Journal of Engineering Issues and Solutions

Intelli-Docs: An AI-Powered Personal Document Assistant Using Retrieval-Augmented Generation and Multimodal Retrieval

Prashant Bhattarai[1,*], Saurab Baral[2] , Kritika Thapa[3], Roshan Chaudhary[4]

[1,2,3,4] *Department of Electronics and Computer Engineering, IOE, Purwanchal Campus, Tribhuvan University, Nepal*
*Corresponding email: prashantbhattarainepal@gmail.com
Received: May 12, 2026; Revised: 13 June, 2026; Accepted: July 11, 2026

Abstract

Managing personal documents remains a hassle: files accumulate across cloud drives, email, and local devices in formats that mix scanned images with digital text, and conventional retrieval based on filenames and folder hierarchies handles neither ambiguous queries nor cross-format access well. This paper presents Intelli-Docs, a personal document assistant that combines retrieval-augmented generation (RAG) with multimodal retrieval so that users can locate and question their documents through natural language queries. The system ingests documents through three coordinated pipelines. A text pipeline extracts and chunks document text, embeds the chunks, and stores them in a vector index. A query pipeline classifies each request as text retrieval, image retrieval, or general conversation and routes it accordingly. An image pipeline processes scanned and photographic documents with three parallel models: a Tesseract engine for optical character recognition, a BLIP model for caption generation, and a CLIP ViT-B/16 encoder for semantic image embeddings, whose outputs are fused into a single textual representation. Retrieved evidence is passed to a large language model that generates the final grounded answer. On image-text retrieval benchmarks, the retrieval component reached Recall@5 of 92% and mean average precision of 0.89, with a mean query latency of 2.7 s. We describe the architecture, the evaluation protocol, and the observed trade-offs between accuracy and on-device cost, and we discuss the limitations that constrain mobile deployment.

Keywords: *retrieval-augmented generation; large language models; optical character recognition; CLIP; multimodal retrieval; personal document management*

1. Introduction

People now hold most of their personal records in digital form, spanning academic transcripts, financial statements, medical histories, and legal contracts. These records are saved in incompatible formats such as PDFs, photographs, and scanned paper, and they are spread across cloud storage, email attachments, and personal devices. As the shift to digital has resulted in better transportability than before, another issue was created: a fragmented management system whereby there is not one tool that can effectively find and store a user's documents. The other challenge with retrieval systems that are based on using the exact name of the

file and the system you have established using nested folders to store your files is the conditional nature of queries. When a query is phrased loosely, or any target document is stored in an image and not in a machine-readable format, those retrieval systems will struggle to return those requested items. (de Barcelos Silva et al., 2020; Ling et al., 2020).

Take this example that shows the gap. A physician who also oversees some research is looking for her child's vaccinations in order to meet a deadline for school; meanwhile, she's submitting tax records & transcripts in order to apply for a grant. She knows that the documents she needs exist, but they are inconsistent in how they are named; varied between systems (some duplicates), and part of their information is locked in a scanned document. Because of the way the keyword search works, she's getting loose responses related to her search; and virtual assistants are not able to retain the context of previous documents when she's searched for something before, each search starts from the very first entry. The hoarding of personal documentation increases the effort to store and retrieve that documentation a lot.(Whittaker, 2011).

All of the current alternatives solve only part of the overall problem. For instance, cloud storage technologies synchronize or replicate files but do not provide any level of interpretation of the contents of those files; commercial personal assistants generally will not have any long-term memory (persistent) for the user's own documents; and dedicated document-processing workflows typically will extract fields from image files, but they do not have any conversational retrieval mechanisms associated with them (Rusli et al., 2021; Ling et al., 2020). Due to their very limited integration in practice, understanding semantics as a component of durable memory and privacy requires a unified system (that ingests documents regardless of type, stores them securely, and provides clear answers to questions posed in natural language).

Distinct from traditional web searching or enterprise searching is the issue of personal document management; personal documents differ from typical web documents in two key respects. The first distinction between personal documents and standard web documents is how large or small the document collection is, as well as the variety of different document types that may be present in the personal document collection. For example, a lone individual may only have 1000 total paper documents and 10,000 electronic documents, yet the types of documents included in their paper documents can be very different (e.g., unique types of receipts). Therefore, a system cannot assume that all documents being processed will have the same format. Thus, a personal document management system needs to be designed to accommodate any possible input format in the ingestion stage; the retrieval stage must allow for direct comparison between items of different types, and the overall workflow has to ensure that the user has complete control over where and how their data is stored.

We propose Intelli-Docs, a personal document assistant that unifies multimodal ingestion, semantic indexing, and retrieval-augmented generation (Lewis et al., 2020). The system is designed to interpret documents whether they are text-based or image-based, to store their representations in a vector index, and to answer questions such as “Show me all property-tax receipts from 2023 submitted to Bank X.” Drawing from established components of language representation (Devlin et al 2019), dense retrieval (Karpukhin et al 2020), Optical Character Recognition (Smith 2007) and Vision-Language Modeling (Radford et al 2021; Li et al 2022), we created an organization around a query/classification stage which allows us to keep the inputs of text, image, and conversation requests on separate paths.

The main contributions of this work are as follows:

- A complete design structure consisting of a RAG text processing system, a stage that classifies the query, as well as a simultaneous processing of images of documents, making it possible to obtain

access to heterogeneous documents using one natural language interface.

- A multimodal image-processing design that combines OCR text, BLIP-generated captions, and CLIP image embeddings into a single representation, enabling retrieval of documents.
- A strong evaluation on image-text retrieval benchmarks that reports ranking accuracy, similarity separation, and latency, together with an analysis of the accuracy-cost trade-offs relevant to mobile deployment.

The remainder of the paper is organized as follows. Section 2 demonstrates related systems and positions Intelli-Docs against them. Section 3 explains the datasets, preprocessing, retrieval and generation methods, and application architecture. Section 4 reports the evaluation and discusses the results and limitations. Section 5 concludes and outlines future work.

2. Related Work

Four distinct areas of Research associated with IntelliDocs: retrieval-based generation, intelligent personal assistants, document processing and Optical Character Recognition, and Vision-Language Retrieval. These four strands are first summarised, then compared in Table 1.

2.1 Intelligent Personal Assistants

A series of systematic evaluations on intelligent assistants have noted how the area has shifted from interfaces based upon voice command systems towards a supporting role in completing tasks. In addition to this, notes indicate recurring areas of weakness in terms of personalization, retention of context, and building trust with users (De Barcelos Silva et al., 2020; Zierau et al., 2020). Typically, intelligent assistants are designed for short and slightly casual conversations, and most do not hold onto durable, traceable records of files being used to complete the user's request. While specifically report evaluating areas of trust toward these types of systems point toward transparency and controllability of personal data representatives as key to determining if users will adopt/use them (Zierau et al., 2020), they also support an argument for the design of an intelligent assistant which queries documents given to the intelligent assistant by the user and grounds the responses from the assistant back to these documents. In contrast, the Intelli-Docs design establishes the full collection of documents that a user has stored/created to be the source of knowledge, with each response given by Intelli-Docs being provided from a passage extracted from the user's document collection, thereby allowing all responses to be ultimately derived back to a specific document stored within Intelli-Docs for validation/purposes.

2.2 Document Processing and Optical Character Recognition

There has been a lot of work devoted to transforming document images into structured text. The Tesseract OCR (optical character recognition) engine, one of the most widely used open-source OCR engines, provides the character-recognition stage of many pipelines (Smith, 2007). Newer systems are combining OCR technologies with language-processing technologies to help extract information from identity documents (for example, identity document extractors), using natural-language processing-based postprocessing to correct errors and structure fields (Rusli et al., 2021) and using intelligent document processing systems that integrate robotic process automation with machine learning to classify and route large volumes of processed documents at once (Ling et al., 2020). Although these systems do a great job with respect to extracting data, they typically stop far short of enabling user-friendly query retrieval, creating a gap between text recognition and question-and-answering capabilities about the recognized text. All of the visual document-

understanding comparative reviews of current methodologies show that most modern methodologies do not provide for question-answering technologies regarding the recognized text, focusing instead on recognition and layout analysis (Nandi & Sathya, 2024). Intelli-Docs combines OCR with a second input (whereas the other systems focus solely on OCR as being the only processing input) and links it with a retrieval and generation component.

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation(RAG) is a method that helps the large language model provide more accurate or reliable answer. Since the large language model provides answer only from the data it learned during training, RAG help the model to search for relevant information from external knowledge source .It filters out the most relevant documents from the source, and feed them in the model as context, from which the model provides the correct answer. This helps to reduce incorrect or made-up information and allow using the latest data without retraining the model(Lewis et al., 2020).

Most RAG system use dense retrieval, where the user question and the stored documents is convert into vector embeddings. The system compare these vectors and retrieve the documents most similar to the query(Karpukhin et al., 2020). Compared to keyword search, this method understand the meaning of the question better, even if different words is used.

RAG is very useful for personal document management because users keep adding new documents over time. Instead of train the model again when new files is added, the system only need to update its document database. In Intelli-Docs, text documents is stored in a vector database, while image documents first converted into text using OCR and image captioning. After that, both text and image information can be search using the same retrieval process. This allow the system to answer question from both text-based and image-based documents using single knowledge base. Recent studies also show that combine RAG with multimodal retrieval can improve document search by allowing the system to work with both text and images(Mei et al., 2025).

2.4 Vision-Language Retrieval

Vision-language models are AI models which understand both images and text. It converts the images and the text to a shared embedding space, so that we can compare how similar they are.

CLIP is trained on millions of image-caption pairs. It can find images that matches a text description, by comparing their embeddings (Radford et al., 2021).

BLIP helps this by automatically creating and cleaning image captions during training (Li et al., 2022).

In Intelli-Docs, BLIP generates the caption or description of the images, CLIP creates an embedding that captures the meaning of the image, and OCR extracts the readable text. By combining these three methods, the system can find matching documents or image even when the image contains very little readable text.

2.5 Positioning of Intelli-Docs

Table 1 contrasts the strands above along the dimensions most relevant to personal document management. Prior systems tend to excel on one axis while leaving the others unaddressed; Intelli-Docs is designed to combine multimodal input, a persistent vector index, and grounded generation within a single privacy-conscious workflow.

Table 1: Comparison of related approaches with Intelli-Docs.

Approach	Multimodal input	Persistent index	Grounded generation	Key limitation
Cloud drives (filename/ folder)	No	No	No	No content understanding
Personal assistants	Partial	No	Partial	Weak document memory
OCR / IDP pipelines	Yes (image)	Partial	No	No conversational query
RAG QA systems	No (text)	Yes	Yes	Text-only inputs
Vision-language retrieval	Yes (image)	Partial	No	No answer generation
Intelli-Docs (this work)	Yes (text+image)	Yes	Yes	On-device cost

3. Materials and Methods

3.1 System Overview

Intelli-Docs is organized as three coordinated pipelines that share a single vector index and a single generation stage. The text pipeline prepares born-digital documents for retrieval. The query pipeline interprets each user request and routes it to the appropriate handler. The image pipeline converts scanned and photographic documents into text-like representations so that they can be indexed and retrieved by the same machinery as ordinary text. All three converge on a retrieval-augmented generation stage in which a large language model composes a grounded answer from the retrieved evidence. This separation of concerns keeps each modality on a dedicated path while allowing cross-modal matching at retrieval time.

3.2 System Architecture

Figure 1 shows the end-to-end architecture. For text documents like PDF, first the PDF files are processed to extract the text out of it. Then the extracted text is divided into small overlapping chunks using a recursive character splitter. These chunks are then converted into the embeddings which are then stored in the vector database. Now for the retrieval or the semantic search, the embedding is fetched from the vector database and does the similarity search to retrieve the related content.

When a user submits a query, the query goes to the classification model which determines the type of request. The classification is done to know where the query belongs to these three: text extraction, image retrieval, or general conversation.

In the text extraction pipeline, the user submitted query is converted to the embeddings and compared with the stored document embeddings in vector database. The most similar chunks to the query are retrieved.

In the image retrieval pipeline, three models run simultaneously to process the document image. Tesseract OCR extracts readable text from the image, BLIP generates caption or description of the image, and CLIP ViT-B/16 creates an embedding that represents the overall meaning of the image so that the user can query the image using the text. These three outputs are combined into a single representation. The user query embeddings is then compared with the stored document embeddings using the cosine similarity to

retrieve the most nearest documents, even though the image contains very little readable text.

These retrieved information is then passed to the Retrieval-Augmented Generation (RAG), which passes this retrieved information to the language model as the context. The language model then generates the final response for it.

The query classification stage is an important part of this whole system. Instead of sending every query through the three pipelines, this classifier helps to know the intention of the user and selects only the most appropriate one. This reduces the unnecessary computation and improves the response time. For example, if the user wants the image retrieval request, this request is sent only to the image pipeline instead of searching through all text documents. Similarly general conversation requests such as explanation or definition, are directly sent to the language model without performing the document retrieval.

Here each pipeline works independently, so the system is easier to expand. Support for new document type can be added by creating another processing pipeline which produces embeddings compatible with the same vector database.

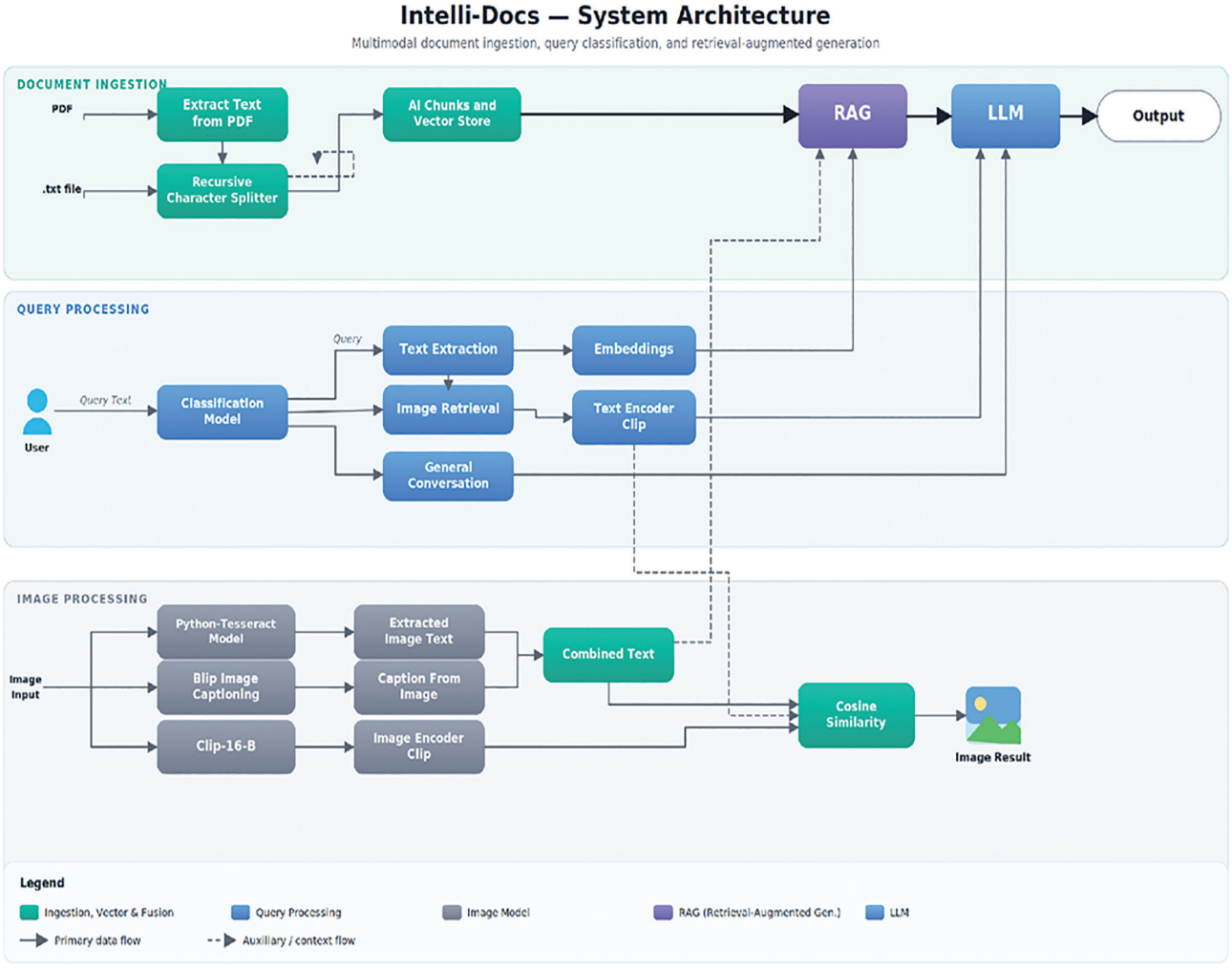

Figure 1: Intelli-Docs system architecture, showing the document-ingestion, query-processing, and image-processing pipelines converging on the RAG and LLM stages.

3.3 Datasets

The image components were evaluated on publicly available and custom image-text collections rather than on private personal records, both to protect privacy and to allow reproducible measurement of the retrieval components. The Flickr30K collection provides 31,783 images, each paired with five reference captions, which makes it a standard benchmark for image-text retrieval and caption generation. A custom tour-image set was assembled to test retrieval and zero-shot classification on photographs with characteristics similar to user-captured document images. A BERT-base-uncased model was fine-tuned on a custom labeled set to support domain-specific classification. Table 2 summarizes the three sources. These collections evaluate the image-retrieval and classification stages specifically; the document-management workflow that surrounds those stages is described qualitatively, and evaluation on a dedicated personal-document corpus is identified in Section 4 as future work.

Table 2: Datasets used to evaluate the image and classification components.

Data source	Description	Format
Flickr30K	31,783 images with five captions each; used for image-text retrieval and caption generation.	Image / TXT
Custom tour-image set	Travel photographs used to test retrieval accuracy and zero-shot classification on user-style images.	Image / TXT
BERT-base-uncased (fine-tuned)	Fine-tuned on a custom labeled set for domain-specific classification related to the tour images.	TXT

3.4 Data Preprocessing

Raw text from both born-digital and OCR-derived sources was normalized before indexing. Cleaning removed extraneous whitespace, stray punctuation, and common recognition artifacts introduced by OCR, such as substituted characters and broken words at line ends. Normalization lowercased text and applied lemmatization so that surface variants of a word mapped to a single form, which prevents trivial spelling differences from splitting otherwise identical content across the embedding space. Named-entity recognition then extracted salient fields such as names, dates, addresses, and monetary amounts, which are useful both for filtering results and for answering questions that hinge on specific values, for example a request restricted to receipts from a particular year. This preprocessing improves the quality of the embeddings and makes document retrieval more accurate. So by cleaning the text first, the system creates better embeddings and reduces incorrect or false matches during similarity search.

3.5 Retrieval and Generation Methods

3.5.1 Retrieval-Augmented Generation

Retrieval-augmented generation lets a language model draw on an external corpus at inference time instead of relying solely on its parameters (Lewis et al., 2020). In Intelli-Docs, document chunks are embedded and stored in a vector index; at query time the most similar chunks are retrieved and supplied to the model as context, so the generated answer is grounded in the user's own documents. This design lets the assistant answer questions about private content that no general model was trained on, and it allows the knowledge base to be updated by adding documents rather than by retraining.

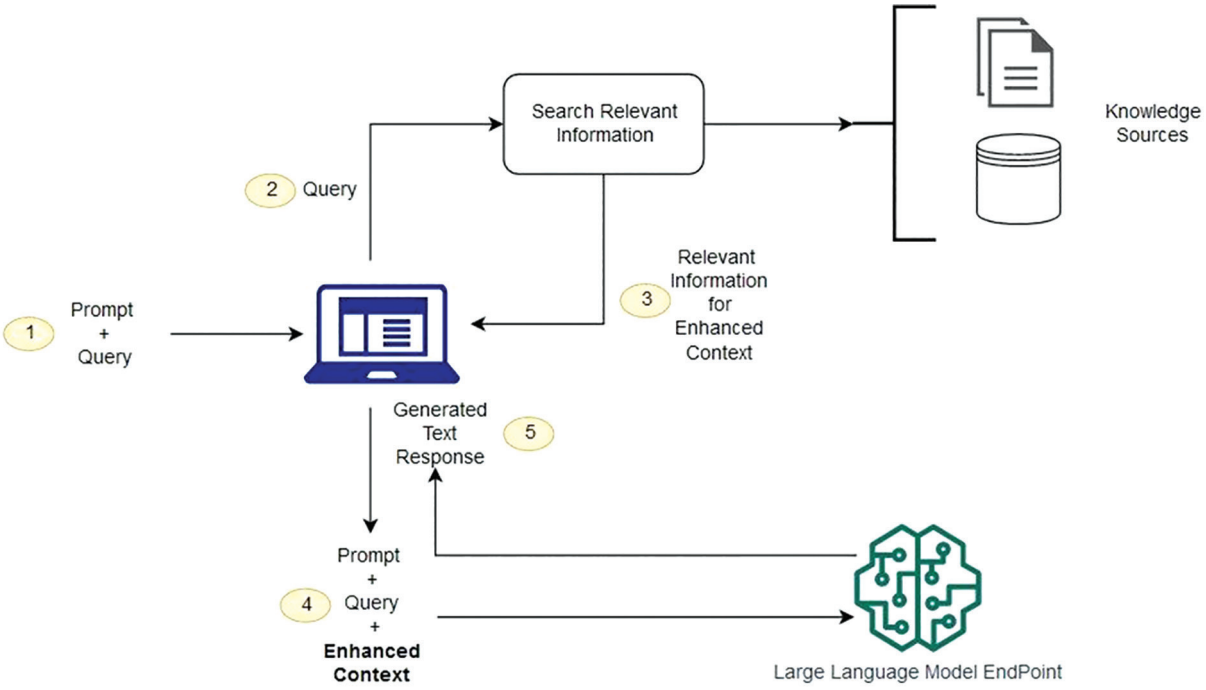

Figure 2: RAG Framework

3.5.2 Image Captioning with BLIP

BLIP generates a descriptive caption for each image document, which contributes a natural-language summary that the retrieval stage can index alongside OCR text (Li et al., 2022). For captioning, the model is trained with a cross-entropy objective that maximizes the likelihood of the reference caption given the image. Writing T for the caption tokens and I for the image, the loss minimized during training is

$$L_{cap} = -\Sigma_t \log P(t_t \mid t_{<t}, I) \text{----------(1)}$$

where the sum runs over the tokens of the caption and each token is predicted from the image and the previously generated tokens.

3.5.3 Image Embeddings with CLIP ViT-B/16

CLIP encodes images into the same space as text, which makes direct image-text comparison possible (Radford et al., 2021). The image encoder is a Vision Transformer, ViT-B/16, that divides the input image I into a set of patches $P = \{p_1, \ldots, p_n\}$, projects each patch, adds positional information, and passes the sequence through a transformer encoder:

$$z_I = \text{TransformerEncoder}(\text{PatchEmbed}(P) + E_{pos}) \text{----------(2)}$$

The resulting fixed-length vector is used both to index image documents and to match them against text queries.

Combining these three image signals is deliberate. OCR recovers exact printed strings, which is essential for documents whose value lies in specific text such as a card number or a name; captioning supplies a natural-language description that helps when a query refers to the content of a photograph rather than to printed

text; and the CLIP embedding captures visual semantics that neither OCR nor captioning expresses directly. A document that carries little readable text, such as a photograph of a landmark, may be missed by OCR entirely but still retrieved through its caption and embedding. Fusing the three into a single representation means the retrieval stage can draw on whichever signal is informative for a given document without a separate index for each.

3.5.4 Text Encoding in CLIP

The CLIP text encoder converts a query into a vector in the shared space so that it can be compared with image embeddings. Each input token is embedded, giving a matrix

$$E = \text{Embed}(T) \in \mathbb{R}^{n \times d} \quad (3)$$

and positional encodings are added to preserve token order:

$$E' = E + \text{PosEnc}(n) \quad (4)$$

The encoded query is then used to retrieve the most similar image documents.

3.5.5 Optical Character Recognition with Tesseract

The Tesseract engine performs the literal text extraction for image documents (Smith, 2007). Through the pytesseract interface, an input image is pre-processed to improve contrast and orientation, segmented into blocks, lines, and characters, and recognized by combining learned character models with layout analysis; language models and spell-checking then correct residual errors. In Intelli-Docs this stage recovers text from documents such as identity cards and licenses, where the printed content is the key retrieval signal. Because recognition quality depends heavily on input quality, the pre-processing that precedes recognition matters as much as the engine itself: skew correction and contrast enhancement reduce the character substitutions that would otherwise propagate into the index and degrade later matches. The recovered text is passed through the same cleaning and normalization described in Section 3.4 before it is embedded, so that OCR output and born-digital text are treated uniformly downstream.

3.5.6 Cosine Similarity

Retrieval ranks documents by the cosine of the angle between the query embedding and each stored embedding, which measures orientation rather than magnitude. For two vectors A and B,

$$\cos\theta = (A \cdot B) / (\|A\| \, \|B\|) \quad (5)$$

Higher values indicate closer semantic alignment. Because OCR text, captions, and embeddings are all mapped into comparable spaces, the same similarity measure supports both text and cross-modal retrieval.

3.5.7 Classification with a Fine-Tuned BERT Model

Query routing and domain classification use a BERT model, which is pre-trained with a masked-language-modeling objective and then fine-tuned on labeled data (Devlin et al., 2019). The masked-language-modeling loss maximizes the probability of the masked tokens given their context:

$$L_{MLM} = -\Sigma_i \log P(token_i \mid context) \quad (6)$$

Fine-tuning adapts these general language features to the classification decisions needed for routing.

3.6 Application Architecture

The deployed system separates the client, the backend, and cloud storage. A Flutter mobile application is the primary screen interface; users upload, search, and query documents with natural language, and the app forwards requests to the backend. A FastAPI backend is the core processing unit: it handles storage and retrieval of the requests, coordinates the OCR, captioning, and embedding models, invokes the language model for generation, and manages file access. Cloudinary stores the uploaded documents and images in the cloud, so that the backend can access them very quickly during processing. This keeps the mobile application lightweight because all the heavy AI models run on the server side instead of running in the user's compatible phone. Running the models on the backend has many of the advantages and disadvantages. Since the phone doesn't do the AI processing, the app remains fast and uses less storage and processing power. But the disadvantage is the user may experience a slight delay when the request is sent to the server and the response is returned. Storing documents in the cloud also helps to save space on the user's device and it allows the backend to retrieve the files whenever they are needed. However, as the documents are uploaded to the cloud, privacy becomes very important. So security measures like user authentication and appropriate data retention policies are used.

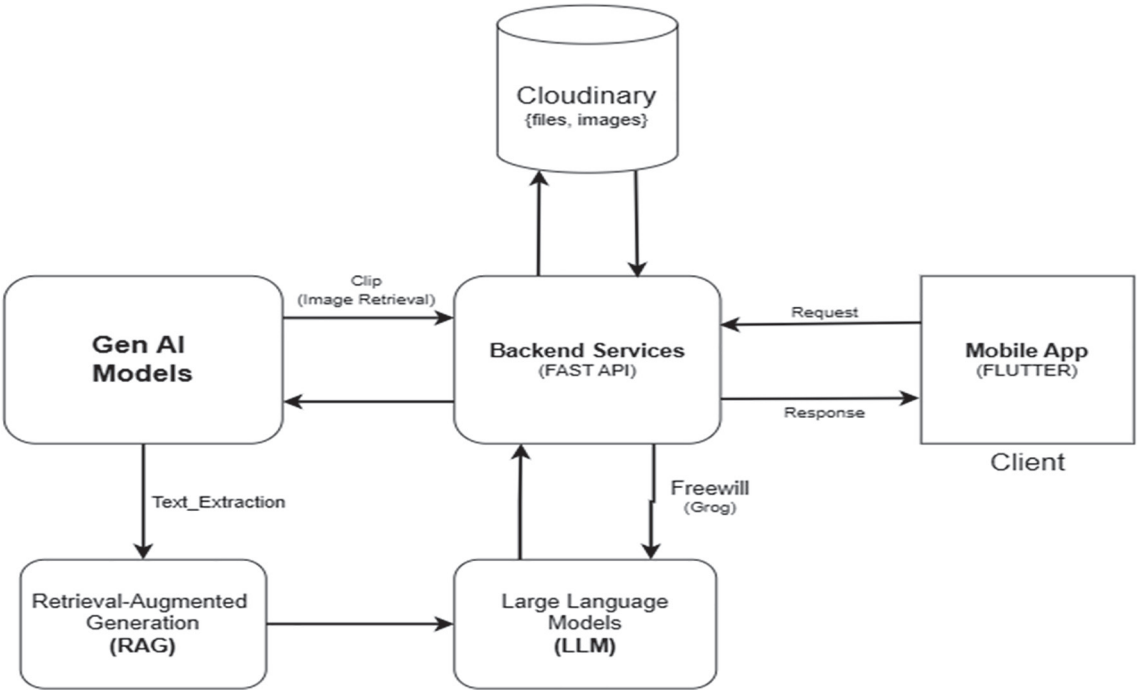

Figure 3: Application Architecture

4. Results and Discussion

4.1 Evaluation Protocol

The retrieval components were evaluated on the image-text collections described in Section 3.3. For each test query, the system has returned a ranked list of documents, and the rank of the correct target was been recorded. From these ranks we computed recall at cut-offs of 1, 5, and 10, the median rank, and the mean reciprocal rank, which together characterize how often and how highly the correct document is retrieved.

Precision at cut-offs of 1 and 5 and the mean average precision summarize ranking quality across the returned lists. To assess how well the embedding space separates matches from non-matches, we recorded the mean cosine similarity of correct (positive) pairs and of incorrect (negative) pairs. Finally, we measured end-to-end query latency, sustained upload throughput, and peak memory to characterize runtime cost. The metrics reported below are the values obtained under this main protocol; a comparison against a keyword-retrieval baseline on a dedicated personal-document corpus is identified as the primary extension of this evaluation.

4.2 Retrieval Performance

Table 3 reports the ranking results. The correct document appeared as the top result 78% of the time and within the top five results 92% of the time, rising to 97% within the top 10. The median rank of the correct document was 2, and the mean reciprocal rank was estimated to be 0.86, indicating that when the target was not first it was usually near the top. Precision was 78% at rank 1 and 84% within the top five, and the mean average precision across recall levels was 0.89. Taken together, these figures show that th`e fused representation retrieves the intended document reliably within a short candidate list, which is the regime that matters for an interactive assistant where users inspect only the first few results.

Table 3: Retrieval and ranking performance on the image-text benchmark.

Metric	Value
Recall@1	78%
Recall@5	92%
Recall@10	97%
Median rank	2
Mean reciprocal rank (MRR)	0.86
Precision@1	78%
Precision@5	84%
Mean average precision (mAP)	0.89

4.3 Similarity Separation

The embedding space distinguished matches from non-matches by a clear margin. Correct pairs had a mean cosine similarity of 0.82, whereas incorrect pairs averaged 0.42. This separation of roughly 0.40 explains the ranking results: because relevant documents sit substantially closer to the query than irrelevant ones, a similarity threshold can admit true matches while rejecting most distractors. The gap also indicates that the fusion of OCR text, captions, and image embeddings produces representations that are discriminative rather than uniformly high, which is what allows precision to stay high at the top of the ranking.

4.4 Computational Efficiency

Runtime cost was measured to gauge suitability for mobile use. The mean query response time was 2.7 s, below the 3 s threshold generally considered acceptable for interactive retrieval. The system sustained about 23 document uploads per second, which is adequate for individual use and comparable to common ingestion rates. Peak memory reached 1.8 GB, higher than typical mobile-optimized applications and attributable to the concurrent OCR, captioning, and embedding models. The latency and throughput are therefore acceptable, but the memory footprint is the main obstacle to running the full pipeline on-device, which motivates the compression strategies discussed below. The current architecture mitigates part of this cost by keeping

the heavy models server-side and by routing each query to a single pipeline, so that the full image stack is invoked only for image requests; the remaining pressure comes from holding three vision-language models in memory simultaneously, which is the specific target for quantization and selective loading in future work.

4.5 Discussion

The results support the central design choice of converting every document, whether text or image, into a common representation that a single retrieval stage can rank. High recall within the top five results and a large similarity margin together mean that the assistant can present a short, accurate candidate list rather than a long noisy one, which is what makes conversational retrieval usable. The main cost is computational: running three image models in parallel raises memory use and contributes to latency. This trade-off is favorable on server hardware but constrains a fully on-device deployment. Two limitations of the evaluation should be noted. First, the benchmarks are image-text collections rather than a personal-document corpus, so the reported numbers characterize the retrieval components rather than the full document-management workflow. Second, the evaluation does not yet include a head-to-head comparison against a keyword-retrieval baseline; adding one would quantify the improvement over conventional search directly. Both are addressed in the next subsection and in the concluding section.

The gap between Recall@1 (78%) and Recall@5 (92%) is informative about the failure modes. When the target is not ranked first, it is usually still within the top few, which points to near-miss confusions rather than outright retrieval failures: visually or textually similar documents compete for the top position, and the fused representation resolves most but not all of these within a short list. The remaining errors concentrate in two situations. The first is documents that are visually similar but semantically distinct, where the CLIP embedding dominates and pulls in look-alike images. The second is low-quality scans, where OCR produces noisy text that weakens the literal-text signal and shifts the burden onto the caption and embedding. Both patterns suggest concrete improvements, such as re-ranking the top candidates with a stricter text-match check and improving image pre-processing before OCR, and neither undermines the finding that the correct document is almost always retrieved within the first few results.

4.6 Limitations

Despite implementing the RAG, language-model, and OCR components, the system faces limitations that affect its scalability and usability, most of which stem from the cost of running several models together. The language-model and OCR stages consume substantial memory, processing, and bandwidth, that will eventually strain the mobile hardware and can cause battery drain and heat on older devices. Reliance on cloud services introduces latency, storage cost, and a dependence on connectivity that reduces usefulness in low-bandwidth settings. The very most current implementation also lacks offline OCR and language processing, and it supports a limited set of document formats. These specific constraints do not weaken the core retrieval results, but they define the engineering work required before the full pipeline can run comfortably on a mobile phone.

5. Conclusion and Future Work

This paper presented here Intelli-Docs, a personal document assistant that unifies multimodal ingestion, semantic indexing, and retrieval-augmented generation(RAG) so that users can find and question their records through natural language. By classifying each and every request and routing it through a text pipeline, an image pipeline, or a conversational path, and by fusing OCR text, BLIP captions, and CLIP embeddings into a common representation, the system extracts both born-digital and image-based documents with a single

mechanism. On image-text benchmarks the retrieval component has reached Recall@5 of 92% and mean average precision of 0.89 while keeping mean query latency below 3 s, and the easy and clear separation between positive and negative similarity scores accounts for the most reliable top-of-list ranking.

Several directions would strengthen both the system and its evaluation. The most immediate one is a quantitative comparison against a keyword-retrieval baseline on a dedicated personal-document corpus, which would measure the benefit of semantic multimodal retrieval directly for the target use case only. On the systems side, on-device model compression such as 8-bit quantization would reduce memory use, and energy-aware scheduling would limit battery impact; adding offline OCR and language processing and broadening format support would improve accessibility where connectivity is limited. Finally, stronger privacy controls, including multi-factor authentication and explicit data-retention policies, and the shift toward edge computing for latency-sensitive steps would move the design closer to a practical, privacy-preserving product.

On the more broader view, the study shows that the components needed for practical personal document management already exist and can be combined coherently: dense retrieval and retrieval-augmented generation supply grounded answers over a private corpus, and vision-language models make image documents first-class members of that corpus rather than opaque attachments. The contribution of IntelliDocs is the integration of these components behind a single natural-language interface and the empirical characterization of the resulting accuracy-cost trade-off. Closing the remained gap between the reported component-level results and a fully validated, on-device product is primarily an engineering effort, guided by the priorities set out above the baseline.

Acknowledgements

We thank Asst. Prof. Tantra Nath Jha and Asst. Prof. Pukar Karki for their guidance, suggestions, mentorship, and feedback throughout this research, and the Department of Electronics and Computer Engineering, IOE Purwanchal Campus, for funding and resources. We also thank the researchers whose prior work informed this study.

References

Bhuvaneswari, C., Pokhariya, H. S., Yarde, P., Vekariya, V., Patil, H., & Natrayan, L. (2024). Implementing AI-powered chatbots in agriculture for optimization and efficiency. In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)* (pp. 1–7). IEEE.

de Barcelos Silva, A., Gomes, M. M., da Costa, C. A., da Rosa Righi, R., Barbosa, J. L. V., Pessin, G., De Doncker, G., & Federizzi, G. (2020). Intelligent personal assistants: A systematic literature review. *Expert Systems with Applications*, *147*, 113193. https://doi.org/10.1016/j.eswa.2020.113193

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186). Association for Computational Linguistics.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2023). *Retrieval-augmented generation for large language models: A survey* (arXiv:2312.10997). arXiv. https://doi.org/10.48550/arXiv.2312.10997

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

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Li, J., Li, D., Xiong, C., & Hoi, S. (2022). BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning* (PMLR Vol. 162, pp. 12888–12900). PMLR.

Mei, L., Mo, S., Yang, Z., & Chen, C. (2025). *A survey of multimodal retrieval-augmented generation* (arXiv:2504.08748). arXiv. https://doi.org/10.48550/arXiv.2504.08748

Ling, X., Gao, M., & Wang, D. (2020). Intelligent document processing based on RPA and machine learning. In *2020 Chinese Automation Congress (CAC)* (pp. 1349–1353). IEEE.

Nandi, K., & Sathya, S. S. (2024). Visual document understanding: A comparative review of modern methods. In *International Conference on Computer Vision and Image Processing* (pp. 411–427). Springer.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning* (PMLR Vol. 139, pp. 8748–8763). PMLR.

Rusli, F. M., Adhiguna, K. A., & Irawan, H. (2021). Indonesian ID card extractor using optical character recognition and natural language post-processing. In *2021 9th International Conference on Information and Communication Technology (ICoICT)* (pp. 621–626). IEEE.

Smith, R. (2007). An overview of the Tesseract OCR engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)* (Vol. 2, pp. 629–633). IEEE.

Whittaker, S. (2011). Personal information management: From information consumption to curation. *Annual Review of Information Science and Technology*, *45*(1), 1–62. https://doi.org/10.1002/aris.2011.1440450108

Zierau, N., Engel, C., Söllner, M., & Leimeister, J. M. (2020). Trust in smart personal assistants: A systematic literature review and development of a research agenda. In *Proceedings der Internationalen Tagung Wirtschaftsinformatik (WI)*. Potsdam, Germany.