# A Hybrid Document Reading, Few-Shot Classification, and LLM Extraction Pipeline for Accounting Document Processing

**David Slavik**
Artificial Intelligence Laboratory
University of Zagreb Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
dslavik@foi.unizg.hr

**Tim Juić**
Junior Developer
Softcon
Jakopovec, Marijana Lončara 20A, Croatia
tjuic@softcon.hr

**Abstract.** *Croatian accounting and insurance-brokerage companies handle large volumes of Croatian-language paperwork, including invoices, ledgers, contracts, commission lists, and many kinds of insurance policies, that need to be read, classified, and have key fields entered into a bookkeeping system. We present AppAI, a pipeline for processing documents that combines a vision-language model for grounded text extraction, a few-shot SetFit classifier trained on Croatian bookkeeping and insurance document classes labeled by project partners, and a self-hosted large language model for structured attribute extraction per document type. We position this design against recent work on OCR-free document understanding, few-shot text classification, and layout-aware extraction. On a single-partner held-out external validation set, classification reached 94.4% accuracy across 14 document types; over scored structured fields, extraction reached 87.80% field accuracy.*

**Keywords.** document classification, vision-language models, few-shot learning, large language models, information extraction

## 1 Introduction

Croatian accounting companies that also work as insurance brokers handle a steady stream of paperwork in Croatian: output invoice ledgers (cro. *izlazni računi*, *IRA*) and input invoice ledgers (cro. *ulazni računi*, *URA*), general invoices (cro. *fakture*), contracts (cro. *ugovori*), commission lists (cro. *provizijske liste*), and many kinds of insurance policies: compulsory motor vehicle liability (cro. *obvezno auto osiguranje*), CASCO (Casualty and Collision) insurance (cro. *kasko osiguranje*), property (cro. *osiguranje imovine*), liability (cro. *osiguranje odgovornosti*), travel (cro. *putno osiguranje*), life (cro. *životno osiguranje*), voluntary and supplementary health (cro. *dobrovoljno* and *dopunsko zdravstveno osiguranje*), employee collective accident (cro. *kolektivna nezgoda*), crop (cro. *osiguranje usjeva*), animal (cro. *osiguranje životinja*), and construction-risk (cro. *objekti u izgradnji*) insurance, plus vehicle registration certificates (cro. *prometne dozvole*). The workflow requires staff to identify the document type, then transcribe fields such as policyholder, OIB (cro. *Osobni identifikacijski broj*, the Croatian personal identification number), policy number, premium, covered risks, and dates into a bookkeeping or claims system. In the partner workflow motivating this work, much of this is still done by hand: slow, error-prone, and harder as volume grows.

AppAI is designed as a service for this workflow: given a document, it reads content in the Croatian language from digital PDFs or scans, classifies the document within a bookkeeping and insurance taxonomy, and extracts the attributes needed by downstream business systems. We built this as a hybrid pipeline, not one end-to-end model, because no single method is best at all three steps. Vision-language models are good at OCR-free, layout-aware reading. Few-shot classifiers work well when labels are scarce and the taxonomy keeps changing. Instruction-tuned LLMs can extract a different schema per document type without retraining. These documents also carry personal and financial data subject to Croatian and EU privacy requirements, so every model in the pipeline is self-hosted rather than called through a third-party API; Section 3 describes the anonymization measures applied before any document enters a training or evaluation corpus.

Two simpler alternatives were set aside: a single end-to-end vision-language model, which would need retraining or re-prompting whenever the taxonomy or a per-type schema changes; and per-class LLM prompting for classification (our earlier design, Section 4), which cost about five seconds per document. Its main contribution is the applied integration of these existing techniques into a working, self-hosted pipeline for Croatian accounting and insurance-brokerage document processing, evaluated on genuinely held-out, single-partner data.

The rest of the paper is organised as follows. Sec-

tion 2 reviews the literature behind each stage; Section 3 describes the architecture as currently implemented; Section 4 covers the classification model's training and evaluation; Section 5 evaluates the full pipeline on a held-out validation set; Section 6 discusses the results and directions for future work; Section 7 concludes.

# 2 Related Work

## 2.1 OCR-free and vision-language document reading

Classical document pipelines typically separate text recognition from downstream understanding: an OCR engine such as Tesseract (Smith, 2007) detects and recognizes text, and subsequent models consume the OCR output, often together with layout information. Transformer-based recognizers such as TrOCR (Li et al., 2023) improved recognition accuracy, but still function primarily as text-recognition components rather than end-to-end document-understanding systems.

Donut (Kim et al., 2022) showed a different approach: an OCR-free transformer that processes document images directly and reports competitive performance on visual-document-understanding benchmarks, avoiding a separate OCR stage and its error propagation. More general instruction-following multimodal systems such as LLaVA (Liu et al., 2023) and GPT-4V (OpenAI, 2023) established prompted vision-language interaction as a general capability.

By early 2025, VLMs with document understanding capabilities, such as Qwen2.5-VL (Bai, Chen, et al., 2025), were reporting OCR, document parsing, layout, and grounding capabilities closer to the requirements of AppAI's reading stage. Alongside these generative approaches, conventional document-conversion tools such as Docling (Livathinos et al., 2025) can parse programmatic text from digital PDFs and use OCR/layout/table models where needed to build a structured representation, a non-generative alternative to VLM-based conversion.

In the current implementation, AppAI follows the OCR-free direction, but uses a general-purpose vision-language model rather than one trained specifically for documents: a Qwen3-VL-8B-Instruct instance (Bai, Cai, et al., 2025), served through vLLM (Kwon et al., 2023), is prompted to return recognized text and bounding boxes for detected text regions in one pass. Docling is also available as a configurable alternative for use cases where a lightweight, non-generative conversion pipeline is preferred. We did not adopt layout-fused encoders like the LayoutLM family or Donut's own classifier head, mainly because AppAI's reading stage is decoupled from classification and has to support whatever prompt the extraction stage needs later, not a fixed label set.

## 2.2 Few-shot document-type classification

Once text is available, a small taxonomy that keeps changing is often better served by methods that need only a handful of labeled examples than by training a large classifier from scratch for every new type. SetFit (Tunstall et al., 2022) first fine-tunes a small sentence-embedding model on contrastive pairs drawn from a small labeled set, such as eight examples per class, then trains a lightweight classifier on top of the resulting embeddings. In the reported benchmark settings, a 110M-parameter SetFit model scores close to T-Few 3B, a prompt-based few-shot method roughly 27 times larger, while needing far less compute: SetFit is estimated to train and run about 19 times faster, and in practice took about 30 seconds to train on a single V100 GPU ($0.025 per run) versus roughly 700 seconds on an A100 for T-Few 3B ($0.70 per run), 28 times cheaper.

Multilingual embedding models such as BGE-M3 (Chen et al., 2024) provide multilingual retrieval within a single backbone, building on the broader sentence-embedding tradition established by Sentence-BERT (Reimers & Gurevych, 2019), while Croatian-specific encoders also exist, notably BERTić (Ljubešić & Lauc, 2021), an ELECTRA-style model trained on Bosnian, Croatian, Montenegrin, and Serbian web text. Contrastive training methods such as SimCSE (Gao et al., 2021) support the same representation-learning intuition behind SetFit: training on positive and negative pairs helps embeddings place similar texts close together and dissimilar texts farther apart in the embedding space.

A 2024 study re-evaluating document image classification with large language models (Scius-Bertrand et al., 2024) reports zero-shot GPT-4 at 61.8% accuracy on OCR text and GPT-4V at 69.9% on page images, on a low-label RVL-CDIP subset, with OCR-text embedding k-NN reaching up to 67.8% as labeled examples grow from 160 to 1,600. This comparison is useful for AppAI because it evaluates prompted classification against embedding-based baselines under low-label conditions, where repeated API calls, variance, and operational cost also matter.

This data efficiency will matter for AppAI: its Croatian bookkeeping and insurance taxonomy is expected to grow as partner companies and document types are added, and collecting thousands of labeled Croatian examples for every new class would be impractical, so a SetFit classifier was the natural fit. We fine-tune the multilingual BGE-M3 embedding model rather than BERTić, mainly because BGE-M3 was already available and tested on a much broader set of benchmarks; Whether a regional encoder such as BERTić would do better on this taxonomy is still untested. The corpus later grew to 2,427 documents (Section 4) as more classes were added; SetFit's few-shot efficiency is per class, independent of total corpus size.

### 2.3 Generative extraction with large language models

Classification identifies the document type; downstream processing also requires the document contents. DocLLM (Wang et al., 2024) shows layout-aware generative extraction can work without a full vision encoder, by feeding OCR text and spatial layout straight into the model. More generally, instruction-following models such as GPT-4 (OpenAI et al., 2023) helped popularize prompt-based workflows that return structured output, although reliable production use still requires validation around the model output. Dedicated Croatian generative models are also emerging, including HR-LLM, a completed pilot effort that built an early monolingual Croatian LLM still in beta, presented at NeTTT 2024 (Štefanec et al., 2024).

We assign extraction to a separate instruction-tuned LLM rather than extend the classifier, since the output schema differs by document type and changes more often than the type taxonomy does. We follow the simpler pattern used by GPT-4 rather than DocLLM's layout-aware approach: a self-hosted LLM (Qwen3-14B (Qwen Team, 2025), served through vLLM) receives a document-type prompt template plus the OCR text, and returns the requested attributes as JSON for downstream schema validation. We use a general multilingual model rather than a monolingual Croatian one because efforts such as HR-LLM have so far produced research pilots and resources still in beta rather than production alternatives for this pipeline.

### 2.4 Benchmark scope and evaluation caution

RVL-CDIP (Larson et al., 2023) remains one of the most widely used benchmarks in document classification, but it is known to contain label noise, ambiguous classes, and train/test overlap. A 2026 follow-up by the same group quantifies these issues further (12% label error, roughly 35% test-train duplication) and releases corrected RVL-CDIP variants, finding that fixing label errors improves out-of-distribution generalization by an average of 8.1% (Larson et al., 2026). The *Beyond Document Page Classification* study (Van Landeghem et al., 2024) goes further, arguing that real document classification is usually multi-page and spans far more varied domains and class distributions than a single-page, generic-class benchmark captures.

AppAI is not directly benchmarkable against RVL-CDIP, since its taxonomy is specific to Croatian accounting and insurance documents. These critiques mainly matter as a reason to evaluate against a held-out dataset labeled by project partners instead, rather than use a public leaderboard number to claim deployment readiness; this is the approach taken in Section 5.

## 3 System Architecture

AppAI runs as a set of independently deployable services behind a central backend that orchestrates the pipeline, so that each stage can be scaled, updated, or swapped for an alternative without redeploying the others. The document flow has three stages: reading, classification, and extraction, illustrated in Fig. 1.

**Reading.** When a document is ingested, the system routes it to Qwen3-VL-8B-Instruct, deployed in FP8 precision through vLLM, as the primary VLM-based reading engine. The model is prompted to return recognized text and bounding boxes for detected text regions. Docling is available as a configurable alternative for cases where a more lightweight, non-generative conversion pipeline is preferred. This stage outputs recognized text plus layout metadata, cached for later stages. The bounding boxes are currently kept mainly for caching and auditability; classification and extraction, below, use only the recognized text. Section 6 returns to this and discusses using the layout signal more directly for document types where text alone is not enough.

**Classification.** The recognized text goes to a separate classification component, which embeds it with a BGE-M3-based SentenceTransformer model (Chen et al., 2024) fine-tuned with SetFit, then runs a logistic regression head trained on a labeled corpus provided by project partners, covering Croatian bookkeeping classes (output and input invoice ledgers, general invoices, contracts, commission lists), vehicle registration certificates, and insurance policy classes (compulsory motor vehicle liability and CASCO insurance, property, liability, travel, health, life, employee collective accident, crop, animal, and construction-risk insurance). It returns a predicted document-type label and an uncalibrated classification score. Section 4 covers how this model was trained and evaluated.

**Extraction.** Once a document type is known, the system selects a prompt template listing the attributes to extract, such as policyholder name, OIB, policy number, covered risks, premium, and dates for an insurance policy, or invoice number, sender, and total amount for an invoice, and sends the recognized text plus the template to a self-hosted LLM (Qwen3-14B-AWQ (Qwen Team, 2025), served through vLLM, temperature 0.1, thinking mode off, 16,384-token context with 2,048 reserved for output). The model returns the attributes as structured output. Schema validation checks required fields and data types before the response goes out; semantic correctness is evaluated separately in Section 5. Each template states the required and optional fields, a few document-specific hints, and an instruction to return only structured output, parsed straight out of the model's response.

Keeping the three stages loosely coupled has practical value: the classification taxonomy and extraction schemas can change without retraining or redeploying

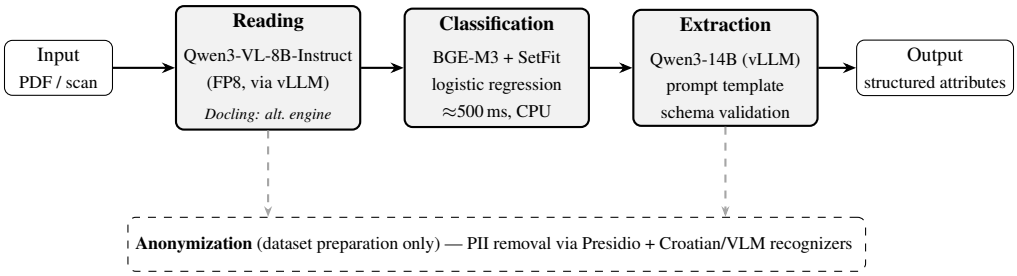



**Figure 1.** AppAI system architecture. The production pipeline (top) processes each document through three stages; the anonymization component (dashed) operates during dataset preparation for training and evaluation and is not part of the live inference path.

the reading stage, and the reading stage itself can be swapped (VLM versus Docling) without modifying the rest of the pipeline. A separate anonymization component applies Presidio-based PII detection (Data Privacy Stack, 2024), custom Croatian recognizers, and a VLM-based visual-region detector to remove identifying text and visual elements (e.g., signatures, stamps, handwriting) before documents enter training or evaluation corpora; it is not a stage of the live inference pipeline above.

# 4 Classification Model: Training and Evaluation

Classification is the best-tested stage so far: it has gone through two trained iterations as the labeled corpus grew. Classification in AppAI was not always done this way; an earlier version sent one LLM prompt per candidate class, 16 calls per document, taking around five seconds, with each prompt hand-curated around concrete cues (e.g. CASCO's theft, fire, vandalism, insured vehicle value, and deductible amount). The system was migrated to a SetFit classifier instead: after OCR/text extraction, one embedding pass plus a logistic regression lookup takes about 500 ms on a multicore CPU server (an AMD EPYC 7V13, 64 cores).

Training and evaluation for both SetFit iterations were done by our project team, using GPU infrastructure provided by project partners. Both iterations fine-tune BAAI/bge-m3 (568M parameters) with SetFit, using a CoSENT contrastive loss for the embedding stage and a scikit-learn logistic regression head for classification, on an 80/20 train/test split. These held-out splits, labeled by project partners, support comparing model iterations within the same corpus, under the current taxonomy and document collection. The split was randomized at the document level rather than stratified by issuer, template, time period, or near-duplicate document family, so internal-split accuracy can partly reflect template recognition; the external validation set in Section 5, evaluated on genuinely unseen documents, is the more decisive measure of generalization. The 114-document split used to pick the embedding backbone (Table 2) is likewise a model-selection set, not an independent test set.

Table 1 reports two engineering iterations. Because the OCR reader, corpus size, label taxonomy, and training settings changed together, the comparison reflects overall system progress rather than the isolated effect of one variable.

Before settling on this approach, three options were shortlisted. Classifying directly on frozen sentence embeddings needs only a small labeled set and no training step, but was expected to struggle distinguishing similar classes, for example two insurance types with overlapping wording. Fine-tuning a BERT-style classifier typically needs a larger labeled set and a full training run, with potentially higher accuracy at a much higher labeling cost. SetFit sits in between: it needs only a small labeled set, uses a fast contrastive fine-tuning step, and, according to the literature, adapts better than frozen embeddings while needing far less data than full classifier fine-tuning. SetFit was selected as the most practical trade-off for the available labeled data.

**First iteration.** Within SetFit, three embedding backbones were fine-tuned and compared on the same corpus of 612 documents across 17 classes: a small multilingual MiniLM, multilingual-E5-base, and BGE-M3. Table 2 summarizes the comparison. The MiniLM result also reflects its shorter input window, so the comparison is operational rather than a controlled comparison of backbone choice. A fourth candidate, multilingual-E5-base used only as a frozen embedding model, represented the embedding-only alternative above and reached 91% test accuracy on the same held-out split; it was dropped before the check on the full corpus, consistent with the expectation that frozen embeddings adapt less well to closely related classes within the same domain.

Four classes were underrepresented in this corpus: 1 document for the input invoice ledger, 6 for crop insurance, 12 for life insurance, and 14 for animal insurance, with roughly 50 documents per class otherwise. BGE-M3 had the most parameters and the largest training and inference footprint of the four, but also the highest accuracy, and was chosen as the system's backbone going forward.

**Table 1.** Training configuration and results for the two SetFit classifier iterations.

| Parameter | Iteration 1 | Iteration 2 |
|---|---|---|
| Base model | BAAI/bge-m3 (568M params) | BAAI/bge-m3 (568M params) |
| OCR method | Docling | Qwen3-VL-8B-Instruct (FP8) |
| Documents | 612 (17 classes) | 2,427 (15 classes) |
| Docs per class | $\sim$50 (4 classes $<$ 50) | $\sim$200 (4 classes $<$ 200) |
| Train / test split | 498 / 114 | 1,943 / 484 |
| Max input tokens | 512 | 768 |
| Batch size | 8 | 6 |
| Epochs | 3 | 2 |
| Contrastive iterations | 10 | 10 |
| Loss function | CoSENT | CoSENT |
| Test accuracy | **98.25%** (114 held-out docs) | **98.87%** (484 held-out docs) |
| Training time | $\sim$30 min | $\sim$3 h |
| Training GPU | RTX 4090 (24 GB) | RTX 4090 (24 GB) |

**Table 2.** Embedding backbones fine-tuned and compared for the SetFit classification stage, all trained on the same 612-document corpus.

| Model | Params | Max tok | Test acc. (114 held-out) |
|---|---|---|---|
| MiniLM | 118M | 128 | 93.86% |
| E5-base | 278M | 512 | 97.37% |
| BGE-M3 | 568M | 512 | **98.25%** |

The first iteration's few held-out errors were manually traced to incomplete OCR text rather than a genuine embedding or classifier failure; the same OCR-error-propagation pattern from Section 2.1 showed up again at a larger scale below. When we later evaluated this model against a larger pool of 2,436 documents, after rerunning OCR with Qwen3-VL-8B-Instruct at FP8 precision, accuracy dropped to 92.98%. This decrease was driven primarily by vehicle registration certificates, whose per-class accuracy fell to 49%. The first-iteration model had been trained on Docling text, where many vehicle registration certificates had little usable text; even after rerunning OCR at evaluation time, the class remained poorly learned. After retraining on VLM-read text, the second-iteration model reached 100% accuracy for vehicle registration certificates in the internal per-type breakdown, and none of the 6 external validation misclassifications (Section 5) involved this type either.

**Second iteration.** We reran text extraction on the whole corpus with the VLM-based reader, growing the processed pool to 2,436 documents; after excluding crop insurance (6 documents) and the input invoice ledger (3 documents) for the data-scarcity reasons given below, 2,427 documents across 15 trained classes remained for training, with about 200 documents per class otherwise, input truncated to 768 tokens, 2 training epochs, and the same CoSENT loss and contrastive-iteration count. This model reached 98.87% accuracy on its 484-document held-out closed-set test split. Both excluded classes fall below the 8-sample setting reported in the SetFit literature, and in a domain of closely similar classes a split this small produces too few contrastive pairs for stable learning and too few test documents for meaningful evaluation. They are therefore treated as gaps in data collection, not as validated classes in the reported accuracy across 15 classes. The health-insurance class in the corpus was also split during this growth, into voluntary and supplementary subtypes; both are trained separately here but reported jointly as "health insurance" in Section 5.

**Error analysis.** We went through the remaining misclassifications and low-confidence cases in the second iteration by hand and found that many reflected label ambiguity in the source corpus rather than a straightforward model failure. Of the 9 compulsory-liability documents assigned to the CASCO class in the full re-evaluated pool, at least 4 were manually confirmed to contain both coverages on a single policy despite having only one ground-truth label; the remaining cases were genuine misclassifications. A few documents are in German rather than Croatian, including a carrier's liability insurance policy that the model classified as CASCO insurance. Three documents labeled as commission lists are actually commission-posting notices. This matches the point raised in Section 2.4: taxonomies built on a single label per page often do not map cleanly onto real document collections (Van Landeghem et al., 2024). It also fits recent audits showing that error rates for individual categories reach double digits, even in benchmarks as established as RVL-CDIP (Larson et al., 2026). The clear next step is to define a labeling rule for policies that combine multiple types of coverage, rather than keep tuning the model.

Fig. 2 qualitatively illustrates the learned embedding space. The resulting neighborhoods are visually consistent with a well-separated CoSENT-trained representation; several points between clusters correspond to mislabeled or multi-coverage documents discussed above.

# 5 Pipeline Evaluation

We evaluated the pipeline on a held-out validation set of 107 documents covering 14 document types, provided by a single one of the project's partner companies and annotated with expected document type and

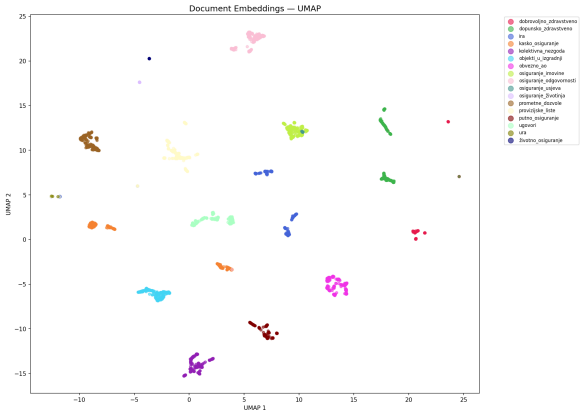



**Figure 2.** Exploratory UMAP projection produced with the second-iteration embedding model over the full labeled corpus, including classes excluded from training, colored by document class.

expected field values per document. Although drawn from one partner company, the validation set includes documents from multiple insurers and brokers, so it tests unseen issuer-template variation but not full cross-partner generalization. The validation set covers 14 document types, an observed subset of the full classifier taxonomy. The first SetFit iteration used 17 historical labels, while the second trained on 15 after excluding crop insurance and the input invoice ledger because of data scarcity. The validation set did not include every trained label, and none of its 107 documents appeared in any training corpus. Classification was evaluated on all 107 documents; extraction was evaluated over the scored structured fields shown in Table 3. Expected field values were annotated by partner-company staff against the source documents; we did not measure inter-annotator agreement or perform double annotation on this set, which we note as a direction for more rigorous validation.

**Classification.** The system classified 101 of 107 documents correctly (94.4%), after correcting label-naming differences between the validation set and the production taxonomy. This compares with 98.87% on the internal held-out split from Section 4.

**Extraction.** Fields from the expected insights annotations were compared against extracted values using four matching criteria, ordered from strictest to loosest: (1) exact string match after Unicode normalisation; (2) substring containment; (3) numeric equivalence within 2% relative tolerance, for monetary amounts; (4) Jaccard token overlap above 0.65, for company names and addresses. Strict identifiers (OIB, policy numbers) required exact match.

Because rows without scored extraction fields were excluded, this is a scored-field extraction result, not a full-pipeline score over all documents. Across 1,336 scored structured fields, 87.80% matched (1,173/1,336); free-text description fields scored 60.87% (28/46), a result we treat as exploratory given

**Table 3.** Per-type structured-field extraction accuracy on the held-out validation set, with document counts, scored field counts, and 95% Wilson confidence intervals (CI).

| Document type | n(doc) | n(field) | Struct. acc. | 95% CI |
|---|---|---|---|---|
| Output invoice ledger (IRA) | 10 | 40 | 100.00% | 91–100% |
| Vehicle registration certificates | 4 | 28 | 100.00% | 88–100% |
| Compulsory motor liability | 12 | 205 | 96.10% | 92–98% |
| Health insurance | 8 | 128 | 91.41% | 85–95% |
| Group accident insurance | 10 | 117 | 88.03% | 81–93% |
| Liability insurance | 8 | 112 | 86.61% | 79–92% |
| Travel insurance | 5 | 81 | 86.42% | 77–92% |
| Property insurance | 11 | 176 | 85.80% | 80–90% |
| Life insurance | 5 | 90 | 85.56% | 77–91% |
| CASCO insurance | 12 | 218 | 84.86% | 80–89% |
| Contracts | 4 | 16 | 81.25% | 57–93% |
| Construction-risk insurance | 12 | 125 | 76.00% | 68–83% |
| Animal insurance | 1 | 13 | n/a[a] | — |
| Input invoice ledger (URA) | 4 | 0 | n/a[b] | — |
| **Overall (scored rows)** | 101 | 1,336 | **87.80%** | 86–89% |

[a]single document: too sparse for a reliable per-type rate
[b]undertrained: 3 documents, below SetFit's 8-sample minimum (Section 4)

the small sample rather than a reliable estimate. Table 3 shows per-type results, together with the number of validation documents of each type. Every scored type rests on a small sample, from 4 documents (contracts) to 12 (CASCO, compulsory motor liability, construction-risk), so per-type percentages are indicative rather than precise; Table 3 reports a 95% confidence interval alongside each rate. The overall structured-field score excludes rows marked n/a. The lowest-scoring type is construction-risk insurance (76.00%), where several fields require multi-part values, such as multiple insured parties or itemized sums insured. These are harder to match exactly and mirror the broader difficulty with multi-part liability limits discussed after Table 3. The next lowest reflects a narrower, single-cause failure: two free-text fields for contracts (wrong contracting party, a regrouped commission-rate table).

At the field level, personal-data and simple-amount fields (name, address, date of birth, total amount, policy number) matched all evaluated instances in this validation subset; this provides evidence that the LLM handles clearly defined, single-value fields well in the current sample. The hardest field was multi-part liability limits (25%), where ground truth and LLM group the same sub-amounts under different party labels. Covered risks (59%) and insured-party name (68%) were also difficult, the former because different insurers use different vocabulary for the same risk; these counts are similarly modest.

The input invoice ledger produced no predictions because it was excluded from training (too few documents available). None of the n/a rows count toward the overall structured-field accuracy.

The overall score is computed over the 101 scored validation documents; the 5 n/a-row documents are excluded. Being field-level, it does not yet capture document-level exact match, abstentions, or the full

cost of multi-value fields; a fuller end-to-end evaluation is a priority for future work.

# 6 Discussion and Future Work

The classification stage reaches 94.4% on truly held-out external documents, and the extraction stage achieves 87.80% structured-field accuracy over scored extraction rows. We report classification and scored-field extraction accuracy separately, not a single end-to-end field score across unscored documents. The main remaining failures are classification-related: 4 of 6 external misclassifications are input-invoice-ledger documents excluded from training, and the other 2 are plausible combined-coverage confusions; vehicle registration certificates had no external-set misclassifications. A related next step is defining a labeling convention for documents that span more than one class. Beyond that, collecting more data for underrepresented classes is the next step.

A second direction follows from the bounding-box point in Section 3. Vehicle registration certificates are the dataset's most visually complex class (dense tables, security-pattern backgrounds, stamps, phone-camera scans), so their OCR quality is more capture-sensitive than that of the mostly text-based document types. This class caused the most errors in the first classification iteration (Section 4), not because it is hard to learn, but because most scans yielded little usable text from the text-only OCR pipeline. Rerunning OCR with a vision-language model addressed most of that failure mode: per-class classification accuracy on the same re-evaluated pool rose from 49% (98/200) to 100% (200/200). We have not directly evaluated the reading stage itself against ground-truth transcriptions; the comparison between the VLM-based and Docling-based readers here is indirect, made through its effect on downstream classification accuracy, and a direct reading-stage evaluation remains open work. The broader lesson is that for visually dominant, low-text document types, the more robust fix may not be better text extraction, but using the VLM's own visual and layout understanding directly. Classification could use the page image and its grounded regions, rather than treating every document type as a text problem first and falling back to the image only when that fails.

The same VLM-based visual grounding, currently used only to detect non-textual regions for anonymization (Section 3), could be reused in the main pipeline to answer questions text alone cannot: whether a contract or policy is actually signed and where, whether an official stamp or seal appears on the document, or whether a payment QR code or barcode is present on an invoice, stored as structured metadata alongside the extracted text.

# 7 Conclusion

We presented AppAI, a hybrid pipeline for reading, classifying, and extracting structured attributes from Croatian bookkeeping and insurance documents, combining a vision-language reading stage, a few-shot SetFit classifier, and a self-hosted LLM extraction stage. Its main contribution is the applied integration of these existing techniques into a working pipeline for a concrete, previously unaddressed Croatian accounting and insurance document-processing setting, evaluated on genuinely held-out data. The classification stage is the most thoroughly evaluated part of the system: across two trained iterations, it reached over 98% accuracy on Croatian-specific closed-set test splits covering 15 to 17 classes, and the error analysis traced many remaining mistakes to OCR quality and source-label noise rather than to the classifier itself. On a single-partner held-out validation set (Section 5), the classification stage reached 94.4% accuracy on truly novel documents and the extraction stage reached 87.80% structured-field accuracy over scored validation fields. The reading stage already produces a layout and visual-grounding signal that classification and extraction do not yet use; making fuller use of it remains an open direction for future work.

# Acknowledgments

This research was carried out under the project number IP.1.1.03.0138 within the Integrated Territorial Programme 2021–2027, co-financed by the Ministry of Regional Development and EU Funds.

# References

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., … Zhu, K. (2025). *Qwen3-VL Technical Report* (tech. rep.). Qwen Team. https://arxiv.org/abs/2511.21631

Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., … Lin, J. (2025). *Qwen2.5-VL Technical Report* (tech. rep.). Alibaba Group. https://arxiv.org/abs/2502.13923

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *Findings of the Association for Computational Linguistics: ACL 2024*, 2318–2335.

Data Privacy Stack. (2024). Presidio: Data Protection and De-identification SDK. https://presidio.dataprivacystack.org/

Gao, T., Yao, X., & Chen, D. (2021). SimCSE: Simple Contrastive Learning of Sentence Embeddings. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., & Park, S. (2022). OCR-Free Document Understanding Transformer. *Proceedings of the European Conference on Computer Vision (ECCV)*, 498–517.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., & Stoica, I. (2023). Efficient Memory Management for Large Language Model Serving with PagedAttention. *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*. https://doi.org/10.1145/3600006.3613165

Larson, S., Lim, G., & Leach, K. (2023). On Evaluation of Document Classification with RVL-CDIP. *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Larson, S., Nagy, A., Desai, S., Desai, C., Lima, N. C., Yuan, Y., Betala, S., Prajapati, K. K., Suleiman, J. T., Duwal, S., & Leach, K. (2026). Revising RVL-CDIP: Quantifying Errors and Test-Train Overlap. *Proceedings of the ACM Symposium on Document Engineering (DocEng)*. https://arxiv.org/abs/2606.31446

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). TrOCR: Transformer-Based Optical Character Recognition with Pre-Trained Models. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual Instruction Tuning. *Advances in Neural Information Processing Systems 36 (NeurIPS)*.

Livathinos, N., Auer, C., Lysak, M., Nassar, A., Dolfi, M., et al. (2025). Docling: An Efficient Open-Source Toolkit for AI-Driven Document Conversion. *AAAI-25 Workshop on Open-Source AI for Mainstream Use*.

Ljubešić, N., & Lauc, D. (2021). BERTić – The Transformer Language Model for Bosnian, Croatian, Montenegrin and Serbian. *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*.

OpenAI. (2023). GPT-4V(ision) System Card [System card]. https://openai.com/index/gpt-4v-system-card/

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). *GPT-4 Technical Report* (tech. rep.). OpenAI. https://arxiv.org/abs/2303.08774

Qwen Team. (2025). Qwen3-14B Model Card. https://huggingface.co/Qwen/Qwen3-14B

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. https://doi.org/10.18653/v1/D19-1410

Scius-Bertrand, A., Jungo, M., Vögtlin, L., Spat, J.-M., & Fischer, A. (2024). Zero-Shot Prompting and Few-Shot Fine-Tuning: Revisiting Document Image Classification Using Large Language Models. *Proceedings of the 27th International Conference on Pattern Recognition (ICPR)*.

Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*.

Štefanec, V., Farkaš, D., Thakkar, G., & Tadić, M. (2024). Building a Large Language Model for Croatian. *Proceedings of the International Conference New Trends in Translation and Technology (NeTTT)*.

Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., & Pereg, O. (2022). Efficient Few-Shot Learning Without Prompts. *Proceedings of the NeurIPS 2022 Workshop on Efficient Natural Language and Speech Processing (ENLSP)*.

Van Landeghem, J., Biswas, S., Blaschko, M., & Moens, M.-F. (2024). Beyond Document Page Classification: Design, Datasets, and Challenges. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2962–2972. https://openaccess.thecvf.com/content/WACV2024/html/Van_Landeghem_Beyond_Document_Page_Classification_Design_Datasets_and_Challenges_WACV_2024_paper.html

Wang, D., Raman, N., Sibue, M., Ma, Z., Babkin, P., Kaur, S., Pei, Y., Nourbakhsh, A., & Liu, X. (2024). DocLLM: A Layout-Aware Generative Language Model for Multimodal Document Understanding. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8529–8548.