Enhancing Visual Search Capabilities Through Visual Language Model

P Rohith
Dept.of CSE Nagarjuna College Of Engineeringand TechnologyBengaluru, India pallerohith1011@gmail.com

Nayana S
Dept.of CSE Nagarjuna College Of Engineeringand Technology Bengaluru, India nayana6723@gmail.com

Prithviraj P
Dept.of CSE Nagarjuna College OfEngineering and TechnologyBengaluru, India prithvi3147@gmail.com

Baba Fakruddin Ali B H*
Dept.of CSE Nagarjuna College OfEngineering and TechnologyBengaluru, India dr.babafali@gmail.com

Maulya Naik
Dept.of CSE Nagarjuna College Of Engineeringand TechnologyBengaluru, India maulyanaik08@gmail.com

Harshavardhana Doddamani*
Dept.of CSE Nagarjuna College Of Engineeringand Technology Bengaluru, India drhvdglb@gmail.com

How to Cite this Article:

Rohith, P., S, N., P, P., H, B. F. A. B., Naik, M. & Doddamani, H. (2026). Enhancing Visual Search Capabilities Through Visual Language Model. International Journal of Creative and Open Research in Engineering and Management, 02(6). https://doi.org/10.55041/ijcope.v2i6.162

https://doi.org/10.55041/ijcope.v2i6.162

Abstract—General-purpose Vision-Language Models (VLMs) like CLIP are suffered from a significant "domain gap" when they are applied to specialized fields, failing to differentiate nuanced visual categories. While fine-tuning is a known solution, the critical, secondary "data noise problem" that is arisen from using LLMs for dataset creation is addressed by this paper. It was found that nearly 19% of our initial LLM-generated culinary dataset was consisted of generic, "noisy" captions (e.g., "A photo of a food dish"). This work presents a comprehensive end-to-end methodology anchored in a rigorous data refinement framework designed to eliminate noise. This is combined with an iterative, sequential fine-tuning strategy that progressively has the learning rate decayed to prevent overfitting. This combined method was proved highly effective, with the model's performance being transformed on unseen validation data from a 77.56% baseline (on noisy data) to a peak accuracy of 93.00% (on the refined dataset). A reproducible blueprint for adapting general VLMs to niche domains is provided by this work, demonstrating that methodical data refinement is considered as critical as the model's architecture.

Index Terms: Vision-Language Models (VLMs), CLIP, Fine-Tuning, Domain Adaptation, Data Cleaning, Data Refinement, Semantic Search, Vector Search, FAISS, Contrastive Learning.

I. INTRODUCTION

The demand for intelligent search systems has been intensified by the exponential growth of digital content, particularly in e-commerce and media. Traditional visual search technologies, which are relied on metadata tags or low-level feature matching (e.g., color histograms), often are failed to bridge the "semantic gap" between a user's textual query and the visual content of an image. This is resulted in limited relevance and accuracy, especially in specialized domains with fine-grained visual distinctions, such as culinary arts.

Vision-Language Models (VLMs) are offered as a transformative solution by integrating computer vision and natural language processing.Models like CLIP (Contrastive Language–Image Pre- training) [1] have a joint embedding space learned where images and text can be semantically compared, enabling flexible, multi-modal search experiences.
The effective application of general-purpose Vision-Language Models (VLMs) to specialized domains is currently met with significant challenges. Primarily, there is a Domain Gap, where models pre-trained on broad web data, such as CLIP, struggle to handle the highly fine-grained, domain-specific queries required in specialized fields like medicine or engineering. This problem is compounded by Data Scarcity and Noise. While fine-tuning these models is crucial, it demands large-scale, high-quality, domain-specific datasets. The alternative—automated caption generation using Large Language Models (LLMs)—often introduces significant label noise, a factor proven to reduce model precision and interpretability. Finally, developers must contend with Overfitting. When a massive pre-trained model is fine-tuned on a comparatively small, specialized dataset, it tends to memorize the training data instead of learning generalizable features, causing it to fail when encountering new, unseen examples.
First, a robust data curation strategy that is involved with LLM- based caption generation followed by an essential data cleaning phase that was filtered out ~19% identified noise, is proposed, being resulted in a high-fidelity dataset of 6,283 image-caption pairs.
Second, an iterative, sequential fine-tuning strategy that is mitigated for overfitting by progressively being trained on small data chunks while previous weights are being loaded and the learning rate is being decayed, is introduced.

The rest of this paper is organized as follows: Section II provides a comprehensive review of Related Work, contrasting traditional systems with the foundational Vision-Language Models and defining the critical gaps addressed. Section III details the System Design, including the novel Data Refinement strategy, the CLIP model architecture, and the iterative fine-tuning process. Section IV presents the Results and Discussion, quantifying the 93.00% performance gain and discussing its equivalence to optimized compressive learning. Finally, Section V offers the Conclusion and outlines potential avenues for Future Enhancements.

II. RELATED WORK

This research is built upon the intersection of four key domains: 1) foundational Vision-Language Models (VLMs), 2) the core architectures that are enabled by them, 3) efficient vector indexing systems for retrieval, and 4) related information retrieval techniques.

The primary foundation of our work is the development of models that have a joint representation of images and text learned. The most prominent is CLIP (Radford et al.) [1], which was introduced with a highly effective contrastive learning method to align images and text within a shared embedding space. CLIP's ability to perform well in "zero-shot" settings (understanding concepts it wasn't explicitly trained on) is a significant advantage. However, its reliance on general-purpose web data is meant that it is struggled with the fine-grained, domain-specific queries that are targeted by our project.
This challenge of web-scale data is also addressed by Jia et al. [2], who was demonstrated that VLM training can be scaled using massive, noisy supervision (e.g., raw image alt-text pairs). While this is made robust for models to real-world data variations, a key trade-off is highlighted by it: data noise can be reduced in the model's final precision and interpretability.
Other architectures like FLAVA (Singh et al.) [3] have been explored for multi-task learning across both vision and language, being shown strong performance on a wide array of tasks. This power, however, often is come at the cost of high computational complexity. Research by Xie et al. [12] further is explored for the zero-shot capabilities of VLMs, but also is confirmed that they can be struggled with ambiguous queries or out-of-distribution samples, reinforcing the need for domain-specific fine-tuning.

The Transformer (Vaswani et al.) [4] is the revolutionary architecture that is provided as the foundation for modern NLP and vision models, enabling a deep understanding of long-range dependencies through self-attention. While foundational, pure Transformer models are known for their high memory usage and inference latency. This architecture is formed as the basis for text models like BERT (Devlin et al.) [16], which is provided with powerful, context-aware language embeddings, and vision models like ResNet (He et al.) [15] (which was introduced with residual connections to train deeper networks) and LVIT (Wang et al.) [20] (which is explored for more efficient, lightweight transformers for vision). A key limitation of BERT and ResNet on their own is that they are unimodal, being lacked native multi-modal integration.

The success of these models is made inseparable from large-scale datasets. ImageNet (Deng et al.) [14] was pioneered for this, being become a standard for computer vision training.
However, it is suffered from known dataset biases and, crucially, is not a multi-modal dataset. The "meaning" that is captured by these models is stored in vector embeddings. This concept was evolved from earlier NLP techniques. Word2Vec (Mikolov et al.)
[7] was a foundational technique for efficient word embeddings. Its primary limitation is that its vectors are context-independent and have no connection to visual information.

Chen et al. [17] was explored for contrastive learning specifically for visual-only features, which was helped to lay the groundwork for CLIP's subsequent cross- modal (image-text) contrastive approach.The comparison of these vectors often is relied on cosine similarity (Johnson et al.) [8], which is an effective and simple-to-implement metric for semantic matching. Its main drawback is a limited flexibility in handling highly complex or noisy inputs.

FaceNet (Schroff et al.) [18] was demonstrated for the power of a unified embedding space for a highly specific domain (face recognition). This is provided as a strong precedent for our project's goal: being specialized for a general model (CLIP) for a specific domain (culinary arts), as FaceNet model is not generalized to other VLM tasks.

FAISS (Johnson et al.) [5] is a key technology that is enabled for billion-scale similarity search with high retrieval speed, especially on GPUs. This technology is purely a retrieval mechanism and is possessed with no semantic understanding on its own; it is relied entirely on the quality of the vectors that are provided by the model. This is complemented by research into graph-based indexing methods like HNSW (Malkov et al.) [19], which is provided as an alternative efficient structure for large- scale nearest-neighbor search.

Research by Raghavan et al. [9] is focused on optimizing vector databases for fast query execution and scalability. This work, along with studies on real-time retrieval (Huang et al.), is highlighted for the importance of the engineering infrastructure, which often is lacked in depth in the semantic modeling itself.

While our focus is multi-modal search, work in related text-only and recommendation fields was also reviewed. These are included with methods for enhancing query embeddings (Das et al.) and comparative evaluations of ranking algorithms (Xia et al.). Other works are focused on session-based recommendations (Jannach et al. [21], Sheu et al. [24]) and hybrid approaches (Pereira et al. [22], Jamal et al. [23], Kipf & Welling [25]).

The foundation of the current work is laid by the evolution of Vision-Language Models (VLMs), with CLIP being the most prominent, where joint representations of images and text are learned through a highly effective cross-modal contrastive approach. This success is enabled by the revolutionary Transformer architecture, which forms the basis for both the text and vision encoders, representing an advancement over unimodal models like BERT (text) and ResNet (vision), which are inherently lacking in native multi-modal integration. While reliance on massive, often noisy, web-scale data has made VLM training scalable, this noise is known to reduce final precision, confirming a need for domain-specific specialization. Early work on vector representation was pioneered by techniques like Word2Vec, but its vectors were context-independent and lacked any connection to visual information, highlighting why the quality of the VLM's generated semantic vector is crucial.The specialization of the general VLM model is considered essential, as demonstrated by the strong precedent set by domain-specific embedding models like FaceNet. The feasibility of rapid classification and retrieval is ensured by high-speed vector indexing technologies such as FAISS and HNSW, which are relied upon for billion-scale similarity search. These retrieval systems only provide speed and are dependent entirely on the semantic quality of the specialized vectors that are generated by the fine-tuned VLM. Ultimately, the system design must integrate this specialized semantic output with the efficient retrieval mechanism, contrasting sharply with traditional methods like CBIR which are limited by a lack of semantic understanding.

III. SYSTEM DESIGN

Our methodology is integrated with a novel data refinement pipeline with an iterative fine-tuning strategy to adapt a general-purpose Vision-Language Model (VLM) for a specialized culinary domain. The implementation is comprised of three core stages: 1) Data Curation and Refinement, 2) Iterative Model Fine-Tuning, and 3) System Architecture for High-Speed Retrieval.

A. Data Curation and Refinement

To overcome the "domain gap", a specialized dataset was created. This process was begun by collecting 10,845 high-resolution culinary images from the Unsplash API.

To generate semantically rich captions, the Google Gemini API was utilized. A key step was prompt engineering the model was prompted to act as an "expert chef and food blogger" and to provide a detailed, vocabulary-rich description of the dish's ingredients, colors, and textures, rather than just a simple label. The pipelining is demonstrated in Figure 1.

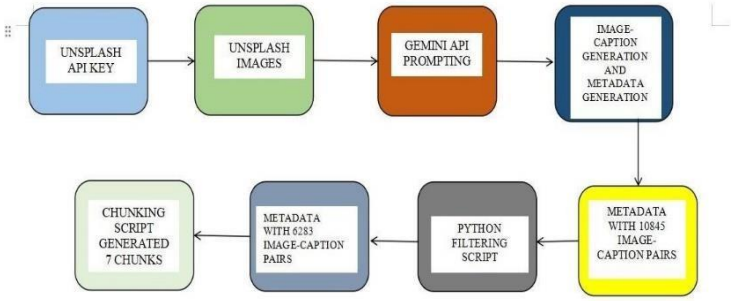

Figure 1. Data Curation Pipelining

This process, however, was introduced with significant "data noise". Our analysis of the 10,845 generated captions was revealed that 1,463 entries (~19%) were generic, non-descriptive fallbacks (e.g., "A photo of a food dish."). Training on this data would be poisoned for the model by being associated with nearly one-fifth of the images with the same meaningless phrase.

A data refinement methodology was implemented by running a filtering script to programmatically identify and discard all image-caption pairs containing these generic fallbacks. This crucial step was resulted in a final, high-fidelity dataset of 6,283 cleaned image-caption pairs, which was formed as the basis for our model training. And, below is the pseudo code used to acquire the required dataset using prompt engineering technique.

B. Iterative Model Fine-Tuning

The core of our system is the OpenAI/clip-vit-base-patch32 model, which was fine-tuned using the InfoNCE (Noise Contrastive Estimation) loss function. This contrastive loss is trained for the model to maximize the cosine similarity of correct image-text pairs while it is minimized for all other pairs in a batch.
To prevent catastrophic overfitting on our relatively small, specialized dataset (6,283 pairs), an iterative and sequential fine-tuning strategy was employed. The process was as follows:The cleaned dataset was split into smaller chunks (e.g., 1000 items each).

a) The model was trained on an initial chunk.

b) The best-performing model weights from that stage were saved.

c) These saved weights were then loaded as the starting point for training on the next data chunk.

d) This process was repeated for all chunks, while the learning rate was progressively being decayed (e.g., from 2e-6 down to 1e-7).
This iterative refinement was allowed for the model to make finer and finer adjustments as it was becoming more

specialized, successfully managing overfitting and achieving a final peak validation accuracy of 93.00%.
Pseudocode Iterative Sequential Fine-Tuning :
Input: Data_Chunks (List of 7 cleaned dataset parts), Base_Model Output: Final_Specialized_Model

// Define decaying learning rates for refinement Learning_Rates ← [2e-6, 1e-6, 5e-7, ..., 1e-7] Current_Model ← Base_Model

For each i from 1 to 6:
Dataset ← Load(Data_Chunks[i]) LR ← Learning_Rates[i]

// Load best weights from the previous stage If i > 1:
Current_Model ← LoadWeights(Path_To_Stage_{i-1}_Best_Model)

Initialize Optimizer with LR

// Training Loop for current stage For epoch in 1 to MAX_EPOCHS:
Train_Loss ← Train(Current_Model, Dataset)
Val_Loss, Val_Acc ← Validate(Current_Model, Validation_Set)

If Val_Loss < Best_Val_Loss:
Best_Val_Loss ← Val_Loss
Save Current_Model as "Best_Stage_{i}_Model" Else:
Increment Patience_Counter

If Patience_Counter > Limit:
Stop Training (Early Stopping) Return "Best_Stage_6_Model"

C. Why CLIP MODEL?

The CLIP (Contrastive Language-Image Pre-training) model was selected for this project due to its unique architectural strengths and its inherent capability to address the project's primary requirement: bridging the semantic gap between visual and textual data**.**

The fundamental reason for its adoption lies in its mechanism for multimodal embedding. The model is designed to learn a shared, high-dimensional vector space where the semantic meaning of an image is closely aligned with the semantic meaning of its corresponding text caption. This capability allows for the direct comparison of vectors generated by the image encoder (Vision Transformer) and the text encoder (Transformer). This contrasts sharply with traditional search methods, which are unable to grasp the complex, nuanced relationship between a textual query and the visual content of an image. Consequently, CLIP enables both Text-to-Image search (finding an image from a description) and Image-to-Text search (finding the best description for a picture), making the search process far more intuitive and expressive. Its architecture is depicted below in Figure 2.

A secondary, yet critical, advantage is its transferability and zero-shot capability. Having been pre-trained on a massive volume of web-scale data, the base CLIP model possesses an impressive general understanding of concepts. This provides a strong starting point that can then be specialized for the target domain. The model is intentionally fine-tuned on a high-quality, domain-specific dataset, addressing the "domain gap" problem that arises when generic models struggle with fine-grained distinctions (e.g., differentiating between specific culinary dishes). By fine-tuning the pre-trained CLIP model, this capability is harnessed and maximized, ensuring that the final output vectors precisely capture the required domain-specific semantics.

Finally, the output of the CLIP model is perfectly suited for high-speed retrieval systems. The 512-dimensional vector embeddings generated by the model are directly indexed using FAISS (Facebook AI Similarity Search). This implementation allows for the immediate, real-time comparison of query vectors against the entire dataset of images and captions.

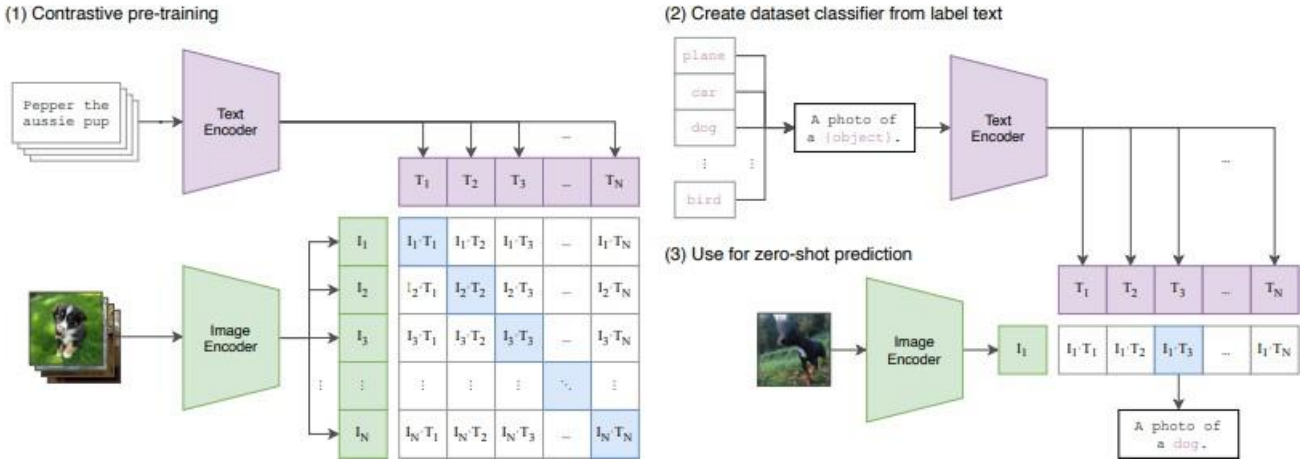

Figure 2. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes [1]

System Architecture for High-Speed Retrieval

The final component is a full-stack system that is designed for real-time inference.

Vector Indexing: Using our final 93% accuracy model, 512-dimensional vector embeddings were generated for all 6,283 cleaned images and captions. These vectors were then indexed into two separate, high-speed FAISS (Facebook AI Similarity Search) libraries (IndexFlatL2). One index is stored with the image vectors, and the other is stored with the text vectors.

API Deployment: A high-performance REST API, being built using FastAPI, was developed to serve the system. On startup, the API is loaded with the fine-tuned CLIP model, processor, and both FAISS indexes into memory to ensure low-latency query processing.

Pseudocode of Vector Indexing:

Input: Trained_Model, Cleaned_Dataset Output: Image_Index, Text_Index

Initialize Image_Index, Text_Index (FAISS FlatL2) For each batch in Cleaned_Dataset:

Images, Texts ← batch

// Generate 512-dim vectors

Image_Vectors ← Trained_Model.EncodeImage(Images) Text_Vectors ← Trained_Model.EncodeText(Texts)

// Normalize vectors for cosine similarity Normalize(Image_Vectors) Normalize(Text_Vectors)

// Add to FAISS Image_Index.Add(Image_Vectors) Text_Index.Add(Text_Vectors)

Save Image_Index, Text_Index to disk

Pseudocode of Real Time Search(API):

Function TextToImageSearch(User_Query_Text):

// 1. Convert text to vector

Query_Vector ← Trained_Model.EncodeText(User_Query_Text) Normalize(Query_Vector)

// 2. Similarity Search

// Find top K nearest image vectors in the index

Distances, Indices ← Image_Index.Search(Query_Vector, K=10)

// 3. Retrieve Metadata Results ← []

For idx in Indices:

Image_Path ← Metadata_Map[idx].Path Results.Append(Image_Path)

Return Results

Multi-Modal Search: The system is supported for two search modalities:

a. **Text-to-Image Search:** A user's text query is encoded by the CLIP text encoder. The resulting vector is used to search the FAISS image index for the top-K most similar images.

b. **Image-to-Text Search:** A user's uploaded image is encoded by the CLIP image encoder. The resulting vector is used to search the FAISS text index for the top-K most relevant pre-existing captions.

IV. RESULTS

This section is presented with the quantitative and qualitative results of our end-to-end implementation. The data is demonstrated for the critical impact of our data refinement methodology and the success

of our iterative fine-tuning strategy.
which was transformed for the general-purpose CLIP model into a high-accuracy, domain-specific search engine.

Quantitative Analysis: Our training was conducted in distinct stages to measure the impact of our interventions. The model's progression from a generalist to a specialist is summarized in Table I.

Initially, training on the original, noisy dataset (Stages 1-3) was shown limited gains and rapid overfitting, with performance being plateaued at a 77.56% validation accuracy and a validation loss of 0.7323.

The "critical turning point" was the Data Cleaning phase. After 1,463 (~19%) generic, "noisy" captions were filtered out, training was resumed (Stage 4). The results were immediate and dramatic:

i. Validation Loss dropped by 36% (from 0.7323 to 0.4644).

ii. Validation Accuracy jumped from 77.56% to 85.00%. Subsequent stages (5 and 6), which involved training on more clean data while progressively lowering the learning rate (e.g., from 5e-7 to 1e-7), pushed the model to its final peak performance of 93.00% validation accuracy and a best validation loss of 0.2345.

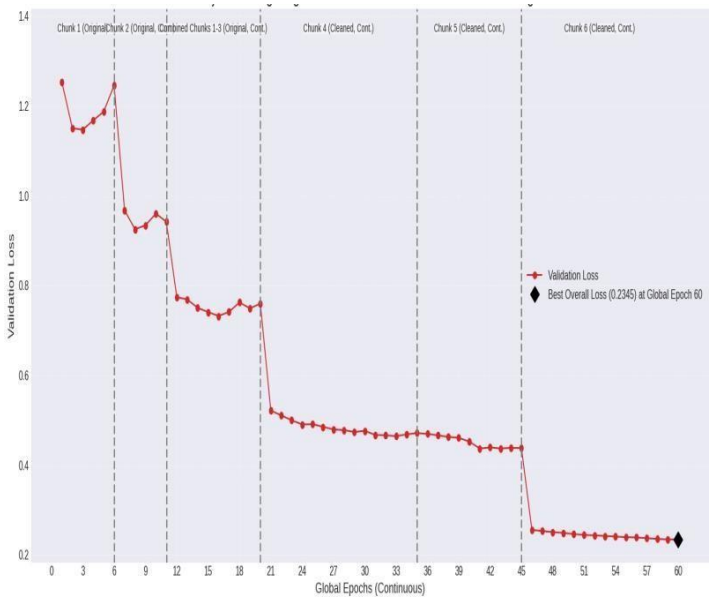

Figure 3. Validation loss vs Epochs

This progression is illustrated in Figure.1 and Figure.2. The validation loss graph (Figure. 1) clearly shows the model overfitting in Stages 1-3 before the immediate, sharp drop at Stage 4, followed by a steady optimization.

The validation accuracy (Figure. 3) mirrors this, hitting a hard plateau at 77.56% and then "stepping up" to 85.00% and climbing to 93.00% only after the noisy data was removed. This progression is illustrated in Figure.1 and Figure.2. The validation loss graph (Figure. 1) clearly shows the model overfitting in Stages 1-3 before the immediate, sharp drop at Stage 4, followed by a steady optimization. The validation accuracy (Figure. 3) mirrors this, hitting a hard plateau at 77.56% and then "stepping up" to 85.00% and climbing to 93.00% only after the noisy data was removed.

Table 1. Iterative Fine-Tuning Stages and Performance Progression

Stage	Data Used	Starting Model (Loaded From)	Key Parameter Change (LR)	Best Val Loss	Peak Val Acc. (%)
1	Orig. Chunk 1	Base CLIP	2e-6	1.1475	69.33 %
2	Orig. Chunk 2	Best of Stage 1	1e-6	0.9256	75.00 %
3	Orig. Chunks 1-3	Best of Stage 2	1e-6	0.7323	77.56 %
---	--- (Data Cleaning Applied) ---	---	---	---	---
4	Cleaned C4	Best of Stage 3 (Orig.)	5e-7	0.4644	85.00 %
5	Cleaned C5	Best of Stage 4 (Cleaned)	5e-7	0.4371	89.00 %
6	**Cleaned C6**	**Best of stage 5 (cleaned)**	**1e-7**	**0.2345**	**93.00 %**

Qualitative Validation: Semantic Understanding

zBeyond quantitative metrics, we performed qualitative tests to confirm the model's fine-grained semantic understanding. As shown in Figure. 6, we tested the model with an image of croissants and several text descriptions. The model assigned a 100.00% probability to the correct, nuanced description ("A close up of freshly baked croissants on a wooden board.") and 0.00% to all incorrect culinary descriptions (e.g., "South Indian Masala Dosa"). This result validates its specialization and ability to differentiate specific domain concepts.

System Integration Demonstration

Finally, the fully integrated full-stack application was tested Figure. 3 shows the end-to-end system results.

Text-to-Image: A text query ("biriyani") sent from the React frontend correctly queried the Fast API backend and FAISS image index, returning 10 relevant images demonstrated in Figure 4

Figure 4. Text-to-Image retrieval demo

Image-to-Text: An uploaded image ("chicken curry with rice.avif") correctly queried the FAISS text index, returning 10 semantically relevant, pre-existing captions from the dataset demonstrated in Figure 5.

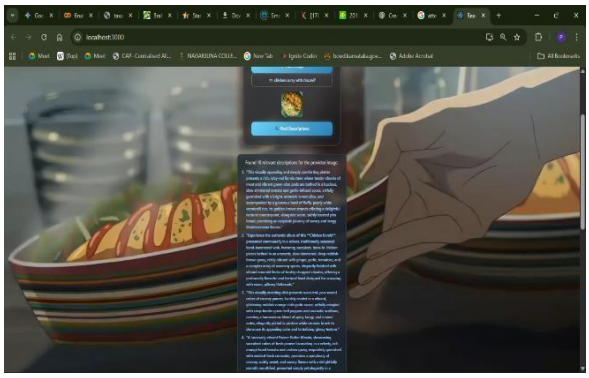

Figure 5. Image-to-Text retrieval demo

These tests confirm the successful integration of the specialized model and high-speed vector search into a functional, real-time application.

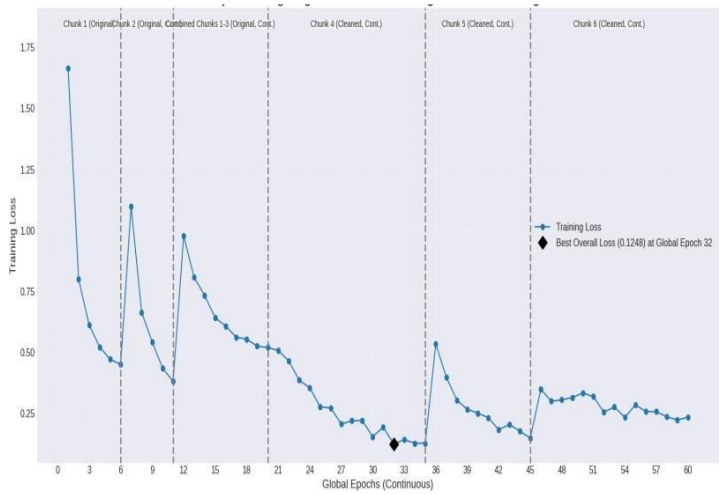

Figure 6. Training Loss vs Epochs

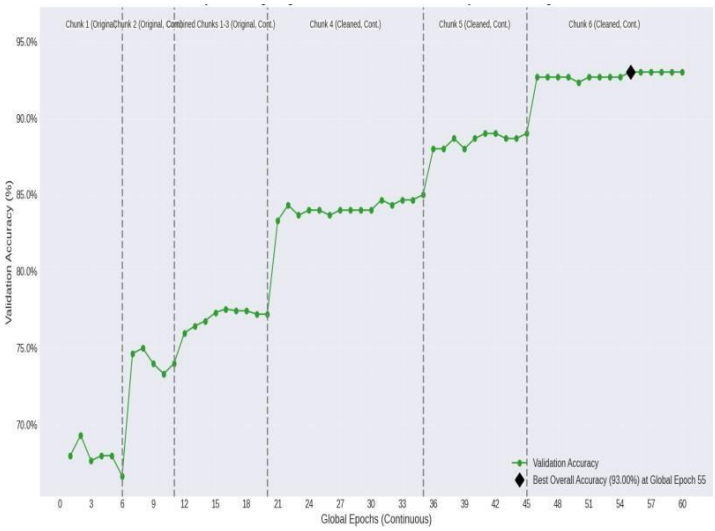

Figure 7. Validation Accuracy vs Epochs

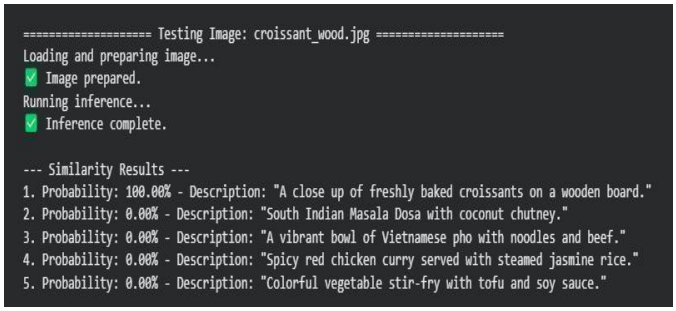

Figure 8. High-Confidence Retrieval on Unseen Validation Data.

V. CONCLUSION

A high-performance, domain-specific visual search engine was successfully designed, implemented, and validated. This project fundamentally resolved the challenge of applying a general-purpose Vision-Language Model (VLM) to the specialized culinary domain and provides a clear blueprint for adapting large, pre-trained models to niche applications. First, it was established that while large language models (LLMs) are powerful tools for rapid dataset creation, they introduce significant "noise". The necessity of a rigorous data refinement methodology was proven, as the removal of 1,463 noisy, non-descriptive captions (approximately 19% of the initial data) was found to be the most critical factor in improving model stability and generalization. Second, an effective domain specialization strategy was demonstrated. Overfitting was successfully managed by sequentially training the model on cleaned data chunks. This process involved loading the best weights from the previous training stage and progressively decaying the learning rate (from 2×10^{-6} to 1×10^{-7} This iterative specialization process transformed a model with a struggling baseline accuracy of approximately 77% into a specialized expert, achieving a final peak performance of 93.00% validation accuracy. Finally, the specialized model was successfully integrated with a high-speed FAISS vector index and deployed via a robust FastAPI and React full-stack architecture.

VI. FUTURE ENHANCEMENTS

A key area for research and development involves Generative Captioning Integration, which would involve clubbing the fine-tuned CLIP image encoder with a powerful Natural Language Processor (NLP) decoder (such as a fine-tuned GPT-2 or BART model)1. This would allow the system to generate completely new, context-rich captions for images it has never seen, providing detailed, novel descriptions rather than just retrieving pre-existing ones2. Additionally, the system's search power can be significantly improved by implementing Vector Arithmetic, allowing support for advanced, exclusionary queries (e.g., searching for "spicy curry without chicken") by subtracting the embedding of the negative term from the positive query vector before querying FAISS3. For better system stability and scalability, the current fixed, step-wise learning rate decay should be replaced with a more sophisticated approach like a Cosine Annealing with Warmup scheduler to ensure the model reaches an optimal minimum more efficiently4. Finally, to ensure
$24/7$ availability and scalable service capacity, the FastAPI backend should be migrated to a persistent cloud service (e.g., Google Cloud Run or AWS ECS) using Docker containers5.

REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal,
G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and
I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021, pp. 8748-8763. 1

[2] 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 30 (NIPS), 2017,
pp. 5998-6008. 2

[3] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "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 (NAACL-HLT), 2019, pp. 4171-4186. 3

[4] C. Jia, Y. Yang, Y. Xia, Y. T. Chen, Z. Parekh, H. Pham, Q. V. Le,
Y. H. Sung, Z. Li, and T. Duerig, "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021, pp. 4904-4916. 4

[5] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M. Rohrbach, and D. Kiela, "FLAVA: A Foundational Language and Vision Alignment Model," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 15638-15650. 5

[6] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547, 2021. 6

[7] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824-836, 2020. 7

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal,G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021, pp. 8748-8763.

[9] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal,G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, andI. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021, pp. 8748-8763. 1

[10] 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 30 (NIPS), 2017,pp. 5998-6008. 2

[11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "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 (NAACL-HLT), 2019, pp. 4171-4186. 3

[12] C. Jia, Y. Yang, Y. Xia, Y. T. Chen, Z. Parekh, H. Pham, Q. V. Le,
Y. H. Sung, Z. Li, and T. Duerig, "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision," in Proceedings of the 38th International Conference on Machine Learning (ICML), 2021, pp. 4904-4916. 4

[13] A. Singh, R. Hu, V. Goswami, G. Couairon, W. Galuba, M.Rohrbach, and D. Kiela, "FLAVA: A Foundational Language and Vision Alignment Model," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 15638-15650. 5

[14] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Transactions on Big Data, vol. 7, no. 3, pp. 535-547, 2021. 6

[15] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 824-836, 2020. 7

[16] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009,
pp. 248-255. 8

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778. 9

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823. 10

[19] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in Proceedings of the 37th International Conference on Machine Learning (ICML), 2020, pp. 1597-1607. 11

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proceedings of the 1st International Conference on Learning Representations (ICLR), Workshop Track, 2013. 12

[21] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proceedings of the 5th International Conference on Learning Representations (ICLR), 2017. 13

[22] T. Tiangolo, "FastAPI," [Software], 2018. Available: https://fastapi.tiangolo.com/ 14

[23] Facebook AI Research, "Faiss: A library for efficient similarity search," [Software], 2017.Available:https://github.com/facebookresearch/faiss 15

[24] Meta (formerly Facebook), "React: A JavaScript library for building user interfaces," [Software], 2013. Available: https://reactjs.org/

[25] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 8024-8035. 17

[26] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proceedings of the 2020 Conference on EMNLP: System Demonstrations, 2020, pp. 38-45. 18

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai,T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proceedings of the 9th International Conference on Learning Representations (ICLR), 2021. 19

[28] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal,A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert- Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu,C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems 33 (NeurIPS), 2020, pp. 1877-1901. 20

[29] Meta AI, "Llama 3: Open Foundation and Instruct Models," arXiv preprint arXiv:2404.11032, 2024. 21

[30] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual Instruction Tuning," in Advances in Neural Information Processing Systems 36 (NeurIPS), 2023. 22

[31] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language- Image Pre-training with Frozen Image Encoders and Large Language Models," in Proceedings of the 40th International Conference on Machine Learning (ICML), 2023, pp. 19780-19798. 23

[32] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests," in Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 446-461. 24

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, & I. Polosukhin, "Attention Is All You Need," 2017. 25