Year

2024-25

Project

No. 002

Multimodal Information Retrieval Using Text and Image

Multimodal Information Retrieval Using Text and Image

Aayush Kumar

Avaneesh Sundararajan

C. Nikhil Karthik

Nithish Chouti

Udayini Vedantham

Department of Computer Science & Engineering

Indian Institute of Information Technology Dharwad,

Karnataka, India

Odd Semester 2024-25

Declaration

I declare that this report represents my ideas in my own words. Where others' ideas and words have been included, I have adequately cited and referenced the original source. I declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the source which has thus not been properly cited or from whom proper permission has not been taken when needed.

Aayush Kumar (Roll No. 21BCS001)
Avaneesh Sundararajan (Roll No. 21BCS020)
C. Nikhil Karthik (Roll No. 21BCS024)
Nithish Chouti (Roll No. 21BCS074)
Udayini Vedantham (Roll No. 21BCS130)

Date: 18th November, 2024
Place: IIIT Dharwad

Abstract

In recent years, the field of information retrieval has advanced considerably with the integration of deep learning methods, particularly for multimodal retrieval systems that support both text and image queries. This research presents the development and evaluation of a novel multimodal information retrieval system that effectively processes both text and image-based queries. Built on Elasticsearch's indexing capabilities, the system integrates Sentence Transformers for text analysis and Vision Transformers (ViT) for image processing, enabling users to search using text, images, or a combination of both. Through rigorous testing using research papers from IEEE, Springer, and CVPR conferences, our comparative analysis demonstrated the system's effectiveness across different modalities. The Sentence Transformer model (all-MiniLM-L6-v2) achieved superior text retrieval performance with a precision of 0.688, recall of 0.888, and F1 score of 0.748, outperforming traditional BERT models. For image-based queries, the Vision Transformer emerged as the most effective among tested models including MobileNetV2, ResNet50, and EfficientNetB7, achieving a precision of 0.55, recall of 0.75, and F1 score of 0.61. The system's multimodal capabilities demonstrated balanced performance with mean precision of 0.654, recall of 0.745, and F1 score of 0.681, particularly excelling in scenarios requiring both textual and visual context. Implemented using FastAPI for asynchronous processing, this system offers practical applications in academic research, e-commerce platforms, and multimedia archives where dual-modality search capabilities are increasingly essential.

Index Terms: *Multimodal Information Retrieval, Deep Learning, Elasticsearch, Sentence Transformers, Vision Transformers.*

1 Introduction

The course of information retrieval has made a huge stride, with deep learning permitting the systems to obtain relevant documents from a query input. Most traditional search engines impose a great weight on deducing queries in text form while being unable to completely capture the real intent of the user in multimedia- or multimodal-related input retrieval. In a bid to address this gap, the present project will delve into the development process of a multimodal information retrieval system whereby the user could query by using text and images together or by each method separately. Such applications would serve the purpose of multimedia search engines, e-commerce platforms, and academic research, wherein the detailed information could be based on images or the combination of images and text explaining the given image. In applications such as e-commerce, where users often search for products using both descriptions and images, a system that supports multimodal queries can significantly improve search relevancy, enhancing user satisfaction and conversion rates.

This system is based on a powerful Elasticsearch engine, among those providing a full-text search supporting their use in similarity searches based on vector spaces. This project has introduced a technique wherein the vector embeddings of text and image data are stored in Elasticsearch, thus ensuring efficient retrieval of documents relevant semantically. Such vector embeddings are made on pre-trained deep learning models capable of encoding text and images into vector representations, retaining their semantic meanings. The flat surface of embodied knowledge is then stored in Elasticsearch, therefore supporting searches based on similarity for both the modalities.

This project uses Sentence Transformers for embedding textual data and a Vision Transformer (ViT) model to process visual components of images, designed to capture the refined sense of meaning both in the text and visual content respectively. Embedded into the information retrieval pipeline itself, these models support seamless multimodal queries, whereby the relevant documents can be retrieved by the user through any combination of text, image, or both.

In general, this system is an archetype of new advancement in terms of information retrieval,

where deep learning techniques are going to complement Elasticsearch in actively and dynamically constructing multimodal queries in a broad range with efficient handling.

2 Related Works

The evolution of information retrieval systems has been revolutionized by transformer-based architectures. Vaswani et al.'s seminal work "Attention is All You Need" [1] introduced the transformer architecture, establishing a new paradigm in natural language processing through self-attention mechanisms. This breakthrough enabled models to capture complex, long-range dependencies and semantic meanings in text data more effectively than previous approaches.

Building on transformer fundamentals, BERT [2] emerged as a powerful language model, while Sentence-BERT [3] refined this approach specifically for semantic similarity tasks. Reimers and Gurevych demonstrated that Sentence-BERT, through fine-tuning on sentence pairs, produces embeddings that capture nuanced textual meanings—crucial for high-quality information retrieval. The DPR (Dense Passage Retrieval) system [4] further advanced text retrieval by introducing dense embeddings for both queries and passages, achieving significant improvements over traditional sparse retrieval methods.

In computer vision, Vision Transformers (ViTs) [5] marked a paradigm shift from convolutional architectures. Dosovitskiy et al. demonstrated ViTs' effectiveness in extracting high-level semantic features from images, particularly valuable for complex visual information processing. CLIP [6] further revolutionized visual learning by training on massive image-text pairs, enabling zero-shot transfer capabilities and robust visual feature extraction.

Recent research has focused intensively on bridging the gap between textual and visual modalities. The UNITER framework [7] proposed a unified architecture for image-text representation learning, introducing novel pre-training tasks that significantly improved cross-modal understanding. Similarly, UnicoderVL [8] demonstrated the effectiveness of a universal encoder for both visual and linguistic inputs, achieving state-of-the-art results in image-text retrieval tasks.

Oscar [9] introduced an object-semantics aligned approach, showing how object-level under-

standing can enhance multimodal retrieval accuracy. Their work demonstrated that incorporating object detection signals into the pre-training process can significantly improve cross-modal alignment.

Large-scale industrial implementations have provided valuable insights into practical deployment challenges. The M6 model [10] showcased successful handling of multiple modalities at scale, while ALIGN [11] demonstrated effective training on billion-scale image-text pairs. These implementations highlighted both the possibilities and challenges of deploying multimodal systems in production environments.

LightningDOT [12] addressed the crucial aspect of efficiency in multimodal retrieval, introducing lightweight cross-modal encoding methods that maintain accuracy while significantly improving speed. This work is particularly relevant for real-world applications where response time is critical.

The evolution of vector search capabilities in systems like Elasticsearch [13] has been crucial for practical implementations of multimodal retrieval systems. Modern vector similarity search methods enable efficient indexing and retrieval of semantic embeddings, supporting both text and image modalities effectively.

3 System Architecture

The architecture of the platform is aimed at a flexible and robust multimodal information retrieval system, leveraging both deep learning models and efficient query processing using Elasticsearch. The architecture boils down to three main components: Text Query Processing, Image Query Processing, and Combined Query Processing; hence, the search experience becomes smooth across multiple data types.

3.1 Text Query Processing

The textual inputs are encoded using a pre-trained Sentence Transformer model, namely ‘all-MiniLM-L6-v2‘, to textual queries represented in vector embeddings. The embeddings allow

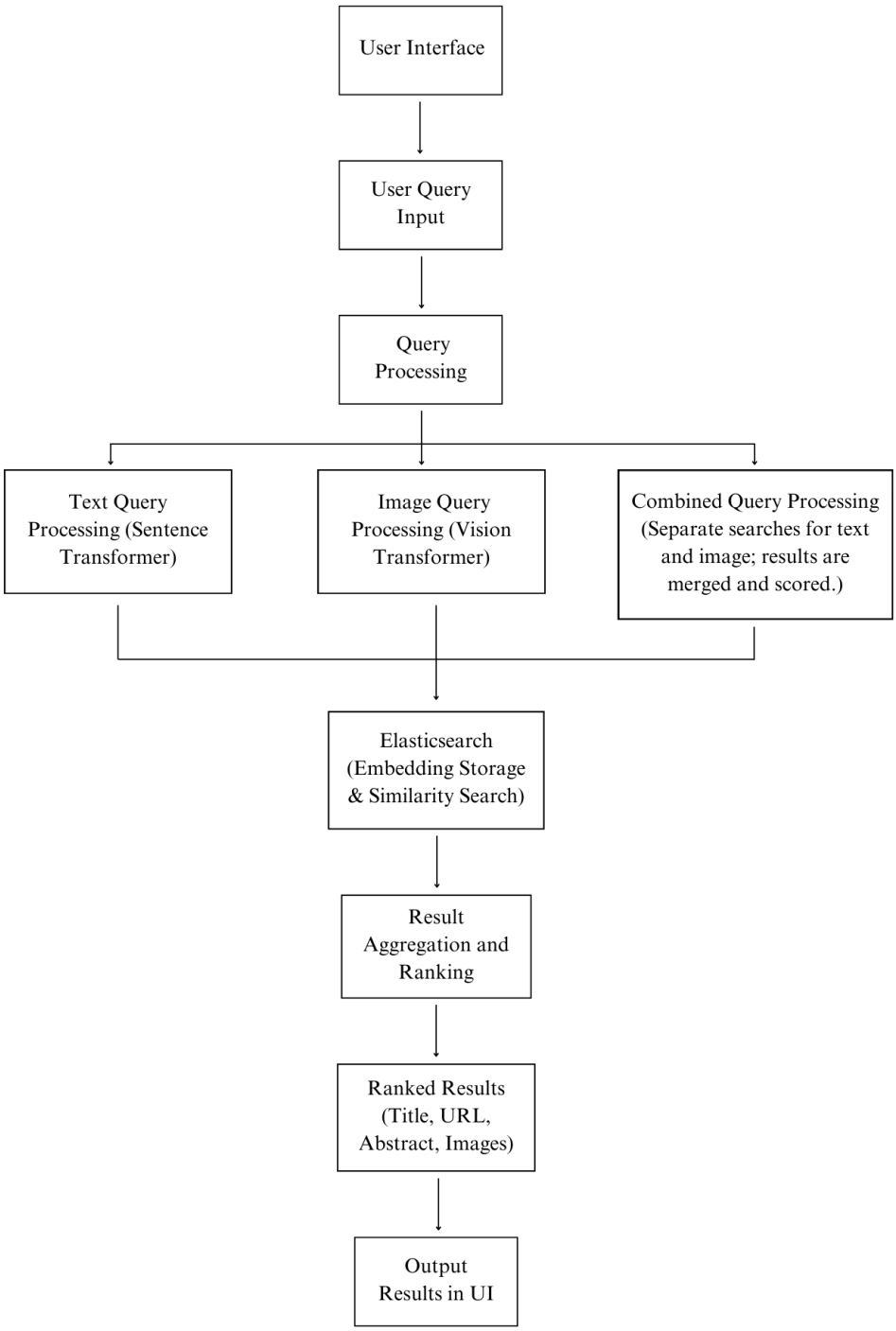

Figure 1: Flow diagram illustrating the architecture of the multimodal information retrieval system.

the system to capture the semantic essence of the query, thereby permitting meaningful similarity comparisons with stored text embeddings in the Elasticsearch database.

3.2 Image Query Processing

Image input is processed with the help of the Vision Transformer (ViT) model; images flow through ViT to obtain embeddings capable of representing the visual content of the input. With the ViT features, the system effectively encodes and retrieves relevant image information stored in Elasticsearch.

3.3 Combined Query Processing

When users provide both text and images in their search, our system processes each type of input separately at first. It creates unique digital representations (embeddings) for the text and image components independently. These separate representations are then brought together into a single unified search pattern. By merging the text and visual information this way, our system can thoroughly search through both types of content at once, making the search results more accurate and relevant to what the user is looking for.

FastAPI provides REST API functionality with an asynchronous structure suitable to handle query requests. The architecture allows asynchronous processing of queries, which move from input reception through the steps of embedding generation, querying in Elasticsearch, to result aggregation. FastAPI really is the core point of interaction for the user to the Elasticsearch backend, where the complete embeddings get stored and processed.

4 Methodology

This project aims at embedding deep learning methods both for text and image processing, supplemented by the use of Elasticsearch to manage vector embeddings for quick storage and retrieval. It passes through several major steps.

4.1 Model Selection and Training

1. Text Embedding Model:

The project sets forth Sentence Transformers to convert text queries into vector embeddings. The specific model utilized is `all-MiniLM-L6-v2`, which has been proven efficient for generating compact embeddings that are semantically similar. This model was pre-trained on extensive data, enabling it to accurately capture and encode the details contained in textual data.

2. Image Embedding Model:

The Vision Transformer (ViT) model is utilized to process image input. ViT extracts high-order semantic features from images by treating them as sequences of patches, allowing for transformer-based modeling of images. The model is pre-trained on large-scale image datasets, providing a strong foundation for generating embeddings that capture visual content.

4.2 Data Preparation

1. Image and Text Preprocessing:

The ViT model receives images resized to 224x224 pixels, normalized prior to processing. Text queries are tokenized and converted into embeddings using the Sentence Transformer model. This preprocessing ensures compatibility between both modalities and allows for effective cross-modal comparison during retrieval.

2. Embedding Generation:

Upon receipt of a query, embeddings are generated for both image and text inputs. Text embeddings are obtained from the Sentence Transformer model, while image embeddings are derived using the ViT model. Each embedding is stored as a NumPy array for efficient similarity computation during the search phase.

4.3 Integrating into Elasticsearch

1. Indexing:

The embeddings generated from both modalities are stored in Elasticsearch indices. Each document in the index includes both text and image embeddings, along with metadata such as title, URLs, and abstracts, facilitating fast retrieval based on vector similarity.

2. Similarity Search:

Upon receiving a query, the system performs a cosine similarity search against the embeddings stored in Elasticsearch to find the most relevant documents. The similarity scores are used to rank the documents, prioritizing those most semantically aligned with the query input.

4.4 Processing Query

Text Query:

When users search with text, our system converts their words into a special format using Sentence Transformer technology. It then uses Elasticsearch to find documents that best match the search terms, ranking them based on how closely they align with the user's query.

Image Query:

For image-based searches, we use Vision Transformer technology to understand the visual elements of the uploaded image. The system then searches through our database to find visually similar content, matching things like patterns, objects, and overall composition.

Combined Query:

Users can also search with both text and images together. In these cases, our system processes both inputs separately and then combines the results. We give extra importance to documents that match both the text description and visual elements, helping users find exactly what they're looking for.

4.5 Building an API

To make all this work smoothly, we built our search interface using FastAPI. This modern tool lets us handle many user searches at the same time without slowing down. When someone searches, our system quickly processes their request and returns the results in an organized format that's easy to understand. The system can handle multiple users searching simultaneously, making it efficient and user-friendly.

5 Implementation Details

5.1 Data Collection/Dataset Creation

The dataset for this project was meticulously curated from a selection of research papers published in prestigious conferences, including IEEE, Springer, and CVPR. The data collection process involved the following steps:

- **Content Collection:** Research papers were sourced online, focusing on those containing rich visual and textual content. Each paper's embedded images were extracted and systematically organized to facilitate effective retrieval.
- **Folder Structure:** The collected dataset was structured into a well-organized directory hierarchy to ensure easy access and indexing in Elasticsearch. Each paper was stored in a dedicated folder, organized as mentioned in Fig. 2.

  The `info.txt` files contain critical information such as the paper title, URL, and abstract, enabling comprehensive indexing and facilitating effective retrieval capabilities.

5.2 Libraries and Frameworks Used

The implementation leverages several key libraries and frameworks that enhance functionality and streamline the development process:

- **FastAPI:** A modern, high-performance web framework for building APIs with Python, enabling the creation of asynchronous endpoints that efficiently handle user queries.

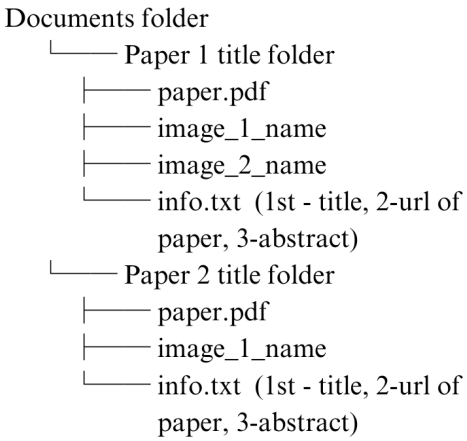

Figure 2: Dataset Document Structure

- **Torch and Torchvision:** These libraries provide robust tools for developing and deploying deep learning models. Torch serves as the core library, while Torchvision offers pre-trained models and image transformations essential for processing visual data.

- **Sentence Transformers:** This specialized library is utilized for generating high-quality text embeddings. It includes various pre-trained models, notably the `all-MiniLM-L6-v2`, which is employed for processing text queries in this project.

- **Elasticsearch:** An open-source search and analytics engine, Elasticsearch is utilized for storing, indexing, and querying the vector embeddings generated from both text and image data, thus enabling efficient similarity searches.

5.3 Code Explanation

The implementation consists of several critical Python files, each serving a distinct role within the system architecture:

- **main.py:** This file establishes the FastAPI application and defines the API endpoints for text, image, and combined queries. It facilitates the connection to the Elasticsearch instance and orchestrates the data flow between the frontend and backend, effectively managing user requests and returning relevant results.

- **Image_Functions.py:** This module encompasses helper functions dedicated to image processing. It includes functionalities for image loading, resizing, normalization, and embedding generation using the Vision Transformer model.
- **Text_Functions.py:** Similar to the image functions, this module contains helper functions tailored for processing textual input. It addresses tasks such as tokenization, embedding generation using Sentence Transformers, and any necessary preprocessing to ensure compatibility with the retrieval system.

5.4 Query Flow

The system's query processing flow is designed to efficiently handle user requests, as illustrated below:

1. **Receiving a Query:** Users submit a query through the FastAPI interface, which can comprise either a text input, an image upload, or both modalities combined.
2. **Embedding Generation:** Based on the query type, the system invokes the appropriate embedding generation function:
    - For text queries, the `Text_Functions.py` module generates a vector embedding from the provided text input.
    - For image queries, the `Image_Functions.py` module produces an embedding from the uploaded image.
    - In the case of combined queries, embeddings for both input types are generated concurrently.
3. **Similarity-Based Searching:** The generated embeddings are transmitted to Elasticsearch, which performs a similarity search against the indexed embeddings. The search process relies on cosine similarity to calculate relevance scores for all documents.
4. **Returning Ranked Results:** Once the system executes the similarity-based search using Elasticsearch, it aggregates the results and ranks the documents according to their similarity scores, with higher scores indicating greater relevance to the user's query. The final

structured response includes key information for each document, such as titles, URLs for accessing the full papers, abstracts summarizing the main contributions, and any associated images extracted from the documents. This organized presentation enhances the user experience, allowing for efficient access to pertinent information, while also enabling users to refine their queries for iterative searching, thus facilitating a deeper exploration of the topic.

5.5 Challenges with Hardware Resources

Modelling for multimodal query processing continuously needs vast parallel computational resources. Among the challenges faced during implementation, by far, the most pressing was that of high memory consumption, especially due to the ViT model occupying around 1GB of memory. System slowdowns commonly ensued thereafter-especially during extensive testing or processing of parallel queries. Apart from that, each modality in Elasticsearch incurs a memory overhead of generating and storing embeddings that further aggravate the situation.To deal with such problems-an optimized setup of deploying the models to make adequate use of available resources is required and always keeping in mind hardware accelerators such as GPUs or TPUs to provide extent.

6 User Interface Application

Our search engine interface leverages Next.js for robust server-side rendering and efficient routing, creating a seamless user experience. The app name is *MIRAX: "Multifaceted Intelligent Retrieval and Analytics eXperience"*. The UI components are built using shadcn's accessible component library, providing a modern and consistent design system. Tailwind CSS enables rapid styling with utility-first classes, ensuring responsive design across devices. Axios handles API requests reliably, managing search queries and results with clean error handling. The interface features an intuitive search bar, real-time suggestions, paginated results, and a clean layout that prioritizes readability while maintaining fast load times and smooth interactions.

Figure 3: UI Interface

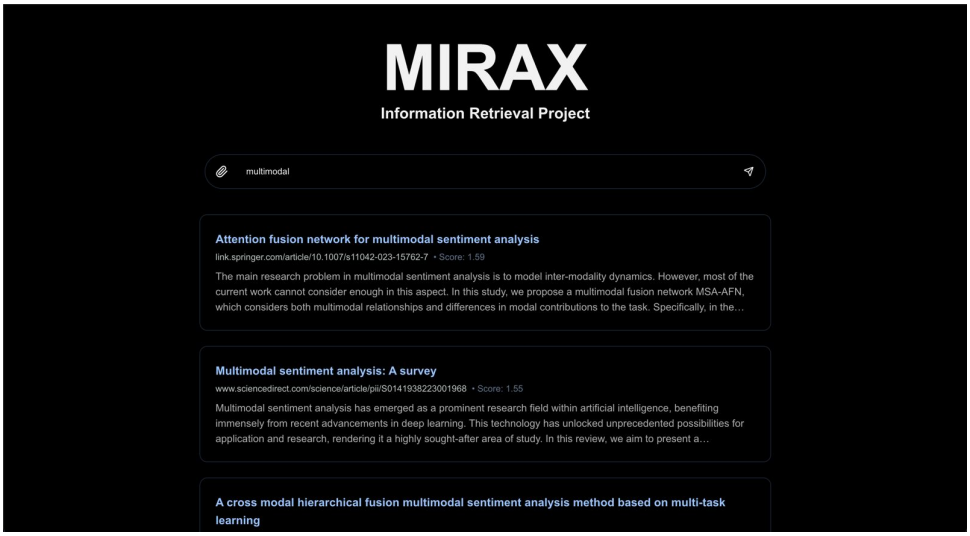

Figure 4: Only Text Based Information Retrieval

Figure 5: Only Image Based Information Retrieval

Figure 6: Combined Query Results

7 Results

In this project, we benchmarked different models to evaluate their effectiveness in retrieving information based on either text or image queries. Each model's performance was measured on precision, recall, and F1 score, allowing us to compare their accuracy and relevance in the context of information retrieval.

7.1 Text-Only Results

For text-based retrieval, the Sentence Transformer (`all-MiniLM-L6-v2`) performed noticeably better than BERT across all metrics. The Sentence Transformer achieved a precision of 0.688, a high recall of 0.888, and an F1 score of 0.748. This balanced performance indicates its suitability for accurately retrieving relevant documents from complex text-based queries. In contrast, the BERT model, while effective, yielded lower scores, with an F1 score of 0.368. Given the Sentence Transformer's superior performance, it was selected for the final system setup.

Model	**Precision**	**Recall**	**F1 Score**
BERT	0.355	0.388	0.368
Sentence Transformer	0.688	0.888	0.748

Table 1: Performance of Text-Only Models

7.2 Image-Only Results

For image-based retrieval, we evaluated several models, including MobileNetV2, ResNet50, EfficientNetB7, and Vision Transformer (ViT). Among these, the Vision Transformer demonstrated the best overall performance with a precision of 0.55, a recall of 0.75, and an F1 score of 0.61. This model's capability to capture high-order semantic features from images proved beneficial for image-based retrieval tasks, making it the preferred choice for embedding image data.

Model	Precision	Recall	F1 Score
MobileNetV2	0.45	0.65	0.51
ResNet50	0.5	0.7	0.559
EfficientNetB7	0.433	0.6	0.48
Vision Transformer	0.55	0.75	0.61

Table 2: Performance of Image-Only Models

7.3 Multimodal Retrieval Performance

Our system's ability to handle both text and image queries was especially useful in retrieving information relevant to both types of input. For these combined queries, we generated embeddings separately for text and image, then merged the results, giving higher weight to documents that matched both. This combination of text and image embeddings led to more precise and contextually relevant results, especially when the query relied on both text and visual details.

In terms of overall performance, our multimodal retrieval system achieved a mean precision of 0.654, a mean recall of 0.745, and a mean F1 score of 0.681 on the test queries. These scores highlight that the system effectively retrieved relevant information across both text and image data. This balanced performance shows that combining text and image retrieval can be powerful, providing stronger, more accurate results in scenarios that involve mixed media content.

8 Conclusion

In this project, we developed a system designed to search through documents using both text and image inputs. Leveraging Elasticsearch as the core search engine, we explored various deep learning models to process text and image data effectively. For text processing, we found that Sentence Transformers outperformed standard BERT models, providing strong results when retrieving information from textual content. For image processing, we evaluated models such as MobileNetV2 and ResNet50, with the Vision Transformer (ViT) demonstrating the highest

performance.

Our experimental results indicate that the system performs effectively for both text-based and image-based retrieval tasks. For text-only queries, the system achieved a precision of approximately 69% and a recall of 89%. For image-only queries, performance was slightly lower, with a precision of around 55% and a recall of 75%. When combining text and image retrieval, the system successfully retrieved relevant documents, achieving an overall mean precision of 0.654, a mean recall of 0.745, and a mean F1 score of 0.681 on the test queries. These metrics highlight the advantages of a multimodal retrieval approach that integrates both text and image information.

The system was implemented using FastAPI, allowing efficient handling of concurrent search requests and ensuring flexibility for future modifications. This modular design enables the integration of additional features or adjustments without the need for extensive redevelopment, making it adaptable for applications such as image-based search engines or e-commerce platforms where users often search using both images and descriptive text.

References

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

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

[3] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

[4] S. Min L. Wu S. Edunov D. Chen V. Karpukhin, B. Oguz and W. Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

[5] A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[6] A. Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[7] Y. Chen et al. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[8] G. Li et al. Unicodervl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020.

[9] X. Li et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[10] J. Lin et al. M6: A chinese multimodal pretrainer. In *KDD*, 2021.

[11] C. Jia et al. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[12] Y. Sun et al. Lightningdot: Pre-training visual-semantic embeddings for real-time image-text retrieval. In *NAACL*, 2021.

[13] C. Gormley and Z. Tong. *Elasticsearch: The Definitive Guide*. O'Reilly Media, 2015.