International Scientific Conference 20 UNITECH 26

INTERNATIONAL SCIENTIFIC CONFERENCE
19-21 November 2026
GABROVO

COMPARATIVE ASSESSMENT OF TESSERACT AND PADDLEOCR FOR MOBILE-CAPTURED DOCUMENT RECOGNITION

Plamen Nakov*

Technical University of Sofia, Bulgaria
**Corresponding author: p.nakov@tu-sofia.bg*

Abstract
Mobile-captured identity documents present coupled challenges in text localisation, transcription and field assignment. This paper provides an engineering reanalysis of a technical study reporting Tesseract and PaddleOCR results for 1,716 Bulgarian identity-document images. Reported exact-match rates increase from 84.67% to 98.60% for the machine-readable zone and from 73.30% to 81.10% for the document number. Joint correctness of both fields increases by 7.80 percentage points. The corresponding relative error reductions are 90.87% and 29.21%. Approximate CPU times of two and three seconds per image motivate a conditional accuracy–latency assessment: a simple time-equivalent loss model changes its preferred engine at an error penalty of about 12.8 seconds. This threshold is a derived scenario, not a measured operating cost. Architectural evidence, reported observations and proposed validation procedures are explicitly separated. The study adds a stage-oriented error taxonomy and a reproducibility specification covering field definitions, paired outputs and execution settings. Its conclusions concern the reported implementations; neither a quantitative Cyrillic benchmark nor document-authenticity validation is established.

Keywords: optical character recognition; Tesseract; PaddleOCR; identity documents; Cyrillic.

INTRODUCTION

The engineering value of an OCR system depends on whether it extracts the fields required by the application, not merely on whether its output resembles readable text. A single incorrect identifier can invalidate a record even when most characters on the document are correct. For smartphone photographs, text recognition must also accommodate the capture conditions and the association of recognised strings with their intended fields.

A technical case study compares Tesseract and PaddleOCR on 1,716 photographs of Bulgarian identity cards and passports. It reports exact-match accuracy for the machine-readable zone (MRZ), the document number and their joint correctness. Names and addresses in Cyrillic are discussed qualitatively. The reported comparison is useful, but interpreting it requires a distinction between the implemented system, the capabilities of an OCR library and the performance of a downstream decision process.

This paper reanalyses those aggregate results rather than rerunning the experiment. The contributions are a field-level assessment with transparent error transformations, a conditional accuracy–latency decision model, and an evidence specification for a reproducible follow-up comparison. The focus is engineering selection between the two reported implementations. The probability of joint success after future field-specific upgrades is a different question and is not presented here as another measured result.

The available evidence consists of the technical manuscript, its summary table and approximate CPU timings. Individual images, ground-truth strings, per-image predictions and executable configurations were not supplied. Accordingly, the numerical reanalysis preserves the reported percentages; it does not reconstruct observations, estimate

This is an open access article licensed under Creative Commons Attribution 4.0 International
Print ISSN: 1313-230X, Online ISSN: 3033-1404

missing hardware properties or invent a train–test partition.

ARCHITECTURE AND CONFIGURATION

Recognition is neural in both systems

Tesseract 4 introduced an LSTM-based recognition engine, retained in the subsequent system family [2]. The relevant contrast is therefore not a non-neural algorithm against deep learning. Tesseract combines page and line processing with sequence recognition, whereas a typical PP-OCR pipeline exposes text detection, crop processing and recognition as separately configurable stages. This difference creates different diagnostic and integration opportunities; it does not establish a universal accuracy ordering.

PP-OCRv2 [3] and PP-OCRv3 [4] are distinct generations. Differentiable Binarization (DB) localises text using a segmentation-based detector with a differentiable binarisation mechanism [5]. SVTR is a scene-text recogniser based on a single visual model [6]. The PP-OCRv3 report describes replacing the CRNN recogniser with an SVTR-based design [4]. These architectural publications establish what the respective models implement, not which model produced an undocumented application result.

The source study simultaneously refers to a default PP-OCRv2 configuration and an SVTR-based recogniser. Without model identifiers or execution logs, that inconsistency cannot be resolved by silently relabelling the experiment as PP-OCRv3.

Orientation, preprocessing and field assignment

The text-line angle classifier documented for PaddleOCR 2.x distinguishes 0° from 180° [7]. This is different from classifying the orientation of an entire document into four right-angle rotations. Likewise, a document-understanding architecture containing table, layout and formula modules is not automatically the architecture used to obtain a two-field OCR result. Components should be included in an experimental description only when their execution is documented.

Tesseract also provides meaningful controls for image quality and segmentation. Its documentation discusses binarisation, deskewing, crop borders, page-segmentation modes and dictionaries or character patterns [8]. Consequently, it is unjustified to describe its errors as intrinsically uncorrectable while treating PaddleOCR errors as automatically recoverable through fine-tuning. Both implementations require task-appropriate inputs and a controlled evaluation of their settings.

A detector locates text; a recogniser transcribes a crop; field assignment decides which transcription represents the required item. A correctly read date assigned to the document-number field is still an application error. Conversely, a recogniser cannot recover a character that was removed by an upstream crop without introducing information not present in that crop. These distinctions motivate the stage-oriented interpretation in Table 1.

Table 1. Diagnostic stages and the evidence needed to assess them; proposed analysis, not measured failure shares.

Stage	Relevant failure	Required evidence
Capture	Blur, glare or missing content	Original image and quality annotation
Detection	Missed or incomplete field	Field polygon and actual crop
Reading	Incorrect crop transcription	Reference string and raw output
Mapping	Correct text, wrong field	Field-to-string association
Validation	Plausible but incorrect output	Exact match separate from checks

The source describes task-oriented tuning for Tesseract and no task-specific fine-tuning for PaddleOCR. That is a property of the compared implementations, not evidence of equal optimisation effort. A deployment-oriented comparison may legitimately assess each system as configured in practice; a controlled algorithm comparison additionally needs declared tuning resources and frozen evaluation conditions.

MATERIAL AND EVALUATION METHOD

Units of assessment and exact matching

The source corpus comprises 1,716 mobile-captured identity-document images. Separate counts for identity cards, passports, repeated captures, document holders and capture devices are not available. The number therefore describes the reported image collection, not a demonstrated count of independent people or documents. The field-specific denominators and any exclusions also need confirmation from the original evaluation records.

Let $x_{i,f}$ be one when the output for field f of image i exactly matches its reference transcription and zero otherwise. For a declared set of N eligible images, field accuracy is:

$$p_f = \frac{1}{N}\sum_{i=1}^{N} x_{i,f} \quad (1)$$

The labels M and D denote the MRZ and document number. When both fields are required on the same images, joint correctness is:

$$q = \frac{1}{N}\sum_{i=1}^{N} x_{i,M}\, x_{i,D} \quad (2)$$

Equation (2) is a text-extraction metric. It does not test document authenticity, ownership, biometric matching or resistance to fraud. The source term "document verification rate" is therefore reported here as joint OCR correctness. Successful parsing of an identifier is not equivalent to a successful identity-verification transaction.

Exact field matching must also be distinguished from character accuracy and word accuracy. One substitution, insertion or deletion makes the entire field incorrect. A comparison with a character-level score from another dataset would change both the unit of analysis and the loss function. No numerical Cyrillic accuracy can be inferred from the MRZ or document-number percentages.

A reproducible MRZ assessment must identify the document format and whether the scored unit is the whole zone or an individual line. ICAO Doc 9303 specifies document-type-dependent machine-readable layouts and common conventions [9]. It is not appropriate to assume that every identity-document MRZ has the same number of lines. The source uses both zone- and line-oriented wording; its original evaluation unit must be confirmed before exact replication.

Normalisation and numerical transformations

The scoring contract should state how whitespace, line breaks, case and filler characters are handled. Any permitted normalisation must be fixed before evaluation and applied consistently to predictions and references. Character substitutions selected after looking at the answer are corrections, not neutral normalisation. For Cyrillic text, mapping a visually similar Latin character to a Cyrillic one can change the string even when the rendered shapes appear alike.

For an accuracy increase from p_T to p_P, the percentage-point gain is 100 times their difference. The relative reduction in field errors is instead:

$$R = \frac{p_P - p_T}{1 - p_T} \times 100\% \quad (3)$$

These transformations use the reported percentages as point inputs. Printed trailing zeros do not establish the effective precision of the underlying measurements. The analysis therefore retains the original accuracy values, rounds derived comparisons sensibly and avoids constructing integer success counts from percentages with undocumented rounding or aggregation rules.

RESULTS AND ENGINEERING ANALYSIS

Field accuracy and residual errors

Table 2. Exact-match accuracy, %. The corpus contains 1,716 images; field denominators require confirmation.

Metric	Tesseract	PaddleOCR
MRZ	84.67	98.60
Document number	73.30	81.10
Both fields correct	73.30	81.10

Table 2 reproduces the source results. MRZ accuracy increases by 13.93 percentage points, while document-number and

joint accuracy each increase by 7.80 percentage points. The residual MRZ error rate decreases from 15.33% to 1.40%; the number error rate decreases from 26.70% to 18.90%. Equation (3) gives relative error reductions of 90.87% and 29.21%, respectively.

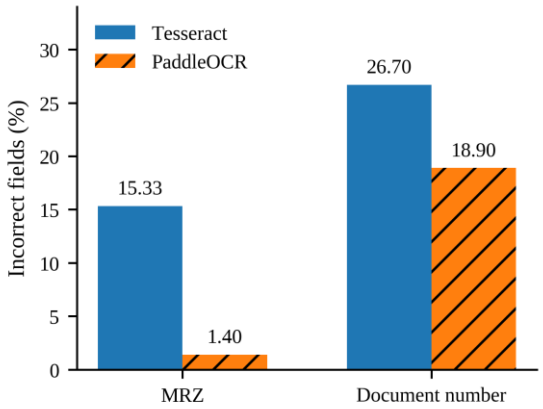


Fig. 1. Residual field-error rates calculated as 100% minus the reported accuracy. These are transformations of the source aggregates, not new observations.

Figure 1 shows why an accuracy-only summary can obscure the remaining engineering problem. The large reduction in MRZ errors does not translate into an equally large joint gain. Under the stated two-field requirement, incorrect numbers remain sufficient to prevent complete extraction. The useful conclusion is field-specific: the reported PaddleOCR implementation improves both fields, but the residual burden is concentrated more strongly in the document number.

The joint rate equals the number rate in both reported rows. This equality is a property of those aggregates at their stated precision, not a general consequence of requiring two correct fields. Joint correctness must be measured on aligned observations; it cannot normally be replaced by the smaller marginal accuracy. Nor does the table reveal which images improve or deteriorate when the engine changes.

For a paired comparison, let n_{01} count cases incorrect under Tesseract and correct under PaddleOCR, and let n_{10} count the reverse transitions. The net accuracy difference is their difference divided by the common denominator. Aggregate accuracies identify that net difference but not the two transition counts separately. A larger overall score can therefore coexist with regressions on some documents. The missing paired records prevent a defensible paired significance calculation or a measured regression analysis.

Latency, useful output and selection criteria

The source reports approximate mean CPU times of 2 s per image for Tesseract and 3 s for PaddleOCR. These values suggest a 1.5-fold latency ratio in that setting, but no hardware-independent speed claim is warranted. Processor model, thread count, image dimensions, initialisation, warm-up and inclusion of preprocessing are insufficiently documented. Neither tail latency nor variability across images is available.

A simple serial-processing scenario illustrates the importance of the objective. With mean time t_e and joint correctness q_e, the expected number of correctly extracted two-field records per second is:

$$g_e = \frac{q_e}{t_e} \tag{4}$$

Using the reported point inputs gives about 0.367 correct records/s for Tesseract and 0.270 correct records/s for PaddleOCR. These are derived ratios, not measured throughput. They assume serial execution with no retries, batching, parallelism or review stage. They do not contradict the higher PaddleOCR accuracy; they answer a different question about useful output under a fixed processing-time budget.

To combine latency and extraction errors explicitly, consider the illustrative loss $J_e = t_e + \kappa(1 - q_e)$, where κ is a user-selected penalty in seconds-equivalent per incorrect record. PaddleOCR is preferred by this model when:

$$\kappa > \frac{t_P - t_T}{q_P - q_T} \tag{5}$$

Substitution gives a crossover of approximately 12.8 s-equivalent. Below that penalty, the shorter reported execution time dominates; above it, the higher joint accuracy dominates. This is a sensitivity

calculation, not a measured review cost or a recommendation to value every error identically. Incorrect outputs are not automatically detectable online, and practical loss may distinguish rejected records from silently accepted errors.

ERRORS, REPRODUCIBILITY AND LIMITS

Stage-oriented diagnosis

The qualitative source observations include blur, cropped MRZ lines, faint numbers and imperfect localisation. These observations justify inspecting the crop stage, but they do not quantify its contribution. A useful follow-up should label failures by the first observable breakdown: unusable capture, missed field, incomplete crop, incorrect transcription or incorrect field assignment. Multiple causes may coexist, so an explicitly defined primary-cause rule and optional secondary labels are preferable to arbitrary percentages.

Recognition on manually annotated crops would test whether errors remain when localisation is supplied. It is a diagnostic condition, not a realistic fully automatic deployment result. Improvement on these crops would support further localisation work, whereas persistent errors would motivate checking script coverage, resolution, the decoding dictionary or the recogniser. Detection overlap alone is insufficient: a small crop error can remove one decisive character while retaining most of the text-region area.

For Cyrillic names and addresses, the source reports fewer character substitutions with PaddleOCR but provides no separate exact-match or character-error statistics. A numerical follow-up needs field-specific references, character inventories and a stated treatment of punctuation and abbreviations. Results should be stratified by field type and string length rather than transferred from an alphanumeric identifier. The present paper retains only the qualitative observation.

A minimum reproducibility specification

Table 3. Minimum records for the proposed reproducible comparison. These records were not supplied in full.

Record group	Required specification
Dataset	Eligible images, grouping, split and field denominators
Reference text	Annotation procedure and exact-match normalisation
Configuration	Versions, weight hashes, dictionaries and preprocessing
Paired outputs	Raw text, field assignment, errors and timeouts
Execution	Hardware, threads, warm-up, timing boundary and repeats

Table 3 specifies the minimum records needed to convert the case study into a reproducible comparison. Model files should be identified by version and checksum, with language dictionaries, input dimensions, segmentation settings and preprocessing preserved. Both engines should process the same eligible images. Missing detections, empty outputs and timeouts must remain visible rather than being silently removed from the denominator.

When training or parameter selection is introduced, repeated photographs of the same document or person should be grouped before partitioning. Tuning should use development data, while the final test set remains untouched. The original study does not document these safeguards sufficiently to establish that they were implemented; they are requirements for the proposed follow-up, not retrospective claims about its data handling.

Runtime reporting should distinguish cold-start from warmed execution and provide the measurement boundary, repeated-run variability and latency quantiles. Raw predictions, normalised strings and format-check outcomes should be retained separately. Input images and personal identifiers require an appropriate access and retention arrangement; this paper neither reproduces personal documents nor asserts an unverified permission or approval.

Scope of the conclusion

The study supports a technical interpretation of the reported configurations, not a universal ranking of OCR libraries or their latest releases. It does not establish the accuracy of a newly trained model, an independently validated Cyrillic benchmark or a causal attribution of the gains to DB or SVTR. A forecast of more than 95% joint success after localisation changes is not included among the measured results because no corresponding post-upgrade observations were supplied.

CONCLUSION

The reported PaddleOCR implementation raises joint MRZ–number correctness from 73.30% to 81.10%, with markedly different residual-error reductions for the two fields. This field-oriented interpretation is more informative than treating the outcome as a single generic OCR score. The architecture comparison also shows why model generation, crop processing and field assignment must be recorded separately before numerical gains can be attributed to a specific component.

Engine selection remains objective-dependent. The approximate source timings favour Tesseract in a serial useful-output calculation, while a time-equivalent error penalty above about 12.8 s favours PaddleOCR in the stated illustrative loss model. Neither calculation replaces deployment measurements. The practical next step is a frozen, paired evaluation with explicit field definitions and stage-level failure records, so that improvements in accuracy, latency and retained correct cases can be assessed together.

Data and preparation statement

Accompanying Python calculations reproduce the derived quantities; they contain no new OCR observations. ChatGPT (OpenAI) assisted with structure, language editing and analytical development.

REFERENCES

[1] Kravets A.G., Semenochkin D.O., Markov A.K. Development of a New Experimental Method for Evaluating OCR Tools for the Task of Digital Document Classification. Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies 2024;(3):114–126. doi:10.17308/sait/1995-5499/2024/3/114-126.

[2] Tesseract OCR. Tesseract User Manual. https://tesseract-ocr.github.io/tessdoc/ (accessed 4 October 2026).

[3] Du Y. et al. PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System. arXiv:2109.03144, 2021. doi:10.48550/arXiv.2109.03144.

[4] Li C. et al. PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. arXiv:2206.03001, 2022. doi:10.48550/arXiv.2206.03001.

[5] Liao M., Wan Z., Yao C., Chen K., Bai X. Real-Time Scene Text Detection with Differentiable Binarization. Proceedings of the AAAI Conference on Artificial Intelligence 2020;34(07):11474–11481. doi:10.1609/aaai.v34i07.6812.

[6] Du Y. et al. SVTR: Scene Text Recognition with a Single Visual Model. Proceedings of IJCAI 2022, pp. 884–890. doi:10.24963/ijcai.2022/124.

[7] PaddleOCR. Text Direction Classification. Version 2.10.0 documentation. https://www.paddleocr.ai/v2.10.0/en/ppocr/model_train/angle_class.html (accessed 4 October 2026).

[8] Tesseract OCR. Improving the Quality of the Output. https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html (accessed 4 October 2026).

[9] ICAO. Doc 9303: Machine Readable Travel Documents. Parts 3–5, 8th ed., 2021. https://www.icao.int/publications/doc-series/doc-9303 (accessed 4 October 2026).