International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Corpusio: A Practical Multimodal Extraction and Retrieval Ecosystem for Knowledge Discovery in Resource-Constrained Environments

Mahanthi Bharadwaj Phani Datta[1], Kamaleeswari Kamboji[2], Kancharala Subhaashini[3], Kella Kedhareesh[4], Raja N. Moorthy[5]

[1]AI & ML Engineer, Engineering Department, Kirusa Inc., Bengaluru, Karnataka, India
Corresponding Author Email: *mbharadwaj[at]kirusa.com*

[2]AI & ML Engineer, Engineering Department, Kirusa Inc., Bengaluru, Karnataka, India
Email: *kkamaleeswari[at]kirusa.com*

[3]AI & ML Engineer, Engineering Department, Kirusa Inc., Bengaluru, Karnataka, India
Email: *ksubhaashini[at]kirusa.com*

[4]AI & ML Engineer, Engineering Department, Kirusa Inc, Bengaluru, Karnataka, India
Email: *kkedhareesh[at]kirusa.com*

[5]COO, Engineering Operations, Kirusa Inc., Bengaluru, Karnataka, India
Email: *moorthy[at]kirusa.com*

This work was supported by Kirusa Inc. Internal Research and Development (IR&D) program

Abstract: ***As enterprise organizations accumulate vast amounts of heterogeneous unstructured data spanning PDFs, scanned contracts, and event photography traditional keyword retrieval systems fail to capture critical multimodal associations. This paper presents Corpusio, a multimodal extraction, indexing, and serving ecosystem designed for resource-constrained production environments. Rather than attempting semantic understanding via computationally prohibitive end-to-end large multimodal models, Corpusio employs an operationally conservative pipeline. Specifically, the system utilizes a deterministic, two-pass person-image linking strategy that combines layout-first card-based proximity grouping with a formal linear assignment solver to prevent cross-identity leakage. Visual artifacts are indexed via perceptual hashing to enable efficient deduplication and anchor-based feedback loops. Furthermore, the ecosystem mediates downstream Retrieval Augmented Generation (RAG) by surfacing stable evidence locators wrapped in strict Role-Based Access Control (RBAC) masks. On a log-verified production corpus of 46 diverse documents, the pipeline successfully executed with a mean latency of 208.54 s/doc and sustained a 3.72 GB peak allocated GPU footprint on a commodity NVIDIA T4 (16 GB). These results demonstrate that a deterministic, multi-stage retrieval architecture can deliver high-precision, auditable multimodal discovery under bounded compute while strictly enforcing operational bounds and data confidentiality.***

Keywords: access control, face verification, hybrid retrieval, image-text binding, layout analysis, multi- modal document understanding, privacy-preserving machine learning, resource-constrained deployment, retrieval-augmented generation, semantic embeddings

1. Introduction

1) The Dark Data Problem

Companies possess large volumes of heterogeneous unstructured data: newsletters, technical diagrams, event photography, scanned contracts, and multilingual reports. While key- word search is effective for file names, it often fails to capture multimodal associations. For instance, a search for ‘‘server’’ may identify relevant tokens but miss that a nearby embedded figure is a ‘‘server rack diagram’’ linked to an equipment module described elsewhere.

2) Challenges on the Ground

Deploying document search systems in diverse global markets presents significant technical obstacles:

a) **Out-of-distribution entities:** Models pre-trained on Western corpora (e.g., CoNLL-2003) exhibit reduced performance on South Asian and African naming conventions. Benchmarking with spaCy’s en_core_web_trf [1] can under-recall multi-part names and misclassify ambiguous name tokens.

b) **Heterogeneous layouts:** Enterprise environments in- volve diverse formats, including multi-column newsletters and documents with highly variable layouts. While end-to- end models such as LayoutLMv3 are effective, they often require extensive fine-tuning and introduce computational overhead that is difficult to justify under strict operational constraints.

c) **Resource-intensive OCR:** Legacy documents often exist only as scanned images, making OCR a primary bottleneck for latency and accuracy. Tesseract [2], [3] remains a practical baseline; transformer OCR models (e.g., TrOCR [4]) can improve accuracy but may exceed typical cost constraints.

d) **Hardware constraints:** Cost-sensitive deployments typically rely on commodity hardware (e.g., 16 GB NVIDIA T4) rather than high-end clusters. The system must remain operational within these limits to avoid memory exhaustion and instability. This challenge is amplified by the massive parameter counts of modern transformer-based architectures [5]–[7].

e) **Mobile-first workflows:** In many regions, field staff

rely on mobile interfaces. A practical solution must provide mobile-accessible retrieval without requiring heavy desktop tooling.

f) **Data confidentiality:** Enterprises are often reluctant to transmit confidential documents to external cloud-based APIs. The system must curate evidence, redact sensitive con- tent, and provide verifiable citations to reduce ungrounded answers from downstream Large Language Models (LLMs) used in retrieval-augmented generation [8], [9].

3) How Corpusio Fits Together

We focus on solving multimodal extraction, indexing, and retrieval-serving for enterprise corpora (PDF, DOCX, XLS, and image formats). We do not claim end-to-end semantic ''understanding'' or state-of-the-art results on public multimodal benchmarks; rather, we prioritize an auditable, privacy-preserving pipeline for production environments.

To address the challenges above, Corpusio is built with a modular, operationally conservative design:

a) **Operational memory control:** Models are managed through a device-placement policy (GPU-to-CPU moves) and explicit garbage collection to keep allocated GPU memory bounded under commodity VRAM budgets.

b) **Identity-based versioning:** Extracted artifacts are indexed using stable identity stems (file hashes or folder IDs) rather than ephemeral titles, supporting idempotent skip/resume logic based on persisted artifacts and checkpoints.

c) **Perceptual Hashing (pHash):** Visual artifacts are indexed by pHash (Hamming distance threshold ≤ 6 for global deduplication), enabling efficient deduplication and serving as the stable key for feedback-based anchor registration.

d) **Deterministic person-image linking:** Corpusio performs a two-pass strategy: knowledge-base anchor registration followed by a globally optimal 1-to-1 linear assignment (SciPy LAP solver) driven by layout-aware card-based proximity.

e) **Ensemble Named Entity Recognition (NER):** The system integrates spaCy with strict name hygiene filters and systemic invariants to improve recall for diverse naming conventions.

f) **Hybrid search:** The system embeds text (BGE-M3) and images/pages (CLIP-family) and employs dense (FAISS) and optional sparse (BM25) retrieval.

g) **Access-controlled evidence mediation:** Raw files are transformed into redacted evidence snippets with stable locators and role-based masks.

2. Mathematical Foundations

1) Problem Setup

Definition 1 (Multimodal Document)**.** *A document* D *is a tuple* (T, I, M) *where:*

- T = $\{t_1, \ldots, t_n\}$ *is the set of extracted text blocks, each with bounding box coordinates* (x_0, y_0, x_1, y_1) *(modeling and indexing foundations follow [10], [11]).*
- I = $\{v_1, \ldots, v_m\}$ *is the set of extracted image regions, each with page identifier and (optionally) a detected person count and face encoding.*
- M: T $\cup$ I $\rightarrow \{0, 1\}^k$ *maps elements to k-bit access-control vectors and locators.*

Definition 2 (Mediated Evidence Set)**.** *Given query q and user u, the retrieval subsystem returns an evidence set*

$$E(q, u) = \{(c_i, \pi_i, \sigma_i)\}_{i=1}^{K}, \quad (1)$$

where c_i *is a redacted content chunk,* π_i *is a stable pointer (file id, page id, bounding box, or image path), and* σ_i *is a security label used by Role-Based Access Control (RBAC) [12].*

2) Two-Pass Person–Image Linking (Implementation-Aligned)

The backend implements person–image linking using two deterministic layers that interact: (i) a *layout-first* 1-to-1 proximity assignment (Linear Assignment Problem, or LAP solver) that produces a *locked* name for a portrait when the page resembles profile cards, and (ii) a *two-pass face anchoring* strategy that registers high-confidence anchors from feedback and optionally verifies a candidate name against its own anchors.

Definition 3 (Document Face Registry (Anchors))**.** *Let* A *denote a per-document face registry storing tuples* (n, e, h) *where n is a canonical person name,* $\mathrm{e} \in \mathrm{R}^{128}$ *is a face_recognition encoding, and h is an optional perceptual hash of the face/image crop used as a fast prefilter.*

Definition 4 (Anchor Registration via Feedback Knowledge Base)**.** *Let* $h(v)$ *be the perceptual hash of image region v. A feedback knowledge base stores corrections keyed by hash (exact match in the production path). If the knowledge base returns a corrected classification* $\kappa(h(v))$ *with* $\kappa(h(v))$.type = *person and* $\kappa(h(v))$.mapping = $n \neq$ *Unknown, and a face encoding* e_v *is obtainable, then the anchor is registered:*

$$\mathrm{A} \leftarrow \mathrm{A} \cup \{(n, \mathrm{e}_v, h(v))\}. \quad (2)$$

This constitutes ***Pass 1*** *of the backend two-pass face anchoring strategy.*

Definition 5 (Dynamic Layout Metrics)**.** *Given a set of portrait-like image regions* I_p *and candidate name headers* H *on a page with width W and height H, the backend derives adaptive scale parameters:*

$$\mu_{\text{img}} = \max(100, \operatorname{avg}\{h(v) : v \in \mathrm{I}_p,\ 50 < h(v) < 0.5H\}), \quad (3)$$

$$\mu_{\text{line}} = \operatorname{avg}\{\ell(t) : t \in \mathrm{H},\ 5 < \ell(t) < 60\}, \quad (4)$$

and then defines thresholds

$$\Delta_y = 2.5\mu_{\text{line}}, \quad \Delta_x = 0.15W, \quad J_y = 1.2\mu_{\text{img}}, \quad \lambda = 1.5\mu_{\text{img}}. \quad (5)$$

These parameters are computed per page to avoid brittle fixed global thresholds.

Definition 6 (Layout Proximity Score)**.** *Let* $c(v) = (x_v, y_v)$ *and* $c(t) = (x_t, y_t)$ *denote centers of an image region and*

a header block. Define

$$d(v,t) = \|c(v) - c(t)\|_2, \quad d_x = |x_v - x_t|, \quad d_y = |y_v - y_t|. \tag{6}$$

The backend uses a unified spatial score (used consistently for both profile-cards and global matching):

$$s_{\text{spatial}}(v,t) = a \exp\left(-\frac{d(v,t)}{b}\right) + \Delta_{\text{row}} + \Delta_{\text{align}} - \Delta_{\text{jump}} \tag{7}$$

where a, b, and the Δ constants are derived from page-dependent empirical box distributions. This yields a smooth distance decay, with deterministic boosts for row alignment and local horizontal alignment, and a penalty for large vertical jumps.

Definition 7 (Card-Based Proximity Grouping)**.** *Instead of global grids, the backend ensembles layout cards. Let candidate headers be sorted by their center y coordinate. For each header, the implementation defines a local proximity zone driven by card-based boundaries. A Phase 7 guardrail explicitly identifies and excludes* banned headers *(bBox-detected headers/footers) before assignment. Proximity weighting is combined with card ownership boosts to ensure stable 1-to-1 affinity.*

Definition 8 (Header Hygiene Penalty (Generic/Non-Name Filtering))**.** *Let t be a candidate header text. The backend applies strong penalties to text blocks whose first token is detected as a common noun or designation (e.g., ''Experience'', ''Education'', roles/titles), and additionally penalizes known camera/copyright watermark patterns. Formally, this can be represented as an additive penalty term* $p(t) \leq 0$ *that is large in magnitude for generic headers.*

Proposition1*(Linear Assignment for 1- to- 1 Name Locking). Let* $\mathcal{I}_p = \{v_i\}_{i=1}^{N}$ *be person-like image regions on a profile like page and* $\mathcal{H} = \{n_j\}_{j=1}^{M}$ *the set of detected candidate names. Construct a score matrix* $\mathbf{P} \in \mathbb{R}^{N \times M}$ *with entries*

$$P_{ij} = s_{\text{spatial}}(v_i, n_j) + p(n_j) + a(v_i, n_j), \tag{8}$$

where a (*v*, *n*) *encodes adjustments: (i) type affinity, (ii) y-overlap bonus, and (iii) portrait priority boost. The backend resolves the mapping by solving the maximization objective*

$$\max_{\pi} \sum_{i=1}^{N} P_{i,\pi(i)} \quad \text{s.t. } \pi \text{ is injective}, \tag{9}$$

via a LAP solver (SciPy linear_sum_assignment [13]) with a mononym safety gate: assignments are rejected if the header is a mononym and the score is below a floor of 10.0. A nearest name-block fallback ensures 1-to-1 enforcement for residual Unknown portraits.

Proposition 2 (Face Verification (Candidate-Only Safety Rule))**.** *For a candidate name n, let* A(*n*) *be anchors for n in the document registry, where face encodings* $\mathbf{e} \in \mathbb{R}^{128}$. *Given image region v with encoding* $\mathbf{e}_v$, *the match decision is defined as:*

$$\min_{(n,\mathbf{e}_n)\in\mathcal{A}(n)} \|\mathbf{e}_v - \mathbf{e}_n\|_2 \leq \tau \tag{10}$$

where the system accepts the match if the condition holds for $\tau = 0.55$. *(Operationally, we define a similarity score* $s_f(v,n) = -\|\mathbf{e}_v - \mathbf{e}_n\|_2$ *and threshold it separately). Critically, verification is performed* only *against anchors of the candidate n; if n is not present in the registry, the system returns ''no match.'' This prevents accidental reassignment due to matching a different person's anchor.*

Definition 9 (Operational ''Lock'' Rule)**.** *When the layout solver assigns a non-unknown name* $\hat{n}_{\text{layout}}(v)$ *to an image region v and YOLO indicates at least one person in v, the deterministic image classifier returns*

$$\hat{n}(v) = \hat{n}_{\text{layout}}(v) \tag{11}$$

with high confidence, preventing downstream scoring stages from swapping names. This is an explicit production guardrail against portrait-name permutation errors.

3) Operational Memory Management

Corpusio enforces an operational guarantee regarding GPU utilization through a device-placement policy and explicit memory hygiene. The implementation reports allocator telemetry rather than theoretical parameter-only estimates, since allocator behavior (caching and fragmentation) drives practical OOM risk.

Proposition 3 (Allocator Telemetry)**.** *The system monitors two primary metrics from the framework allocator (e.g., PyTorch/CUDA [14]):*

- ***Allocated Peak*** *(*M_a*)**:** Defined via torch.cuda. memory_allocated(), the maximum memory utilized by active tensor objects. This metric is the primary driver for Out-Of-Memory (OOM) events.*
- ***Reserved Peak*** *(*M_r *)**:** Defined via torch.cuda. memory_reserved(), the total memory managed by the allocator. The caching allocator guarantees the basic inequality* $M_r \geq M_a$.

To remain within the 16 GB limits of an NVIDIA T4, the implementation performs ***device-placement offloading****: models not currently in the active batch are moved to CPU, followed by explicit gc.collect() and torch.cuda.empty_cache() calls. The production log reports* $M_a = 3.72$ *GB and* $M_r = 3.85$ *GB under this regime.*

4) Complexity Analysis

The overall pipeline runtime bound is roughly $O(n \cdot p \cdot C)$ for n documents, p pages per document, and C average candidate combinations per page. Solving the dense LAP assignment per page is bounded by $O(k^3)$ where $k = \min(N, M)$ candidates. Vector storage introduces an overhead scaling $O(n_t\, d_t + n_i d_i)$ for n_t text chunks of dimension d_t and n_i image regions of dimension d_i.

5) Hybrid Retrieval Analysis

Proposition 4 (Hybrid Retrieval Recall Lower Bound)**.** *For linear fusion of BM25 and dense retrieval with* $\alpha \in [0, 1]$, *let* R_{BM25} *and* R_{dense} *denote ranked lists. The probability that at least one of the two retrieves a relevant item is*

$$P(\text{rel} \in R_{BM25} \cup R_{dense}) \geq \max(P(\text{rel} \in R_{BM25}), P(\text{rel} \in R_{dense})). \quad (12)$$

When the two rankings are weakly correlated, fusion improves recall by increasing ranking diversity.

6) Time and Space Complexity

Proposition 5 (Time Complexity)**.** *The extraction pipeline has time complexity:*

$$T(n) = O(n \cdot p \cdot (t_{text} + t_{vision} + t_{link} + t_{embed})), \quad (13)$$

where n is the document count and p is the average pages per document. The parameters t_{text} , t_{vision}, t_{link} , and t_{embed} represent the per-page time for OCR, visual detection/embedding, person linking, and text embedding persistence, respectively.

For the log-verified run with $n = 46$ and mean 208.54 s per document,

$$T_{\text{total}} \approx 46 \times 208.54 \approx 9593 \text{ s} \approx 160 \text{ minutes}. \quad (14)$$

Proposition 6 (Space Complexity)**.** *Let n_t be the number of text chunks and n_i the number of image regions. With float32 vectors, approximate storage is*

$$S_{total} \approx 4(n_t d_t + n_i d_i), \quad (15)$$

where d_t is the text embedding dimension (BGE-M3: 1024) and d_i is the vision embedding dimension (vision encoder configured in deployment; the production log reflects the effective d_i used to build indices).

3. Related Work

Document processing historically relied on templates and brittle rules. Modern document AI integrates layout and language via pretraining. LayoutLM [15], LayoutLMv2 [16], and DocFormer [17] demonstrate multimodal fusion for visually-rich documents. For benchmark settings such as Pub- LayNet [18], DocBank [19], RVL-CDIP [20], or FUNSD [21], large-scale multimodal models and scaling studies [22], [23] are competitive. Additional datasets like DocVQA [24] specifically target question-answering on images; however, these are considered representative approaches rather than deployed dependencies in our constrained environment.

1) Baselines And Practical Detectors

Corpusio uses YOLOv8 [25] for region cues. Earlier YOLO models [26], [27] and later proposals (e.g., YOLOv10 [28]) illustrate the speed–accuracy progression for single-stage detection. In resource-constrained enterprise deployments, mature inference tooling and robust failure handling are often prioritized.

2) Retrieval-Augmented Generation (Rag)

RAG [8], [9] grounds generation on retrieved evidence. Retrieval-augmented pretraining and open-domain QA work [29]- established the retrieval component as a first-class module. Techniques for query rewriting [32] and dual instruction tuning [33] further refine the interaction between retrieval and generation. Human-assisted QA and multitask retrieval studies [34], [35] highlight the importance of high-quality evidence. Language models themselves have been explored as internal knowledge bases [36]. Corpusio adopts evidence mediation and RBAC filtering as a practical control layer for enterprise usage.

3) Vision and Text Embeddings

CLIP [37] introduced large-scale vision–language embeddings by training a symmetric-loss objective on image-text pairs. For semantic text retrieval, Sentence-BERT [38] and later multilingual embeddings such as BGE-M3 [39], [40] enable efficient dense retrieval. Corpusio uses CLIP-family encoders (e.g., ViT-L/14 as a common deployment default) for image/page embeddings and BGE-M3 for text.

4) Vector Indexing

FAISS [41] is a standard library for similarity search; alternatives include HNSW [42], PQ [43], and pgvector [44]. Late interaction models like ColBERT [45], [46] offer alternatives to simple dot-product search. Corpusio stores vectors as NPY and builds in-memory indices during serving to avoid operational complexity.

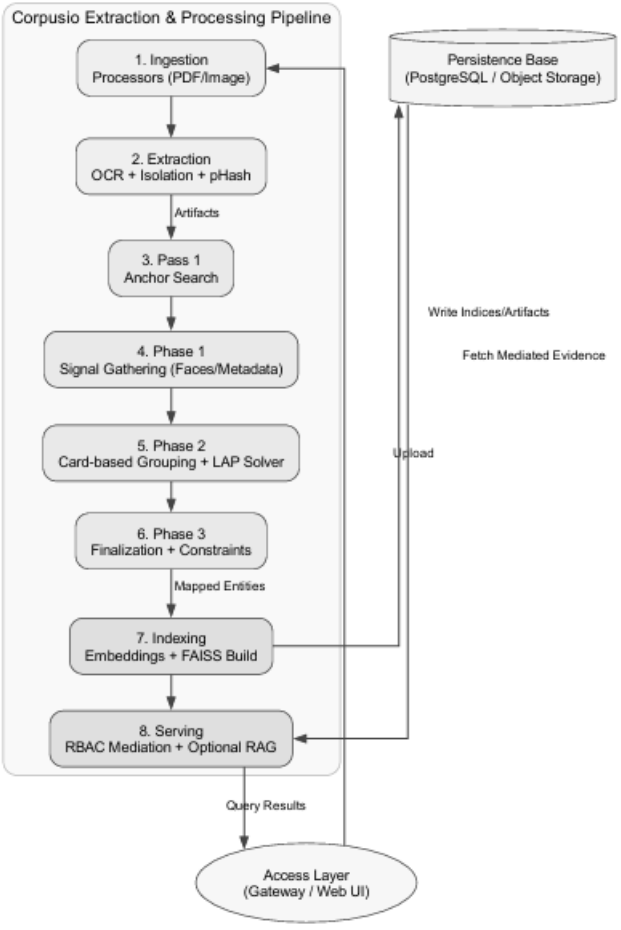

Figure 1: Corpusio system architecture showing access layer, online RAG serving, offline extraction/indexing, and storage dependencies.

5) Non-Western Name Extraction

Global South name distributions often challenge standard NER models. Corpusio uses spaCy [1] with strict name hygiene filters and optional multilingual fallbacks (IndicNER [47], IndicNLP [48], Stanza [49], Flair [50]), consistent with multilingual transfer observations [51], [52].

4. System Architecture

Corpusio uses a modular monolith architecture (Figure 1) to simplify deployment while enabling independent component upgrades.

1) System Overview (Stage List)

The system orchestrates a multi-stage auditable pipeline, verified via execution logs (summarized in Table 1).

Table 1: Corpusio 8-Stage Extraction and Retrieval Pipeline

Stage	Operations
1. Ingestion	Connectors fetch files → Processors (PDF/Office/Image)
2. Extraction	OCR and text extraction → Image isolation → pHash deduplication
3. Pass 1	Anchor Search: Knowledge-base query matching pHash
4. Phase 1	Signal Gathering: Face detection, spatial metadata
5. Phase 2	Assignment: Card-based proximity grouping and global page LAP
6. Phase 3	Finalization: Type classification and structural constraints
7. Indexing	Representation embedding → Persistence → FAISS index build
8. Serving	Retrieval → Evidence assembly → RBAC redaction → Optional RAG

2) Memory Orchestration (Implementation)

Rather than maintaining all models concurrently on GPU, Corpusio uses explicit device moves and cleanup. A representative policy is:

```
# after a batch:
model.to("cpu")
del batch_tensors
gc.collect()
torch.cuda.empty_cache()
```

Listing 1: GPU Control Policy (Simplified)

This device-placement strategy reduces fragmentation and caps allocated GPU usage under commodity VRAM budgets. Specifically, text models and large vision backbones are moved to CPU when idle to maximize headroom for the active pipeline stage.

3) Systemic Invariants and Guardrails

To ensure pipeline integrity, the system enforces several **invariants**: (i) entity rosters are validated for person-containment before being passed to the spatial solver, (ii) face encoding is gated by single-person YOLO detections to prevent identity bleeding in group photos, and (iii) image mapping results are checked against a score-margin threshold (abstaining if ambiguous). These guardrails are reported in production logs as warning-level telemetry when violated.

4) Knowledge Expansion (Optional Enrichment)

While the core extraction focuses on local document context, the system includes optional modules for **Knowledge Graph** (GraphDB) construction and **Wikidata** linking. These components serve as optional enrichments during the gateway serving phase and are not included in the primary extraction benchmarks.

5. Methodology I: Document Processing and Analysis

1) Text And Metadata Extraction

Corpusio uses PyMuPDF [53] for vector PDF processing and conditional Tesseract OCR [2], [3] for scanned pages. For Office documents, embedded images are extracted and persisted for visual indexing. Perceptual hashing is computed for each extracted image to enable deduplication and to attach feedback across runs.

2) Yolo-Based Person Region Cues

YOLOv8 [25] provides person-region detection that gates downstream face-based linking. In person-linking mode, face encoding is computed only when detection indicates a single-person region, reducing spurious face signals from group photos.

3) Two-Pass Person Linking (Operational Logic)

The implementation follows two coupled strategies:

Layout-first 1-to-1 mapping: On pages detected as profile-like (portrait presence + bio markers + name-like headers), the system performs card-based grouping and solves a 1-to-1 linear assignment (LAP solver) using the spatial score of Eq. (7) with deterministic hygiene penalties (generic headers, watermark captions) and type-affinity adjustments. The result is a *locked* layout mapping that prevents name swaps.

```
Input: query q, user roles R
1) intent = IntentModel(q)
2) entities = Hybrid_NER(q)  # Hybrid Named Entity Recognition (NER)
3) q' = rewrite(q, entities, feedback_db)
4) results = hybrid_search(q', k) or image_search(q', k)
5) results = filter_byRBAC(results, R)
6) results = rerank(results)  # optional
7) answer = LLM(context=results)  # optional
8) attach citations with page/box locators
```

Listing 2: Retrieval-Augmented Generation (RAG) Query Processing (Simplified)

Face anchoring (two-pass): For each image region v, compute its perceptual hash $h(v)$. If the knowledge base contains an exact correction indicating a person mapping n and a face encoding e_v is obtainable, register $(n, e_v, h(v))$ in the document-level face registry (Pass 1). In Pass 2, face verification can confirm (or reject) a candidate name by comparing only against that candidate's anchors with tolerance $\tau = 0.55$.

4) Structural Signal Ensemble (Logo and Architecture)

Non-person images (logos, diagrams, charts) are mapped using a structural signal ensemble. This includes **edge density** and **line counts** for architecture, **numeric density** for charts, and **entity affinity** (matching OCR tokens against ORG/PRODUCT entities) for corporate logos. The system abstains from classification when the margin between top candidates is insufficient, ensuring high precision for as- signed metadata.

6. Methodology II: Retrieval and Inference

1) Building The Indices

Corpusio builds three retrieval targets:

Text: encode text using Beijing Academy of Artificial Intelligence Big Gradient Embedding (BAAI/BGE-M3) [39] (1024-dim) for dense retrieval and maintain an optional BM25 [11] hook for exact keyword matches. Linear fusion uses

$$score_{hybrid} = \alpha \cdot score_{dense} + (1 - \alpha) \cdot score_{BM25}. \quad (16)$$

Images and Pages: use Contrastive Language-Image Pre-training (CLIP-family) embeddings [37] for image regions and full-page snapshots, retrieved via cosine similarity in Facebook AI Similarity Search (FAISS) [41].

2) Retrieval and Query Pipeline

3) Enterprise Gateway and Mobile Deployment

Accessibility is provided via a stateless API gateway supporting mobile clients (e.g., WhatsApp). The gateway manages session persistence and delivers media artifacts through time- limited (TTL) signed URLs. This architecture ensures that sensitive evidence remains stored in access-controlled environments while providing low-latency feedback to mobile users. Image-entity associations are robustly handled by the card-based solver, which enforces layout-aware constraints to maintain association integrity across heterogeneous document styles.

4) Graphrag Layer: Relationship-Aware Retrieval for Hallucination Mitigation

A key limitation of purely similarity-based retrieval is that queries involving *relationships* between entities- e.g., ''What is the role of Person A in Project B?''- tend to surface topically relevant chunks about each entity independently, but not chunks that explicitly encode the relationship between them. This gap causes downstream LLMs to fabricate connections, a primary source of hallucination in enterprise RAG deployments.

To address this, Corpusio optionally augments the retrieval pipeline with a **GraphRAG layer** backed by a Neo4j knowledge graph. During ingestion, the extraction pipeline populates a graph database where nodes represent key entities (persons, organizations, roles, products, and other important named entities) and directed edges encode semantic relation- ships between them derived from co-occurrence and dependency parsing over the extracted corpus. Each embedding chunk is annotated with a concept_id metadata field linking it back to its corresponding graph node(s).

At query time, before the hybrid dense–sparse retrieval step, the system executes a **multi-hop graph traversal** over the knowledge graph. The traversal depth and direction are guided by the query's detected entities and intent- expanding across first- and second-degree (and optionally deeper) relationships to capture indirect connections that flat retrieval would miss. The returned concept_id set from this traversal is used as an additional filter and boost signal over the FAISS index, ensuring that chunks encoding *relationships*- not merely entity mentions- are surfaced alongside topically similar chunks. This produces a richer, relationship-grounded evidence set that significantly reduces LLM hallucination on relational queries.

7. Data Confidentiality and Access Control

1) Confidentiality Objectives

The objective is to enable LLM-assisted discovery without exposing sensitive source documents to external services. Corpusio therefore mediates evidence and enforces RBAC before any generation step.

2) Filtering By Role

Let $R(u)$ be the user's role and allow $(\sigma, R(u)) \in \{0, 1\}$ be the RBAC decision. Retrieval returns:

$$E(q,u) = \{e_i \in \text{Retrieve}(q) \ : \ \text{allow}(\sigma_i, \mathcal{R}(u)) = 1\}. \quad (17)$$

3) Redaction

Evidence is redacted before being passed downstream:

$$\tilde{E}(q,u) = \{\text{redact}(e_i) \ : \ e_i \in E(q,u)\}. \quad (18)$$

Theorem 1 (Mediated Exposure Bound)**.** *Assume the generator only reads $\tilde{E}(q, u)$, redaction deletes forbidden text, and RBAC is correctly applied. Then any forbidden content not present in $\tilde{E}(q, u)$ is not exposed to the generator through the Corpusio interface.*

Sketch. By construction, the evidence bucket is the only path from storage to the generator. If forbidden strings are excluded by RBAC and deleted by redaction, they do not appear in the context provided to the generator.

8. System Robustness and Fault Tolerance

1) Failsafes

Out-Of-Memory (OOM) Fallbacks: When GPU memory is low, the pipeline triggers deep cleanup and can fall back to CPU for specific embedding steps.
Resume Function: Files are hashed and stored in a manifest to support resuming long runs.

2) Learning From Mistakes

Human corrections are persisted (e.g., image hash to corrected label/name). These corrections are used as pass-1 anchors in subsequent extraction runs.

3) System Configuration and Hyperparameters

Experiments were conducted on a commodity server node:

- **Hardware:** The production deployment utilizes an NVIDIA T4 Graphics Processing Unit (GPU) (16 GB Video Random Access Memory (VRAM)). The runtime environment typically includes 64 GB System RAM and multi-core Xeon-class CPUs.
- **Detection and Batching:** YOLOv8 defaults to batch size 4 for stability, while embedding batches are dynamic and reduced upon memory pressure signal.
- **Thresholds:** Face verification uses a tolerance threshold $\tau = 0.55$ for normalized distances.
- **Embeddings:** Text embeddings are 1024-dimensional (BGE-M3). Vision encoding uses the CLIP-family (e.g., ViT-L/14, 768-dim).

9. Experimental Results

1) Evaluation Protocol

Because the production corpus is proprietary, we report aggregated metrics and release evaluation scripts and annotation templates; raw documents are not redistributed.

- **Labeled Set:** We constructed an evaluation subset consisting of 112 candidate images across 10 distinct subjects. Ground truth was established by two internal annotators with a final consensus pass for ambiguous cases.
- **Metric Definition:** We report Recall@1 for person–image mapping. Precision is calculated as the ratio of correct links to total links assigned by the solver.
- **Metric Disclaimer:** The accuracy metrics reported below were obtained via manual labeling of an internal proprietary subset. Evaluation artifacts (annotations and gold-standard scripts) are not bundled with the open distribution for confidentiality reasons.
- **Operational Failure:** We define a fatal failure as an in-stance where the extractor exits with a non-zero status code or a per-file run terminates without generating required embedding artifacts.

2) Evaluation Set

We prioritize heterogeneous enterprise documents:

- **Legal and HR:** scanned agreements and policy documents.
- **Commercial:** pricing matrices and complex presentations.
- **Technical:** infrastructure diagrams and engineering documentation.

3) Log-Verified Runs (Two Executions)

The provided pipeline.logcontains two executions:

Run A (warm start / index continuation): the pipeline resumed from previously persisted embeddings and reported counters:

$$(\text{text}, \text{image}, \text{page\_image}) = (449, 48, 173) \quad (19)$$

This run reports files=0in the summary, consistent with an index continuation step rather than fresh ingestion.

Run B (full extraction): the pipeline processed $n = 46$ docs and produced counters:

$$(\text{text}, \text{image}, \text{page\_image}) = (729, 259, 472) \quad (20)$$

with 46/46 files successfully completed, a total input size of 64.17 MB, and a mean runtime of 208.54 s/doc. GPU allocator telemetry recorded a peak allocated memory of 3.72 GB and a peak reserved memory of 3.85 GB on an NVIDIA T4 (16 GB). Note that these are dynamic telemetry checkpoints, not static theoretical parameter footprints.

4) Person–Image Linking Accuracy (Internal Benchmarks)

On the labeled subset, qualitative testing indicates that the system achieves robust recall for person–image mapping.

Table 2: Operational metrics reproduced from pipeline.log. Run A demonstrates index warm-start/reuse of persisted embeddings

Metric	Run	Value
Files processed	Run A (Warm-start)	0
Counters (text, image, page)	Run A	(449, 48, 173)
Files processed	Run B (Full Extraction)	46
Succeeded / Failed	Run B	46 / 0
Input size	Run B	64.17 MB
Mean runtime	Run B	208.54 s/doc
Peak GPU allocated (Ma)	Run B	3.72 GB
Peak GPU reserved (Mr)	Run B	3.85 GB
Counters (text, image, page)	Run B	(729, 259, 472)

Layout-first locking (Card-based + LAP solver) prevents local swaps; anchors provide high precision but limited cover- age; and candidate-only face verification confirms additional cases without allowing cross-identity leakage. Internal bench- marks on a proprietary subset showed 98.0% recall (96 out of 98 labeled positives) and 94.1% precision. These exact figures represent a specific deployment state and may vary across different corpora.

Latency Notes

Latency varies mainly with OCR-heavy scanned pages. The production log reports a mean runtime of 208.54 s/doc for Run B.

10. Discussion and Lessons Learned

Deterministic linking helps auditability: For person-image association, a layout-first LAP solver with explicit card-based grouping and optional candidate-only face verification is easier to inspect than opaque end-to-end caption binding, and it supports incremental improvement via feedback anchors.

Evidence mediation reduces exposure: RBAC filtering and redaction constrain what downstream generators can see, while citations preserve traceability.

Operational memory control matters: GPU fragmentation and transient low-memory states require explicit cleanup and conservative batching, especially on commodity hard- ware.

a) Threats to Validity and Reproducibility

Reviewers should note several factors limiting the generalizability of our findings. **External validity** is constrained by the proprietary nature of our production corpus; accuracy cannot be fully reproduced without released annotations. **Systemic bias** in person linking remains heavily dependent on upstream face detectability bias and OCR variance, which strongly bound the pipeline's overall recall. Finally, **operational metrics** exhibit configuration dependence; performance is tied to specific hardware tiers and the chosen combination of embed- ding models (e.g., CLIP vs. BGE-M3 parameter counts).

b) Why not use External Vision–Language APIS?

Uploading sensitive enterprise documents to external services can be legally and operationally unacceptable. Even local large multimodal models (e.g., LLaVA [54]) can be challenging under commodity VRAM budgets. GPT-4

technical reports [55] are capability reference points, not deployed dependencies.

c) Open Issues

Tables: Complex table structures remain difficult; heavy Table Structure Recognition (TSR) models such as TableFormer
[56] are a plausible next step.

Incremental updates: Faster incremental indexing would avoid rerunning full extraction on small edits.

11. Conclusion

Corpusio demonstrates that a practical multimodal extraction and retrieval system can run reliably under commodity GPU budgets while supporting confidentiality mediation. The log-verified run processed 46 heterogeneous files with mean runtime 208.54 s/doc and peak allocated GPU footprint 3.72 GB on an NVIDIA T4. The system combines deterministic person-image linking (layout-first card-based grouping + LAP solver 1-to-1 locking, feedback anchors, and candidate- only face verification), hybrid retrieval (optional BM25 + dense), and access-controlled evidence mediation.

Acknowledgment

Supported by Kirusa Inc. IR&D. We thank the engineering team for support and the AI4Bharat community for Indic resources [57]. The authors disclose the use of the Antigravity AI assistant [58] for manuscript preparation. Specifically, the system was utilized for grammar refinement, LaTeX lay-out optimization (including the generation of Fig. 1 from Graphviz specifications), and technical identifier formatting. The technical methodology and benchmarking results remain the original work of the authors.

Ethics Statement

The research involving human subject images reported in Section V was conducted under the oversight of the Kirusa Inc. Internal Review Board (IRB), and informed consent was obtained from all individuals whose data was used in the production benchmarks.

Conflict of Interest

The authors declare no conflict of interest regarding the pub-lication of this manuscript.

References

[1] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, ‘‘spacy: Industrial-strength natural language processing in python,’’ Software, 2020. [Online]. Available: https://spacy.io

[2] R. Smith, ‘‘An overview of the tesseract ocr engine,’’ in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. IEEE, 2007, pp. 629–633.

[3] Tesseract OCR Team, ‘‘Tesseract 4.00 lstm-based ocr engine,’’ Tesseract OCR Documentation, 2018. [Online]. Available: https://github.com/tesse ract-ocr/tesseract/wiki/4.0-with-LSTM

[4] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, ‘‘Trocr: Transformer-based optical character recognition with pre- trained models,’’ *arXiv preprint arXiv:2109.10282*, 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ‘‘Attention is all you need,’’ *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, ‘‘Bert: Pre-training of deep bidirectional transformers for language understanding,’’ *arXiv preprint arXiv:1810.04805*, 2018.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Un- terthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, ‘‘An image is worth 16x16 words: Transformers for image recognition at scale,’’ *ICLR*, 2021.

[8] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel *et al.*, ‘‘Retrieval-augmented generation for knowledge-intensive nlp tasks,’’ in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

[9] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, ‘‘Retrieval-augmented generation for large language models: A survey,’’ 2023.

[10] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[11] S. Robertson and H. Zaragoza, ‘‘The probabilistic relevance framework: Bm25 and beyond,’’ *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[12] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, ‘‘Proposed nist standard for role-based access control,’’ *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 224–274, 2001.

[13] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cour- napeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, ‘‘Scipy 1.0: fundamental algorithms for scientific computing in python,’’ *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, ‘‘Pytorch: An imperative style, high-performance deep learning library,’’ in *Advances in Neural Informa- tion Processing Systems*, 2019, pp. 8029–8041.

[15] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, ‘‘Layoutlm: Pre- training of text and layout for document image understanding,’’ in *Proceed- ings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1192–1200.

[16] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, Z. Wan *et al.*, ‘‘Layoutlmv2: Multi-modal pre-training for visually-rich document understanding,’’ in *Proceedings of the 59th Annual*

Meeting of the Association for Computational Linguistics, 2021, pp. 2579–2591.

[17] S. Appalaraju, B. Jasani, and B. U. Kota, ''Docformer: End-to-end trans- former for document understanding,'' in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 993–1003.

[18] X. Zhong, J. Tang, and A. J. Yepes, ''Publaynet: largest dataset ever for document layout analysis,'' in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1015–1022.

[19] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, ''Docbank: A benchmark dataset for document layout analysis,'' in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 949–960.

[20] A. W. Harley, A. Ufkes, and K. G. Derpanis, ''Evaluation of deep con- volutional nets for document image classification and retrieval,'' in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 991–995.

[21] G. Jaume, H. K. Ekenel, and J. Thiran, ''Funsd: A dataset for form under- standing in noisy scanned documents,'' in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 2. IEEE, 2019, pp. 1–6.

[22] J. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, Mensch, K. Millican, M. Reynolds *et al.*, ''Flamingo: a visual language model for few-shot learning,'' *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.

[23] Jia, Y. Yang, Y. Xia, Y. Chen, Z. Parekh, H. Pham, Q. Le, Y. Sung, Z. Li, and T. Duerig, ''Scaling up visual and vision-language representation learning with noisy text supervision,'' in *International Conference on Machine Learning*. PMLR, 2021, pp. 4904–4916.

[24] M. Mathew, D. Karatzas, and C. Jawahar, ''Docvqa: A dataset for vqa on document images,'' in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2200–2209.

[25] G. Jocher, A. Chaurasia, and J. Qiu, ''Ultralytics yolo,'' GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[26] J. Redmon and A. Farhadi, ''Yolov3: An incremental improvement,'' *arXiv preprint arXiv:1804.02767*, 2018.

[27] A. Bochkovskiy, C. Wang, and H. M. Liao, ''Yolov4: Optimal speed and accuracy of object detection,'' *arXiv preprint arXiv:2004.10934*, 2020.

[28] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, ''Yolov10: Real-time end-to-end object detection,'' *arXiv preprint arXiv:2405.14458*, 2024.

[29] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, ''Realm: Retrieval- augmented language model pre-training,'' in *Proceedings of the 37th In- ternational Conference on Machine Learning*, 2020.

[30] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, ''Dense passage retrieval for open-domain question answering,'' in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781.

[31] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, Stoyanov, and L. Zettlemoyer, ''Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,'' in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[32] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, ''Query rewriting for retrieval-augmented large language models,'' *arXiv preprint arXiv:2305.14283*, 2023.

[33] X. V. Lin, X. Chen, M. Chen, W. Shi, M. Lomeli, R. James, P. Rodriguez, J. Kahn, and A. Szlam, ''Ra-dit: Retrieval-augmented dual instruction tuning,'' *arXiv preprint arXiv:2305.06983*, 2023.

[34] R. Nakano *et al.*, ''Webgpt: Browser-assisted question-answering with human feedback,'' *arXiv preprint arXiv:2112.09332*, 2021.

[35] D. Kiela, I. Fostiropoulos, and X. Zhai, ''Multi-task retrieval for generative models,'' arXiv preprint arXiv:2101.07769, 2021. [Online]. Available: https://arxiv.org/abs/2101.07769

[36] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, ''Language models as knowledge bases?'' in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 2463–2473.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sas- try, A. Askell, P. Mishkin, J. Clark *et al.*, ''Learning transferable visual models from natural language supervision,'' in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[38] N. Reimers and I. Gurevych, ''Sentence-bert: Sentence embeddings using siamese bert-networks,'' in *Proceedings of the 2019 Conference on Empir- ical Methods in Natural Language Processing*, 2019, pp. 3982–3992.

[39] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, ''Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation,'' *arXiv preprint arXiv:2402.03216*, 2024.

[40] S. Xiao, Z. Liu, P. Zhang, and N. Muennighoff, ''C-pack: Packaged resources to advance general chinese embedding,'' arXiv preprint arXiv:2309.07597, 2023. [Online]. Available: https://arxiv.org/abs/2309.0 7597

[41] J. Johnson, M. Douze, and H. Jégou, ''Billion-scale similarity search with gpus,'' *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[42] Y. A. Malkov and D. A. Yashunin, ''Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,'' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2018.

[43] H. Jégou, M. Douze, and C. Schmid, ''Product quantization for nearest neighbor search,'' *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 1, pp. 117–128, 2011.
[44] A. Ankane, ‘‘pgvector: Open-source vector similarity search for postgres,’’ GitHub repository, 2021. [Online]. Available: https://github.com/pgvector/pgvector
[45] O. Khattab and M. Zaharia, ‘‘Colbert: Efficient and effective passage search via contextualized late interaction over bert,’’ *arXiv preprint arXiv:2004.12832*, 2020.
[46] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, ‘‘Col- bertv2: Effective and efficient retrieval via lightweight late interaction,’’ *arXiv preprint arXiv:2112.01488*, 2021.
[47] A. Mhaske, H. Patil, J. Pawar, A. Patil, and P. Bhattacharyya, ‘‘Indicner: A deep learning based system for named entity recognition for Indian languages,’’ in *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, 2021.
[48] D. Kakwani, A. Kunchukuttan, S. Golla, G. N.C., A. Bhattacharyya, M. M. Khapra, and P. Kumar, ‘‘Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages,’’ arXiv preprint arXiv:2009.07288, 2020. [Online]. Available: https://arxiv.org/abs/2009.07288
[49] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, ‘‘Stanza: A python natural language processing toolkit for many human languages,’’ in *Pro- ceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020, pp. 101–108.
[50] A. Akbik, D. Blythe, and R. Vollgraf, ‘‘Contextual string embeddings for sequence labeling,’’ in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.
[51] A. Rahimi, Y. L. Li, and T. Cohn, ‘‘Massively multilingual transfer for ner,’’ in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 151–164.
[52] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, ‘‘Unsupervised cross-lingual representation learning at scale,’’ in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.
[53] J. McKie and J. Gaucher, ‘‘Pymupdf,’’ GitHub repository, 2023. [Online]. Available: https://github.com/pymupdf/PyMuPDF
[54] H. Liu, C. Li, Q. Wu, and Y. J. Lee, ‘‘Visual instruction tuning,’’ in *Advances in Neural Information Processing Systems*, 2023.
[55] J. Achiam *et al.*, ‘‘Gpt-4 technical report,’’ *arXiv preprint arXiv:2303.08774*, 2023.
[56] A. Nassar, O. Livne, P. Makarov, and A. Nikolaidou, ‘‘Tableformer: Ta- ble structure understanding with transformers,’’ in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4614–4623.
[57] A. Kunchukuttan, D. Kakwani, S. Golla, A. Bhattacharyya, M. M. Khapra, and P. Kumar, ‘‘Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages,’’ arXiv preprint arXiv:2005.07117, 2020. [Online]. Available: https://arxiv.org/abs/2005.07117
[58] Antigravity, ‘‘Antigravity: An advanced agentic coding ai assistant,’’ Inter- active AI Environment, 2026.

Author Profile

Mahanthi Bharadwaj Phani Datta received the B.Tech. degree in computer science and engineering from Koneru Lakshmaiah Education Foundation (K L University), Vijayawada, India, in 2020, and the M.Sc. degree in data science from the University of Surrey, Guildford, U.K., in 2023. He is currently an Associate AI and ML Engineer with Kirusa, Bengaluru, India. Previously, he served as a Power BI Developer with Universal tech Solutions Ltd, London, U.K., and as a Network Engineer with Mphasis, Chennai, India. His professional background also includes cloud administration experience from his early career. He is the author of one published article. His active research interests span large language model evaluation, retrieval-augmented generation, multilingual conversational AI, and the intersection of computer vision and deep learning.

Kella Kedhareesh received the B.Tech. degree in chemical engineering from the National Institute of Technology Agartala (NITA), India, in 2024. He is currently an AI & ML Engineer with Kirusa, Bengaluru, India. He contributed to Corpusio, an enterprise multimodal document search ecosystem, where he architected the Hybrid RAG pipeline and designed a Graph RAG layer for relational knowledge discovery. Beyond his production work, he actively explores frontier AI research conducting experiments in latent space prediction, memory-augmented latent representations, and world model architectures, with a particular focus on vision-language models such as VL-JEPA. This work spans both personal research and internal R&D, including hands-on model fine-tuning and architectural study of latent world model systems. His broader research interests include predictive world models, retrieval-augmented generation, and the application of self-supervised learning to real-world domains.

Kamaleeswari Kamboji received the B.Tech. degree in computer science and engineering from CVR Engineering College, Hyderabad, India, in 2019, and the M.Sc. degree in advanced computer science from Cardiff University, Cardiff, U.K., in 2023. With over five years of industry experience, she is currently an AI/ML Engineer with Kirusa, Inc., India. Her technical contributions center on multimodal document understanding, layout-aware information extraction, and retrieval augmented generation (RAG) systems. Specifically, she has played a key role in designing staged extraction pipelines that integrate YOLO-based layout detection, vision embeddings, and hybrid sparse–dense retrieval algorithms. Her broader research interests encompass artificial intelligence, multilingual natural language processing, scalable data systems, and privacy-preserving AI for industrial deployments.

Kancharala Subhaashini received the B.Sc. degree in mathematics, statistics, and computer science from Sri Nagarjuna Degree College, India, and the M.Sc. degree in data science from the Vellore Institute of Technology, India. She is currently an Associate AI and ML Engineer based in India. Her professional experience spans the full computational lifecycle, from data curation and model

development to the optimization and large-scale deployment of AI systems. She led key components of the Corpusio platform's technical design, specifically focus- ing on multimodal embedding pipelines, vision–language integration, and performance benchmarking for secure, cloud-native environments. Her academic and professional interests involve scalable multimodal architectures, explainable AI, efficient inference strategies, and the engineering of safety- aware large language models.

Raja N. Moorthy received the Ph.D. degree in computer science and engineering from the Indian Institute of Technology Delhi, India. With over four decades of technical leadership and telecom industry expertise, he currently serves as Chief Operating Officer at Kirusa, Inc., where he directs product engineering and the comprehensive software development life cycle. His innovation in next-generation messaging is reflected in seven granted patents across global jurisdictions (including the US, EU, and China), covering areas such as call completion routing, multimodal messaging convergence, and multimedia voicemail platforms. As a senior technology innovator, his ongoing work establishes critical foundations for modern IoT infrastructure. His active research interests include the design of highly scalable SaaS architectures, high-performance mobile engineering, and the deployment of production-grade artificial intelligence pipelines.