I.R.I.S.: Image Retrieval & Intelligent Search

Punyam Singh
2110110842
ps232@snu.edu.in

Rahul Jayaram
2110110410
rj712@snu.edu.in

***Abstract*—This project introduces IRIS, a cutting-edge image tagging and retrieval system integrated into a web application, the IRIS Smart Gallery. Users can upload images and later search for them using free-text queries. Iris leverages the BLIP (Bootstrapping Language-Image Pretraining) model to generate descriptive captions, which are processed to extract meaningful tags. These tags are further refined through semantic and cosine similarity measures, allowing Iris to rank tag relevance effectively and improve retrieval accuracy. A key feature of Iris is the customizable relevance threshold, enabling users to control the precision of search results based on their needs. Our system has been rigorously evaluated using both standard retrieval metrics—precision, recall, mean average precision (MAP), and Discounted Cumulative Gain (DCG)—and a human-tagged Kaggle dataset for reliable performance benchmarking. The results confirm that Iris provides a scalable, high-quality solution for categorizing and retrieving images from large datasets. Our primary contributions include the implementation of BLIP for image captioning and tagging, the application of similarity metrics to enhance retrieval precision, and an extensive evaluation demonstrating the system's effectiveness with both metric-based and human-generated data.**

I. Introduction

A. Background

With the exponential growth of image data on the internet and across various platforms, effective image tagging and retrieval have become essential for numerous applications, such as content-based image search engines, social media platforms, and digital asset management systems. The challenge lies in the automatic generation of relevant and context-aware tags for images, enabling more efficient searches and a better user experience. Traditional methods for image retrieval often relied on manual tagging or shallow machine learning techniques, which are limited in terms of scalability, flexibility, and accuracy. As the volume of image data continues to grow, automated, intelligent solutions are increasingly needed.

B. Importance of the Topic

The need for automated image tagging is more critical than ever. In today's digital age, where vast amounts of untagged images are uploaded daily, the ability to generate accurate, context-aware tags is key to improving the functionality of image search systems. This capability becomes even more important in scenarios where users need to quickly find specific images from large personal or shared galleries. For instance, imagine searching for a soft copy of your ID proof within a cluttered gallery—this is a common problem faced by users, and it illustrates the real-world necessity for efficient, accurate image retrieval systems. IRIS provides a solution to such challenges by focusing on relevance, helping users find the right image quickly and efficiently.

C. Motivations

The main motivation behind IRIS is to enhance the process of automated image retrieval by focusing on user-specific relevance needs. Today, users often face challenges when trying to locate specific images from large, unorganized galleries. The inability to efficiently search for images based on meaningful tags leads to frustration, as seen in situations like searching for personal documents or images from a family event. By enabling users to search images using free-text queries and allowing them to fine-tune the relevance threshold for image retrieval, IRIS makes it easier to find exactly what is needed. This approach addresses the growing need for intelligent systems that can handle large volumes of image data while prioritizing user relevance and customization.

D. Contributions

Our contributions can be summarized as follows:

- We introduce IRIS, an image tagging and retrieval system that utilizes the BLIP (Bootstrapping Language-Image Pretraining) model for caption generation and tagging. This allows the automatic generation of meaningful and contextually accurate tags for images [1].
- We focus on the user's relevance needs by incorporating customizable relevance thresholds, enabling users to control the precision of search results, ensuring that they retrieve the most relevant images.
- We apply semantic and cosine similarity measures to rank the relevance of tags, improving the accuracy of image retrieval based on user queries.
- We create a user-friendly web application, the IRIS Smart Gallery, where users can upload their images and search them using free-text queries.
- We evaluate the performance of our system using standard retrieval metrics, including precision, recall, mean average precision (MAP), and Discounted Cumulative Gain (DCG), demonstrating its effectiveness in large-scale image datasets. [2]
- We leverage knowledge from information retrieval techniques to build a scalable and efficient system that can be applied in real-world contexts, such as personal image galleries, digital asset management, and content-based search engines.

II. Literature Review/Related Work

Image tagging and retrieval have been extensively explored in recent years, with advancements in deep learning models significantly improving the accuracy and efficiency of these tasks. This section discusses key works in the field that informed the development of the IRIS image tagging and retrieval system.

One notable approach is **BLIP** (Bootstrapping Language-Image Pretraining), which effectively bridges the gap between vision and language tasks. BLIP is a vision-language pretraining framework that achieves state-of-the-art performance across a variety of tasks, including image captioning and image-text retrieval [1]. What makes BLIP unique is its ability to handle both understanding-based and generation-based tasks, offering a flexible solution that adapts well to diverse applications. Unlike other models that rely on noisy image-text pairs from the web, BLIP improves the quality of training data by bootstrapping captions and filtering out noisy ones. This results in highly accurate captions that enhance image-text retrieval performance, achieving a 2.7% increase in recall and a 2.8% improvement in CIDEr scores on various benchmarks [1]. The robust nature of BLIP's architecture, which allows it to generate meaningful captions for images, makes it a suitable choice for IRIS, where the generation of relevant tags is a crucial component for accurate image retrieval.

Other research, such as the **Recognize Anything Model** (RAM), also explores large-scale image-text pairings for annotation-free image tagging. RAM introduces a novel approach where tags are generated via automatic text semantic parsing and image captioning tasks [3]. While RAM has shown impressive zero-shot performance and outperforms models like CLIP and BLIP in certain benchmarks, it primarily focuses on general tagging without incorporating sophisticated retrieval mechanisms. In contrast, IRIS goes a step further by not just generating tags but also applying advanced information retrieval techniques to ensure that the tags are ranked according to their relevance, significantly enhancing the search experience.

In the realm of object detection, the Fast R-CNN approach and its improvement, Faster R-CNN, introduced Region Proposal Networks (RPN) to improve object detection efficiency [4]. While this work focuses on object localization and recognition, the underlying principle of combining high-quality feature extraction with region-specific proposals inspired IRIS to consider image tagging as a first step in the broader image retrieval process, ensuring that the relevance of tags is emphasized before retrieval.

Similarly, the Zero-Shot Image Tagging approach explores the relationship between images and words by using vector offsets to identify relevant tags [5]. This method leverages word vectors to estimate image-tag relevance but lacks the context-driven ranking that IRIS introduces. While zero-shot tagging can identify broad categories, it does not fully address the challenge of relevance ranking, which is crucial for creating an effective retrieval system. IRIS improves upon this by incorporating semantic and cosine similarity measures to rank tags, ensuring that users receive the most relevant results in response to their queries.

The novelty of IRIS lies in its ability to integrate advanced image captioning and tagging with a sophisticated image retrieval model. While BLIP and other models excel at generating captions or tags, IRIS uniquely applies information retrieval techniques to rank and filter these tags, allowing for a highly customizable and user-focused search experience. By enabling users to adjust the relevance threshold, IRIS ensures that search results are tailored to individual preferences, something not commonly seen in traditional image tagging systems. Moreover, the use of evaluation metrics such as precision, recall, MAP, and DCG further strengthens the reliability and effectiveness of the system, providing a comprehensive solution for scalable and accurate image retrieval.

III. Objective

The primary objective of this project is to design and develop *IRIS* [6], an advanced image tagging and retrieval system that addresses the increasing need for efficient and accurate image categorization in large-scale datasets. The system leverages the state-of-the-art **BLIP** (Bootstrapping Language-Image Pretraining) model for automatic caption generation, which forms the basis for deriving relevant tags. These tags are further refined through the use of **semantic similarity** and **cosine similarity** techniques to ensure the retrieval of highly relevant images based on textual queries.

A key goal of *IRIS* is to not only provide accurate image tagging but also to enable **flexible image search** that takes into account user-defined relevance thresholds. This allows users to customize the results they receive, making the system adaptive to varying needs and preferences. The development of this system aims to bridge the gap between general image retrieval and specific user requirements, ensuring that users can quickly and easily locate relevant images based on free-text queries.

To achieve these goals, *IRIS* incorporates a range of functionalities, such as:

- **Automatic Tag Generation**: Using the BLIP model to generate descriptive captions from images.
- **Enhanced Image Retrieval**: Applying similarity measures to refine and rank the relevance of image tags and enhance search accuracy.
- **User Customization**: Allowing users to set a relevance threshold to control the quality of results.
- **Web Application Interface**: Developing a user-friendly front-end that enables users to upload images, search for them using textual queries, and adjust the relevance threshold.

Additionally, the performance of *IRIS* is evaluated using standard **information retrieval metrics** such as **precision**, **recall**, **mean average precision**, and **Discounted Cumulative Gain** (DCG) [2] to ensure that the system meets high-quality standards in terms of image retrieval accuracy and user satisfaction.

Through these objectives, *IRIS* aims to provide a powerful, scalable solution for automatic image tagging and retrieval, suitable for real-world applications in content-based image retrieval, digital asset management, and other domains where efficient image search capabilities are essential.

IV. Proposed Model

The proposed model, IRIS, follows a multi-step process to generate image tags and perform efficient image retrieval based on user queries. The process begins with the use of the BLIP (Bootstrapping Language-Image Pretraining) model for automatic caption generation, which generates descriptive captions for images. These captions are then processed to extract relevant tags, which are stored in a structured format (JSON file) for easy retrieval.

The tags associated with each image are used as metadata to facilitate efficient image retrieval based on user queries. When a user inputs a query, the system retrieves relevant images by matching the query with the tags, leveraging cosine similarity to rank the images in order of relevance.

For end-to-end testing and deployment, we utilized **Streamlit**, a framework that allows us to quickly create interactive web applications for machine learning models. Streamlit was used to deploy the IRIS model, providing a user-friendly interface for interacting with the model, uploading queries, and retrieving images. Additionally, we used **Supabase** to manage image storage in Supabase buckets, making it easy to store, organize, and retrieve large image datasets in the cloud.

The following diagram illustrates the key steps in our approach:

In this system:

1) **BLIP Model**: Used to generate captions for images.
2) **Tag Extraction**: Captions are processed to extract relevant tags.
3) **Tag Storage**: The tags are stored in a JSON file for efficient retrieval.
4) **Supabase**: Supabase buckets are used to store and manage images.
5) **Streamlit**: A web application is deployed using Streamlit to interact with the model and perform image retrieval.

V. Methodology

The methodology employed in this project consists of two primary phases: **1) Image Tag Generation** and **2) Query-based Image Retrieval**. The detailed workflow for each phase is outlined below.

A. Image Tag Generation

The first phase involves generating relevant tags for each image, which facilitates efficient retrieval based on textual queries.

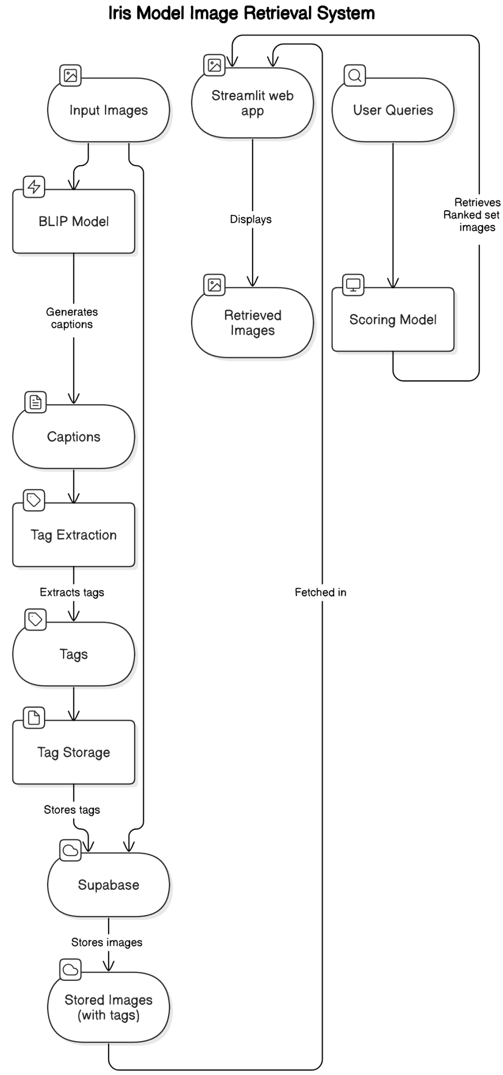

Fig. 1. Overview of the IRIS Model Approach

- *BLIP Model for Caption Generation*: The process begins with the application of the BLIP (Bootstrapping Language-Image Pretraining) model, a state-of-the-art Vision-Language Pretraining (VLP) model. BLIP generates comprehensive captions by understanding the content of images, leveraging a pre-trained vision-language model to produce descriptive, contextually relevant captions.
- *Tag Extraction*: Following caption generation, the caption text is parsed to extract meaningful tags. In our approach, the caption is split into individual words, and a set of tags is derived from these words. This procedure is repeated for each image in the dataset, which is stored in a predefined directory.
- *JSON Storage*: The extracted tags, along with their corresponding image filenames, are stored in a structured JSON format. This storage method allows for efficient access and manipulation of the data when performing image queries.

B. Query-based Image Retrieval

The second phase focuses on retrieving relevant images based on user-provided queries. The retrieval process is designed to assess both literal and semantic similarities between the input query and the image tags.

- *Input Query*: Users can submit a query, either as a textual description or a set of keywords that represent the image they wish to retrieve.
- *Literal Similarity Calculation*: To assess the relevance of each image to the input query, we calculate the **literal similarity** between the query and the tags of each image. This is done using the **TF-IDF** (Term Frequency-Inverse Document Frequency) technique, which quantifies the importance of each word in both the query and the tags. The query and tags are vectorized, and the cosine similarity is then computed between the query vector ($\mathbf{q}$) and the tag vector ($\mathbf{t}$):

$$\text{Literal Similarity} = 1 - \cos(\mathbf{q}, \mathbf{t})$$

- *Semantic Similarity Calculation*: To further refine the retrieval process, we calculate the **semantic similarity** between the query and the tags using a pre-trained *Sentence Transformer* model. This model generates dense vector embeddings for both the query and the image tags, allowing for a more nuanced comparison. The semantic similarity is computed by calculating the cosine similarity between the query embedding ($\mathbf{q}$) and the mean tag embedding ($\mathbf{T}_{\text{mean}}$) for each image:

$$\text{Semantic Similarity} = 1 - \cos(\mathbf{q}, \mathbf{T}_{\text{mean}})$$

- *Ranking of Images*: After calculating both the literal and semantic similarity scores, the total score for each image is derived by summing the individual similarity scores. Images are then ranked in descending order based on their total scores.

$$\text{Total Score} = \text{Literal Similarity} + \text{Semantic Similarity}$$

- *Thresholding and Display*: A threshold is applied to filter out images with low similarity scores, ensuring that only the most relevant images are presented to the user. The filtered images are resized and displayed along with their respective similarity scores for user evaluation.

C. Deployment on Streamlit and Supabase Integration

Following the development of the image tagging and retrieval system, the solution was deployed using **Streamlit** [7], a Python-based framework for building interactive web applications. Streamlit serves as the front-end interface, enabling users to upload images and query for similar images using text input.

For the back-end storage and management of images, **Supabase** [8] was employed. Supabase, an open-source platform that provides database and file storage capabilities, was used to store both the images and their associated tags. Images are uploaded to Supabase storage buckets, while the corresponding tags are stored in a JSON file. This integration ensures seamless communication between the front-end application and the back-end, facilitating smooth image storage, retrieval, and management.

VI. Experiments and results

We evaluated our model using several single-word and bi-word queries to assess its performance across different types of image retrieval tasks. The evaluation metrics primarily focused on precision at various ranks, Mean Average Precision (MAP), and Discounted Cumulative Gain (DCG). The results demonstrated that the model consistently maintained high precision for a significant number of retrieved documents, often achieving a step-by-step precision of 1. This remained true for a considerable number of queries before the model began retrieving slightly irrelevant documents.

A. Methodology

For the evaluation, we used a human-tagged dataset of images, where each image is associated with multiple class labels such as motorcycle, truck, boat, bus, cycle, person, desert, mountains, sea, sunset, trees, sitar, ektara, flutes, tabla, and harmonium. These labels are represented in a CSV format, with each row corresponding to an image and its respective class labels. The dataset structure is as follows:

$$\textbf{Image\_Name}, \text{motorcycle}, \text{truck}, \text{boat}, \text{bus}, \text{cycle}, \ldots$$

We first performed cosine similarity to rank the human-tagged documents that would serve as the benchmark for evaluating the relevance of the retrieved images. Cosine similarity measures the similarity between the query and each document, and based on a predefined threshold, we classified the documents into relevant and non-relevant categories. Images with similarity scores above the threshold were considered relevant, while those below the threshold were labeled as non-relevant.

The methodology for evaluating the model was as follows:

1) Cosine Similarity Ranking: We computed cosine similarity between the query and each image in the dataset to generate a ranking of images based on their relevance to the query.
2) Human-tagged Dataset: We used a dataset of human-tagged images as a benchmark. The tags associated with each image served as the ground truth for evaluating the relevance of the retrieved images.
3) Threshold for Relevance: Images were classified as relevant or non-relevant based on a similarity threshold. Those with similarity scores above the threshold were considered relevant, while others were treated as non-relevant.
4) Image Retrieval: The model retrieved images based on the query, and we evaluated its performance by calculating precision at various ranks, MAP, DCG, and nDCG scores.

B. Precision at k

Precision at k (denoted as `P@k`) is defined as the proportion of relevant documents retrieved in the top k results. The precision at a given rank k is calculated as:

$$P@k = \frac{\text{Number of relevant documents in the top } k}{k}$$

During the evaluation, the model achieved high precision values close to 1 for most queries. This indicates that the top-ranked images were highly relevant for a significant number of queries.

VII. Precision vs Recall

Another important aspect of evaluating the model is the trade-off between precision and recall. Precision measures the proportion of relevant documents among the retrieved documents, while recall measures the proportion of relevant documents that were successfully retrieved by the model. The relationship between precision and recall is typically illustrated by a Precision-Recall curve, where the precision may start high and gradually drop as recall increases.

For our model, we observed that for most recall values, the precision remained 1, indicating that the model initially retrieved only relevant documents. However, as more documents were retrieved (increasing recall), precision started to decrease, reflecting the inclusion of irrelevant documents in the results.

A. Precision-Recall Curve

The Precision-Recall curve shows the change in precision with respect to recall. The graph is plotted with recall on the x-axis and precision on the y-axis. The curve typically starts at the top-left corner (where recall is 0 and precision is 1), and as recall increases, precision tends to decrease. This is a common behavior in information retrieval tasks as more documents are retrieved, the likelihood of irrelevant documents appearing in the results increases.

To compute precision and recall, we use the following formulas:

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total no. of documents retrieved}}$$

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total no. of relevant documents in corpus}}$$

B. Interpolated Precision-Recall Curve

In many cases, it is useful to create an interpolated precision-recall curve, which smooths the precision values for each recall level. This helps to eliminate fluctuations in the curve and provides a clearer representation of the model's performance across different recall levels. The interpolated precision at each recall level is calculated by taking the maximum precision observed for any recall level greater than or equal to the current recall value.

The interpolated precision at a given recall value r is defined as:

$$\text{Interpolated Precision}(r) = \max_{r' \geq r} \text{Precision}(r')$$

This interpolation ensures that precision is non-increasing as recall increases, which is generally the case in information retrieval systems.

C. Graph Visualization

Below is the Precision-Recall curve for the model for an example query, showing the relationship between precision and recall for different recall thresholds. The curve illustrates the typical behavior where precision remains high (close to 1) for low recall values and drops as more documents are retrieved.

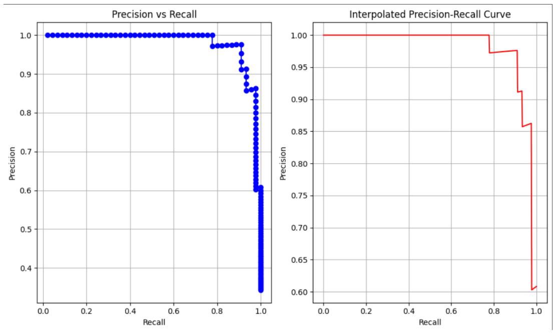

Fig. 2. Precision vs Recall and Interpolated Curve for the Model

The interpolated version of the Precision-Recall curve is also shown below, where the precision values have been smoothed for each recall level:

These graphs provide a clear visualization of the model's performance and the trade-off between precision and recall as the recall threshold increases.

D. Mean Average Precision (MAP)

The Mean Average Precision (MAP) is a metric that summarizes the precision across multiple queries. MAP is calculated as the mean of the average precision (AP) for each query. For a given query, the Average Precision (AP) is defined as:

$$AP = \frac{1}{n} \sum_{i=1}^{n} P@i \times rel(i)$$

Where $P@i$ is the precision at rank i and $rel(i)$ is a binary relevance indicator for the document at rank i. MAP is then the mean of AP values over all queries:

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)$$

Where Q is the total number of queries. In our evaluations, MAP scores ranged from 0.8 to 1, reflecting the model's consistency in retrieving relevant documents, with occasional fluctuations as the threshold for relevance was adjusted.

E. Discounted Cumulative Gain (DCG) and Normalized DCG (nDCG)

Discounted Cumulative Gain (DCG) is another important metric used to evaluate the ranking quality of the model. DCG is calculated by summing the relevance scores of the top k documents, discounted by their rank:

$$DCG_k = \sum_{i=1}^{k} \frac{rel(i)}{\log_2(i+1)}$$

Where $rel(i)$ is the relevance score of the document at rank i, and the logarithmic discount ensures that higher ranks contribute more to the DCG.

To account for the ideal ranking, we calculate the Ideal DCG (IDCG), which is the DCG of the ideal ranking of documents (i.e., the ranking where the most relevant documents are placed at the top):

$$IDCG_k = \sum_{i=1}^{k} \frac{rel_{ideal}(i)}{\log_2(i+1)}$$

Normalized DCG (nDCG) is the ratio of DCG to IDCG, providing a normalized value between 0 and 1, where a value of 1 indicates perfect ranking:

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

In our evaluation, the DCG scores were consistently high, indicating that the model ranked relevant documents highly. Furthermore, the nDCG scores often approached 1, reflecting the model's efficiency in ranking relevant documents with minimal irrelevant ones.

F. Conclusion

The evaluation results demonstrate that the model is effective in retrieving relevant documents for both single-word and bi-word queries. The precision remains high for most queries, and the MAP and DCG scores suggest that the ranking algorithm performs well. Adjusting the relevance threshold further minimizes retrieval errors, enhancing the overall performance of the model.

VIII. Conclusion

In this work, we have presented an end-to-end image tagging and retrieval system based on the BLIP model for caption generation and the integration of both literal and semantic similarity measures. The model generates relevant tags for images and enables efficient retrieval based on textual queries. By employing advanced natural language processing and image captioning techniques, we have developed a robust system capable of providing relevant images in response to user queries. The inclusion of both TF-IDF and Sentence Transformer-based semantic similarity ensures the retrieval of accurate results, thus enhancing the user experience.

The system was successfully deployed on Streamlit, enabling an interactive platform for users to query and retrieve images. Supabase was utilized for image storage, providing scalability and efficiency in managing large image datasets. Overall, this approach demonstrates a promising method for content-based image retrieval, with potential applications in various domains such as e-commerce, image search engines, and digital asset management.

IX. Limitations

Despite the effectiveness of the proposed system, there are certain limitations that need to be addressed:

- **False Positives**: Some non-relevant images may be retrieved for a given query, especially when the query is ambiguous or lacks specificity. Although we have implemented a relevance threshold to minimize this, some false positives may still exist.
- **Language Limitation**: The system currently supports queries in English. While the model can process English-language queries effectively, it may not perform as well with other languages, especially those with different syntax or structure.

X. Future Work

In future work, we plan to improve the system in the following ways:

- **Multi-language Support**: We aim to extend the system to support queries in Hinglish (a mix of Hindi and English) as well as other regional languages. This would enable users to conveniently enter queries in their preferred language and retrieve relevant images.
- **Enhanced Accuracy**: We plan to refine the image tag extraction process by integrating additional pre-trained models for captioning and exploring more advanced similarity algorithms to reduce false positives and improve retrieval accuracy.
- **Interactive Feedback**: Adding a feedback loop where users can rate the relevance of retrieved images would allow the system to learn from user preferences and improve over time.
- **Integration with mobile devices**: After model enhancement and optimizations, we plan to integrate our model with smartphones, so that users can use IRIS with their personal gallery

References

[1] J. Li *et al.*, "Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation," 2022.

[2] O. Jeunen, I. Potapov, and A. Ustimenko, "Dcg as an off-policy evaluation metric for top-n recommendation," *arXiv preprint arXiv:2307.15053*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.15053

[3] Y. Ge *et al.*, "Recognize anything: A dataset and model for universal image recognition," 2023.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with rpns," *IEEE Tns. on P.A. and Machine Intelligence*, 2015.

[5] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:1605.09759*, 2021.

[6] I. Retrieval and I. Search, "Iris: Image retrieval and intelligent search," https://github.com/punyamsingh/IRIS, 2024.

[7] S. Inc., "Streamlit," https://www.streamlit.io, 2021.

[8] ——, "Supabase: The open-source firebase alternative," https://supabase.io, 2021.