A Multimodal Deep Learning Approach to Lost & Found Item Retrieval

Ashhad Ahmed Kamran-[2222511], Xintong Zhu-[2220464], Qazi Maaz Pasha Syed-[2109074]

June 4, 2025

1 Introduction

Traditional lost and found systems in universities, airports, or transport hubs are typically inefficient, relying on manual processes and fragmented communication channels. These systems depend on subjective textual descriptions that often fail to capture unique item characteristics. A simple description like ”a bag” cannot distinguish between a backpack, handbag, or grocery bag, let alone describe color, material, or distinctive features. This reliance on incomplete information leads to low retrieval rates, causing frustration for owners and creating backlogs of unclaimed property for institutions.

The advent of sophisticated AI techniques, particularly in Natural Language Processing (NLP) and Computer Vision (CV), has paved the way for ”semantic search”, where systems can grasp the underlying meaning and context of queries rather than just literal terms. While advancements in individual AI modalities (e.g., text processing or image recognition) are significant, a purely text-based or image-based system remains insufficient for the complexities of lost-and-found. The true power for solving the challenges emerges from multimodal AI, which integrates and processes information from multiple modalities simultaneously. The most direct and influential predecessor to our system's core matching mechanism is Contrastive Language-Image Pre-training (CLIP) [3]. Developed by OpenAI, CLIP demonstrated a groundbreaking approach to learning transferable visual models from natural language supervision.

We have developed a Visual Search Engine that implements this multimodal approach, leveraging deep learning models—ResNet18 for image encoding and GRU for text encoding—to create a joint embedding space where similar images and text descriptions are mapped closely together. Our system supports both text-to-image search and image-to-image search with dynamic indexing capabilities. Building upon CLIP's foundational work, we implement a contrastive loss function that encourages embeddings of corresponding image-text pairs to have high cosine similarity while pushing non-corresponding pairs apart, creating an effective joint representation space for cross-modal search in lost and found scenarios.

2 Method

2.1 Dataset Preparation

To train our Cross-Modal Visual Search Engine, we selected the Amazon Products Dataset 2023 from Kaggle, which contains approximately 1.4 million product entries. This dataset was chosen for several strategic reasons:

1. It provides a diverse range of consumer products like electronics and bags that align with the lost-and-found scenario.
2. Each product entry contains both textual information (product titles) and corresponding images, making it ideal for paired text-image data.
3. The substantial size of the dataset ensures sufficient training data for learning robust cross-modal embeddings.

We built a data preprocessing pipeline that transforms the raw Amazon dataset into a format suitable for contrastive learning between images and text. The system extracts relevant fields, namely product titles and image URLs, from the dataset CSV file as the primary text-image pairs for training. Product titles serve as concise textual descriptions that capture essential item characteristics, while the corresponding product images provide visual representations.

2.2 Model Architecture

Our Cross-Modal Visual Search Engine employs a double-encoder architecture (Figure 1) that learns joint representations for images and text in a shared embedding space.

- **Image Encoder:**

  The image encoder uses a pre-trained ResNet-18 convolutional neural network [1] as the backbone, leveraging pre-trained weights for effective transfer learning. The original classification layer (fully connected layer) of ResNet-18 is replaced with a new linear layer. This layer projects the extracted image features into an N-dimensional embedding vector. The dimension N is chosen to match the output dimension of the Text Encoder. The final image embeddings are L2-normalized to ensure that similarity comparisons using dot products equate to cosine similarity.

- **Text Encoder:**

  To achieve robust semantic understanding of text, a pre-trained Sentence Transformer model [2] (e.g., all-MiniLM-L6-v2 from the sentence-transformers library) is employed. The encoder takes a list of raw text strings (the preprocessed titles from the dataset) as input and directly outputs dense sentence embeddings of a fixed dimension (e.g., 384 for all-MiniLM-L6-v2). Similar to the image embeddings, the resulting text embeddings are also L2-normalized.

- **Combined System (LostFoundSystem):**

  The nn.Module encapsulates both the ImageEncoder and the TextEncoder. It includes a learnable temperature parameter, which is optimized during training. This temperature scales the logits (similarity scores) before the cross-entropy loss calculation, helping to control the sharpness of the learned distribution.

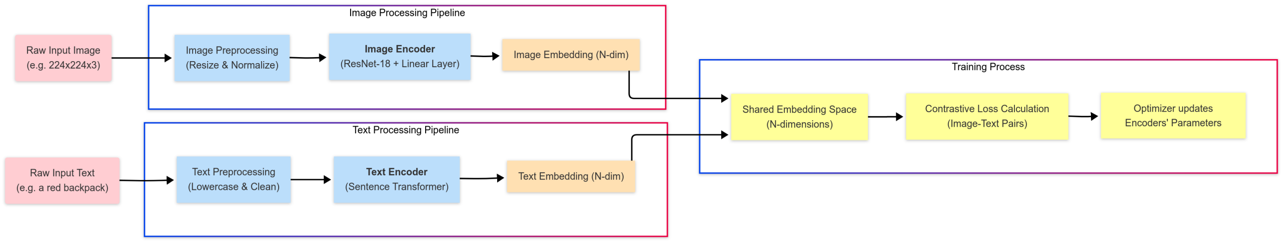


Figure 1: System Architecture of the Multimodal Lost-and-Found Model

2.3 Model Training

The model is trained using a contrastive learning approach. For a given batch of image-text pairs, the goal is to maximize the similarity score (cosine similarity) between corresponding pairs while minimizing the similarity scores for unrelated combinations. For each training batch containing N image-text pairs, the system computes a comprehensive similarity matrix of dimensions N × N. Each element (i,j) in this matrix represents the scaled dot product between the i-th image embedding and the j-th text embedding, with scaling determined by the learnable temperature parameter.

The loss function implements symmetric cross-entropy, which operates bidirectionally (both Image-to-Text and Text-to-Image) to ensure robust cross-modal alignment. The final loss is computed as the average of these two directional losses, ensuring balanced learning across both retrieval directions.

In each epoch, the model iterates through the training data loader, computes the loss for each batch, and performs backpropagation and optimizer steps. After each training epoch, a validation phase is executed where model performance is assessed without gradient updates. Both training and validation losses are continuously monitored to track learning progress and identify potential overfitting. This monitoring approach enables learning rate scheduling based on validation performance, ensuring optimal model generalization to unseen data.

3 Results

3.1 Hyperparameter Tuning and Training Performance

Initial experiments focused on determining an optimal learning rate for the Adam optimizer. Two primary rates, 0.0001 and 0.001, were evaluated over 10 epochs using the same dataset split. The training process involved minimizing a symmetric cross-entropy loss based on image-text similarity, with validation loss monitored for generalization and early stopping (patience of 4 epochs).

As depicted in Figure 2, the learning rate of 0.0001 yielded more promising results, achieving a validation loss of approximately 2.2840. In contrast, the learning rate of 0.001 resulted in significantly higher loss values (best validation loss of 2.8901 within 10 epochs), indicating that the smaller learning rate was more conducive to stable convergence for this particular model and dataset configuration.

After experimenting with different settings like learning rate and batch size, and training

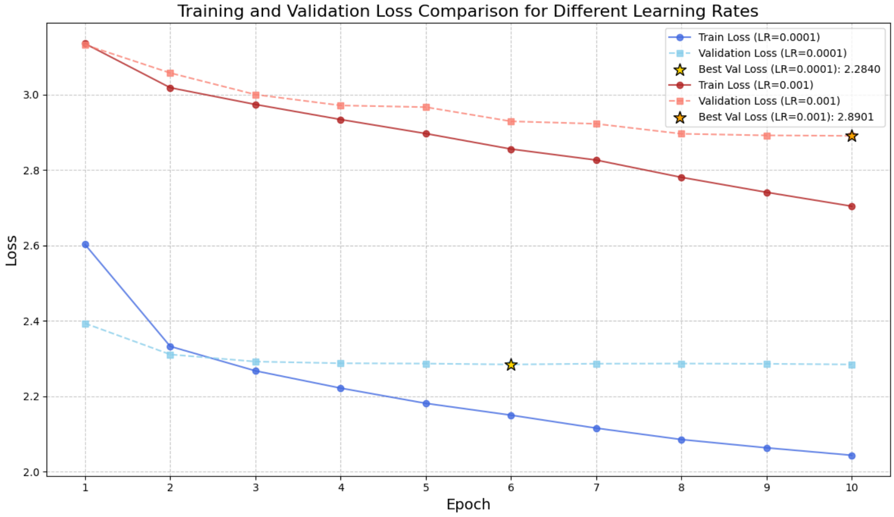


Figure 2: Training and Validation Loss for Different Learning Rates

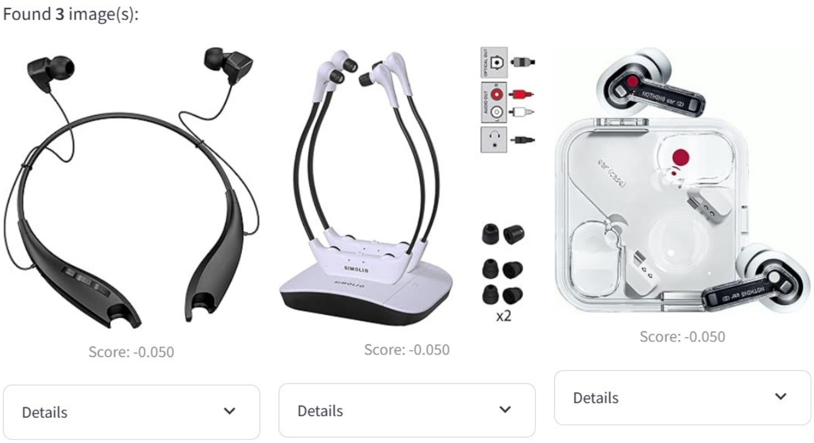
the model for more rounds, we were able to improve its performance significantly. Our best model achieved a validation loss of 1.5831, which represents a major improvement from our earlier attempts.

3.2 System Retrieval Performance

Beyond the quantitative loss metrics, the practical efficacy of the trained model was assessed by qualitatively evaluating its ability to retrieve relevant images based on both textual and image queries. Below are two representative examples demonstrating successful retrieval performance.

- Text Query: "Around The Neck Bluetooth Headphones"

  The system successfully identified and returned the top 3 most relevant product images (Figure 3) matching this textual description. The retrieved results show various bluetooth headphones, demonstrating the model's ability to understand both the product category.



Found 3 image(s):

Score: -0.050

Score: -0.050

Score: -0.050

Figure 3: Top-3 Elements Returned by the System

- Image Query: Uploading a specific product]

Given a query image of a particular product, the system could retrieve the same product as the top 1 result from the database. This demonstrates the model's capability to understand visual features and find products with similar characteristics, like colors and shapes.

4 User Interface

The system features a comprehensive web-based interface (Figure 4) developed using Streamlit, providing intuitive access to the multimodal search capabilities. The interface integrates three core functionalities: flexible input methods, customizable result presentation, and dynamic database expansion.

- **Main Searching Interface:**

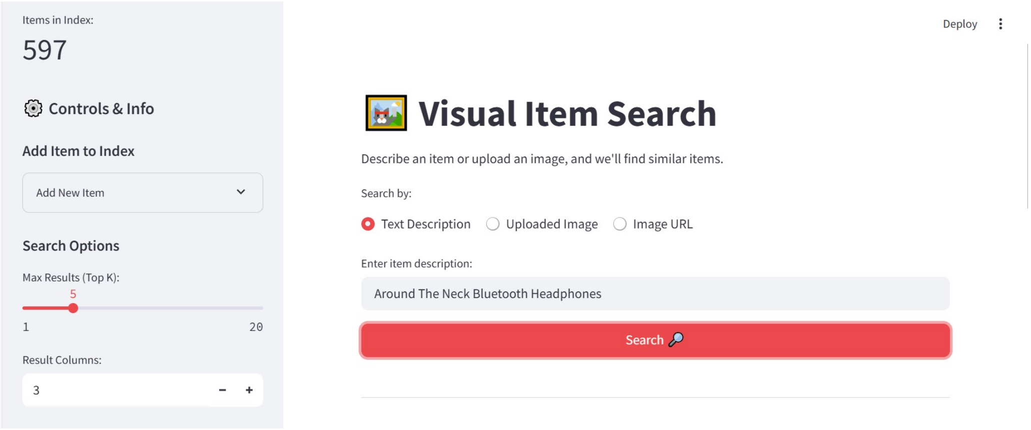
  The main interface offers users three distinct search modalities: Text Description, Uploaded Image, and Image URL. When using text search, users can enter natural language descriptions like "a black Nike backpack with SpongeBob on it". For image-based searches, the interface provides both file upload capabilities for local images and URL input for web-based images, accommodating different user preferences and scenarios.

- **Database Management Features:**

  The system has an "Add Item to Index" section that enables dynamic expansion of the searchable database. Users can contribute new items by providing an image URL and an optional description. Such capability allows the system to grow and adapt to new types of lost items over time, making it more effective for real-world settings.

- **Result Customization Panel:**

  Users can adjust the maximum number of results returned using an interactive slider ranging from 1 to 20 items. The result display can be customized through a "Result Columns" setting that allows users to control the grid layout of search results.



Items in Index:
597

Controls & Info

Add Item to Index

Add New Item

Search Options

Max Results (Top K): 5

Result Columns: 3

Visual Item Search

Describe an item or upload an image, and we'll find similar items.

Search by: ● Text Description ○ Uploaded Image ○ Image URL

Enter item description: Around The Neck Bluetooth Headphones

Figure 4: The web-based User Interface

5 Future Development

The current implementation of the multimodal visual search engine demonstrates the viability of using deep learning to address the challenges of lost and found item retrieval. However, to evolve this system into a more robust, scalable, and feature-rich application, several key areas for future development have been identified. These planned enhancements focus on improvements across data, model training, and user experience.

- Comprehensive Dataset Acquisition: The most critical next step is to replace the dummy data with a large, diverse, and well-annotated dataset of real-world lost and found items. This would enable the models to learn genuine visual and semantic features.

- Advanced Text Encoders: While the integration of Sentence Transformers (SBERT) marks a significant improvement over simpler models, further exploration into larger, more powerful transformer architectures (e.g., variants of BERT, RoBERTa available through libraries like Hugging Face Transformers) could yield even more nuanced semantic understanding of textual queries.

- User Feedback Loop: Developing a more sophisticated mechanism for users to provide feedback on search results (e.g., "this was relevant," "this was not relevant"). These comments could provide valuable data for future model fine-tuning or personalized reranking.

References

[1] Ayyachamy, S., Alex, V., Khened, M., & Krishnamurthi, G. (2019, March). Medical image retrieval using Resnet-18. In *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications* (Vol. 10954, pp. 233-241). SPIE.

[2] Devika, R., Vairavasundaram, S., Mahenthar, C. S. J., Varadarajan, V., & Kotecha, K. (2021). A deep learning model based on BERT and sentence transformer for semantic keyphrase extraction on big social data. *IEEE Access*, 9, 165252-165261. https://doi.org/10.1109/ACCESS.2021.3135361

[3] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Dhariwal, G., Mcgrew, M., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748-8763). PMLR.