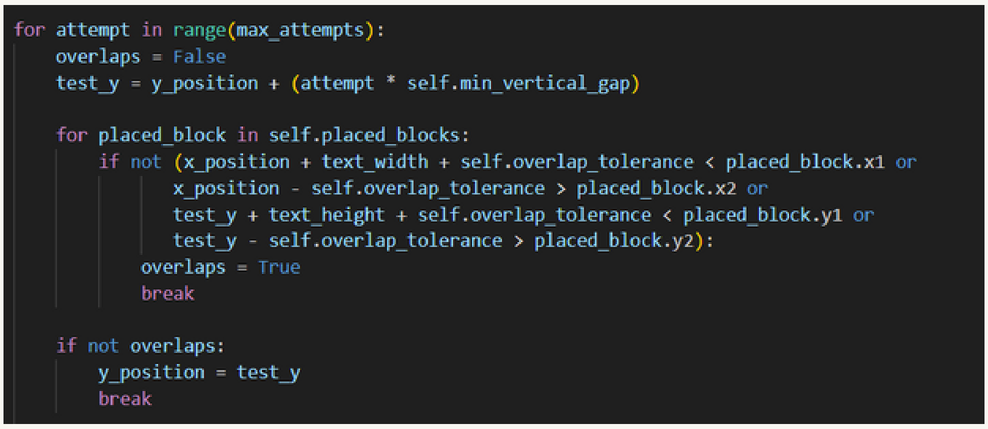
Problem Addressed:

Text blocks overlapping in reconstructed documents

Loss of readability due to content collision

Inefficient space utilization

Solution Implemented:



```python
for attempt in range(max_attempts):
    overlaps = False
    test_y = y_position + (attempt * self.min_vertical_gap)

    for placed_block in self.placed_blocks:
        if not (x_position + text_width + self.overlap_tolerance < placed_block.x1 or
                x_position - self.overlap_tolerance > placed_block.x2 or
                test_y + text_height + self.overlap_tolerance < placed_block.y1 or
                test_y - self.overlap_tolerance > placed_block.y2):
            overlaps = True
            break

    if not overlaps:
        y_position = test_y
        break
```

Key Features:

Dynamic collision detection algorithm

Iterative positioning with configurable tolerance

Minimum vertical gap maintenance

Fallback positioning strategies

4.3 Intelligent Paragraph Detection System

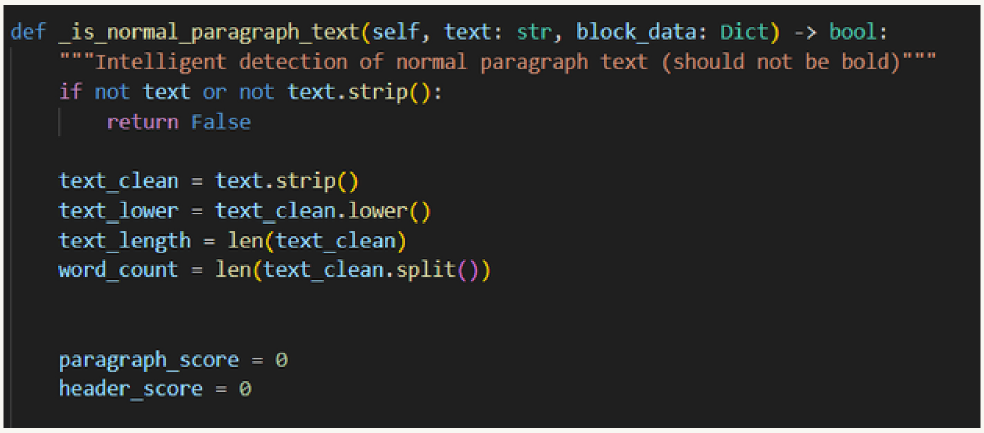
Problem Addressed:

Over-application of bold formatting to body text

Poor distinction between headers and paragraphs

Loss of document typography hierarchy

Solution Implemented:



```python
def _is_normal_paragraph_text(self, text: str, block_data: Dict) -> bool:
    """Intelligent detection of normal paragraph text (should not be bold)"""
    if not text or not text.strip():
        return False

    text_clean = text.strip()
    text_lower = text_clean.lower()
    text_length = len(text_clean)
    word_count = len(text_clean.split())


    paragraph_score = 0
    header_score = 0
```

Advanced Features:

Statistical Analysis: Document-wide text length and word count analysis

Pattern Recognition: Regex-based identification of paragraph vs. header patterns

Contextual Scoring: Multi-factor decision system with weighted scoring

Capitalization Analysis: Intelligent caps ratio evaluation

Common Word Detection: Statistical analysis of word frequency patterns

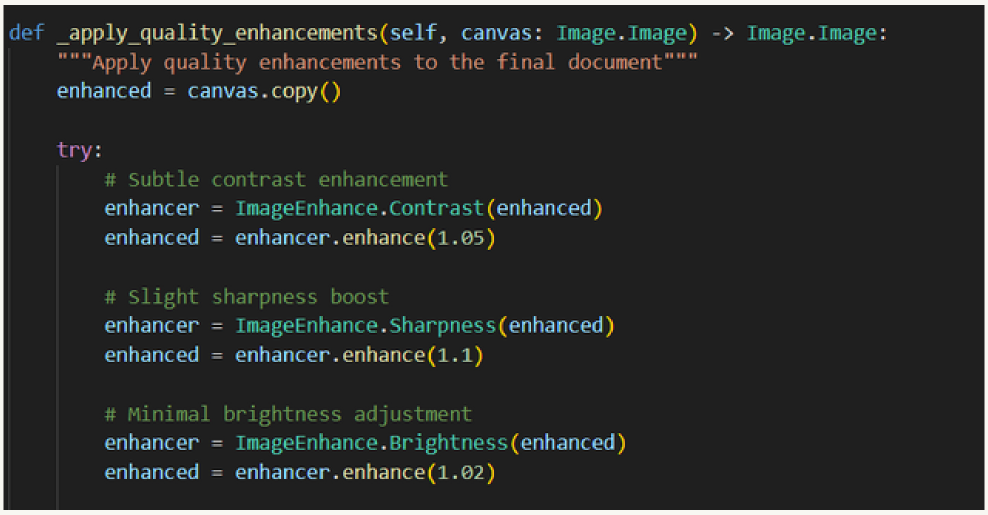
4.4 Smart Text Wrapping and Line Management

Intelligent Line Breaking: Context-aware word wrapping

Paragraph Preservation: Multi-line paragraph handling

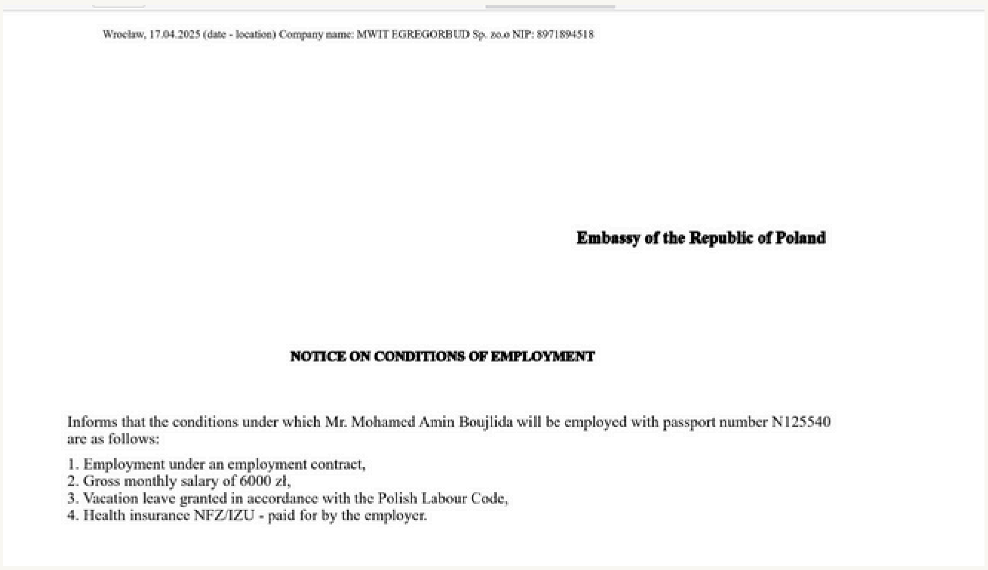
Spacing Optimization: Dynamic line spacing based on content type

4.5 Quality Enhancement Features



```python
def _apply_quality_enhancements(self, canvas: Image.Image) -> Image.Image:
    """Apply quality enhancements to the final document"""
    enhanced = canvas.copy()

    try:
        # Subtle contrast enhancement
        enhancer = ImageEnhance.Contrast(enhanced)
        enhanced = enhancer.enhance(1.05)

        # Slight sharpness boost
        enhancer = ImageEnhance.Sharpness(enhanced)
        enhanced = enhancer.enhance(1.1)

        # Minimal brightness adjustment
        enhancer = ImageEnhance.Brightness(enhanced)
        enhanced = enhancer.enhance(1.02)
```

These features helped recreate the layout in a clear and professional manner, ensuring that it **avoids overlaps, maintains visual harmony, and achieves consistency through careful selection of bold elements, typography, and an easily readable structure.**



Wroclaw, 17.04.2025 (date - location) Company name: MWIT EGREGORBUD Sp. zo.o NIP: 8971894518

Embassy of the Republic of Poland

NOTICE ON CONDITIONS OF EMPLOYMENT

Informs that the conditions under which Mr. Mohamed Amin Boujlida will be employed with passport number N125540 are as follows:

1. Employment under an employment contract,
2. Gross monthly salary of 6000 zł,
3. Vacation leave granted in accordance with the Polish Labour Code,
4. Health insurance NFZ/IZU - paid for by the employer.

5. CONCLUSION & NEXT STEPS

This week marked substantial progress in the refinement of Model 1, with enhancements targeting b**oth functional performance and output quality**. Comprehensive benchmarking was conducted to evaluate CPU, RAM, and disk utilization, as well as concurrency handling and response latency. In parallel, layout generation for Model 2 was significantly improved through the integration of a smarter **formatting system** capable of dynamically managing boldness selection, alignment, and overlap prevention, resulting in a more consistent and presentation-ready output.

For the upcoming and final project week, the primary objective will be to design and implement a **user-centric interface for Model 1**, transitioning from raw FastAPI endpoints to an interactive, visually coherent solution that can serve as a deployment-ready foundation. In addition, collaboration with Model 2 will be intensified, **focusing on OCR pipeline optimization, potential feature enhancements**, and contribution to its performance benchmarking framework where applicable.