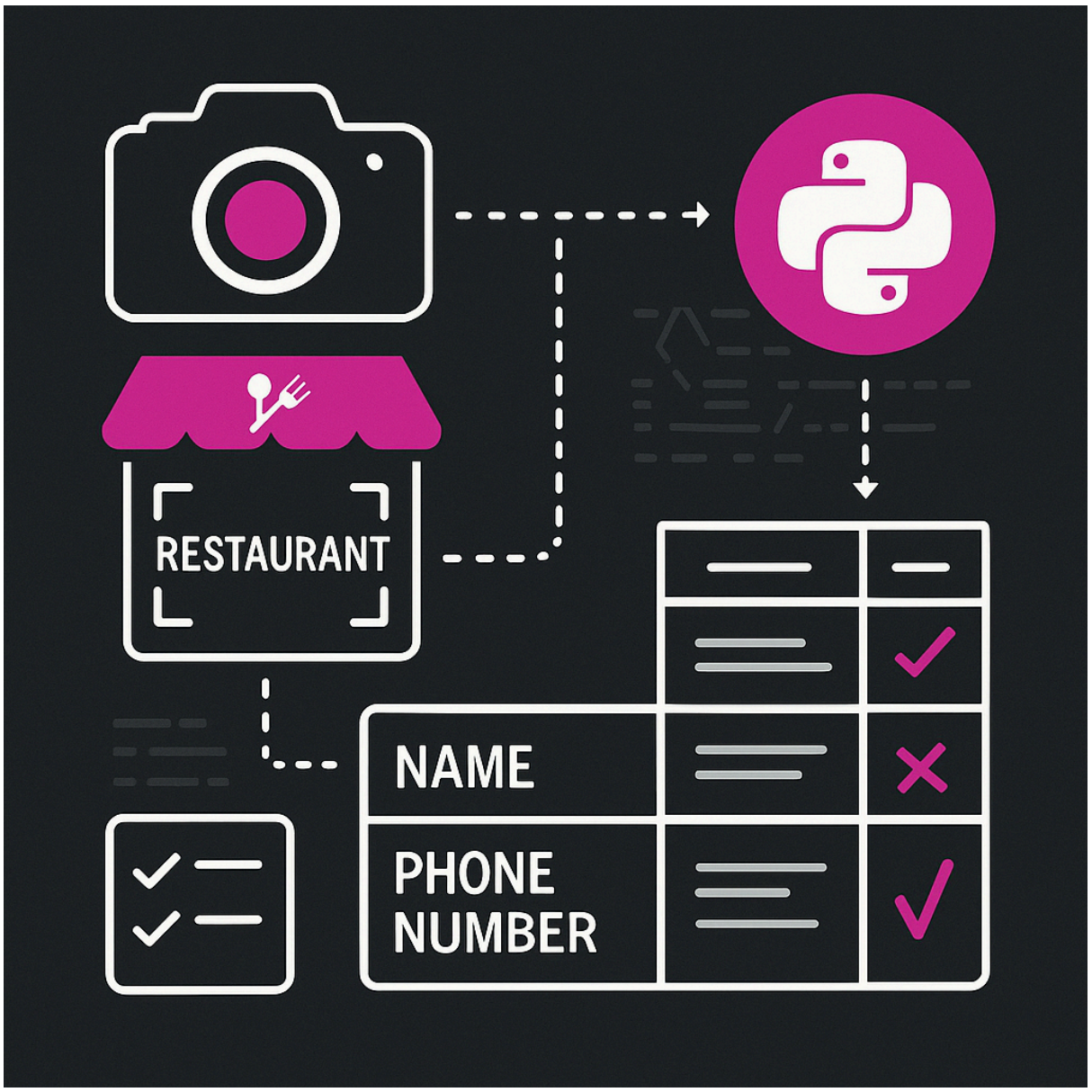
Snappfood | Commercial

Data Analyst Technical Task

Report of Task 3 - OCR Verification

Prepared By: **Amirreza Akbari**



Objectives

The primary goal of this task was to verify whether the names and phone numbers shown on restaurant signage images matched existing records in our structured dataset. The dataset, provided via Google Sheets, included the restaurant name, phone number, and a link to a corresponding storefront photo.

The task demanded careful attention to Persian text extraction, digit normalization, and real-world image noise — all of which posed unique challenges in ensuring OCR accuracy.

Tools and Environment

Several OCR engines and preprocessing techniques were explored to handle real-world Persian storefront images.

OCR Engines

- **Tesseract OCR**: Used with Persian language support (`fas`) and tested with multiple PSM settings (`6`, `11`). Accuracy was limited due to clutter, rotated text, and lack of layout awareness.
- **EasyOCR**: Offered better layout handling and Persian support, but performance degraded on visually complex signs.
- **Google Cloud Vision API**: Considered for its robust multilingual OCR but excluded due to access constraints.
- **PaddleOCR**: Adopted as the final engine. Configured with `lang='fa'` and `use_angle_cls=True`, it showed improved layout and multilingual handling. Deprecated API parameters (e.g., `cls`) were removed for compatibility.

Image Preprocessing (OpenCV)

- Grayscale, CLAHE, Gaussian blur, and adaptive thresholding were applied to enhance text visibility.
- Contour detection (via Canny and `findContours`) was used to isolate the sign region (ROI) for OCR, with fallback to full image when needed.
- Preprocessed images were saved for debugging (`debug_roi_*.png`, `debug_full_*.png`).

Matching and Logic

- Fuzzy name matching was done with `fuzzywuzzy.fuzz.ratio`, using a threshold of 85.
- Phone numbers were extracted using regex and matched by cleaned digit sequences.
- Text normalization and character replacement ensured consistency in Persian text.

Data Overview

The dataset provided for this task included a list of restaurants with three key fields:

- `Name`: The official name of the restaurant.
- `Number`: One or more phone numbers associated with the business.
- `Photo`: A Google Drive link to a storefront image.

The original data was downloaded as `task3_dataset.csv` and then processed to generate a new version with image file paths: `task3_dataset_with_image_paths.csv`. Each image was downloaded using its file ID extracted from the Drive link, saved locally, and indexed by its row number (e.g., `0.jpg`, `1.jpg`, etc.).

These images served as the visual source for OCR, but presented significant variability in:

- Image quality (some were low-resolution or blurry)
- Text placement and visibility (occluded, skewed, or partially cropped)
- Lighting and reflections
- Language mix (Persian, English, or both)

These factors introduced notable challenges in accurately extracting clean and complete text from each image, making preprocessing and fallback handling essential parts of the pipeline.

Methodology

The task was approached in three phases: data preparation, OCR experimentation, and consistency checking.

Data Preparation

- Extracted file IDs from Google Drive links and downloaded all images (`0.jpg` to `9.jpg`).
- Created a new dataset (`task3_dataset_with_image_paths.csv`) with local image paths.
- Normalized `Name` and `Number` fields to handle Persian characters and digits.

Image Preprocessing

Using OpenCV, images were prepared for OCR by:

- Grayscale conversion
- Contrast enhancement (CLAHE)
- Noise reduction (Gaussian blur)
- Adaptive thresholding
- Contour detection to crop a probable Region of Interest (ROI); fallback to full image when needed

OCR Trials

Multiple OCR engines were tested:

- **Tesseract**: Fast but unreliable; misread rotated or cluttered signs.
- **EasyOCR**: Performed relatively better on Persian-English signage; retained for final use.
- **Google Cloud Vision API**: Considered, but not used due to access limitations.
- **PaddleOCR**: Explored briefly; required additional setup and tuning.

Due to limited time and performance across all tools, EasyOCR was kept in the code, but no solution was finalized as fully reliable.

Extraction and Matching

- OCR output was parsed line-by-line:
    - Persian text lines were filtered for names.
    - Digit sequences (≥8 digits) were extracted as phone numbers.
- Fuzzy matching (using `fuzzywuzzy.ratio`) was applied between `OCR_Name` and `Normalized_Name`.
- Number matches were counted using exact string comparison after normalization.

Evaluation

Despite building a structured pipeline and testing multiple OCR engines, the overall performance of the system remained limited.

- **OCR Accuracy**: Most images contained cluttered backgrounds, low contrast, or angled signage, which caused OCR engines to misread or hallucinate text. Names were often fragmented or replaced by unrelated Persian words, while phone numbers were frequently incomplete or misaligned.
- **Name Matching**: Even with normalization and fuzzy matching, name accuracy was low. Only a few matches exceeded the 85% similarity threshold, and many others failed due to partial recognition or irrelevant extractions.
- **Phone Number Matching**: Digit extraction showed slightly better results but still suffered from noise. Partial numbers or incorrect groupings led to false negatives, and small fragments (like area codes) sometimes matched by coincidence.
- **Engine Comparison**: EasyOCR gave relatively more consistent outputs than Tesseract but still struggled with complex signage. No engine performed reliably across all images.

Overall, the pipeline functioned end-to-end, but the outputs were not dependable enough for real-world verification without manual review.

Final Outputs

The following deliverables were produced:

- `task3.ipynb`: The main Jupyter Notebook containing all code, experiments, and commentary
- `task3_dataset_with_image_paths.csv`: The enriched version of the original dataset, with local image file paths
- `task3_output_easyocr.csv`: The final output containing extracted OCR results and consistency checks

Conclusion

This task involved building an OCR-based verification system to compare signage images with structured restaurant data. While the pipeline was implemented end-to-end and multiple OCR engines were tested, the results were ultimately limited by the complexity of the input images and the capabilities of the available tools.

Several engines, including Tesseract, EasyOCR, and PaddleOCR, were explored. EasyOCR was retained in the final implementation due to its relative ease of use and support for mixed-language text. However, no approach produced consistently reliable outputs, especially on low-quality or cluttered images.

The matching logic, normalization methods, and overall structure of the solution were solid, but the accuracy of name and number extraction was not high enough for practical deployment. The final result demonstrates a functional baseline but would require significant improvements for production use.

Suggestions for future improvements include:

- Training a custom OCR model fine-tuned on Persian business signage
- Incorporating text detection models (e.g., CRAFT, DBNet) to localize text regions before OCR
- Manually annotating a small set of images to support supervised evaluation or active learning

In its current form, the task is a good representation of the challenges in applying OCR to real-world Persian signage, and lays the groundwork for more robust solutions.