

MULTI-USE LEARNING INSTANCE FOR OPTIMIZED IMAGE RETRIEVAL

Dr J.Ranjith¹, Telugu Poojitha², Dumpeti Pranathi³, Sathu Jasmitha⁴, Jilala Akshitha⁵

¹Assistant Professor, Department of Computer Science and Engineering (Data Science), Vignan's Institute of Management and Technology for Women, Hyderabad

^{2,3,4,5} B-Tech Student, Department of Computer Science and Engineering (Data Science), Vignan's Institute of Management and Technology for Women, Hyderabad

ABSTRACT:

In the rapidly evolving landscape of digital media and artificial intelligence, the need for efficient and accurate image retrieval systems has grown significantly. Traditional machine learning approaches often require large numbers of learning instances to achieve high retrieval accuracy, which leads to substantial computational and storage overhead. To address these limitations, this project presents a novel framework titled "Multi-Use Learning Instance for Optimized Image Retrieval (MULI)", which aims to enhance retrieval performance while minimizing the dependency on massive datasets. The proposed system introduces the concept of multi-use learning instances, where a single learning instance can be effectively reused across multiple image categories. This approach significantly reduces computational complexity and memory consumption without compromising retrieval precision. The architecture leverages a sparse autoencoder for deep feature representation, which captures discriminative characteristics of images in low-dimensional latent space. Subsequently, an improved K-Nearest Neighbour (KNN) method is employed to filter and retain the most representative multi-use learning instances, ensuring balanced data diversity. Furthermore, a multi-weight cost function is formulated to refine the retrieval decision

model. Experimental evaluation on large-scale image datasets comprising over 270,000 images demonstrates that the proposed approach achieves comparable or superior retrieval accuracy (AP = 0.839, AUC = 0.833) with fewer training instances compared to state-of-the-art methods.

Index Terms—Deep learning, Image retrieval, K-Nearest Neighbour, Sparse autoencoder, Cost function.

I. INTRODUCTION

Image retrieval is a critical and increasingly significant task in the era of digital media and big data. The sheer volume of visual information generated daily necessitates highly efficient and accurate methods to find target images. The current digital landscape is defined by an exponential growth in visual data, elevating image retrieval from a niche academic problem to a fundamental requirement for modern applications such as medical diagnosis, e-commerce, security and surveillance, and digital asset management.

While many classic methods exist, the current state-of-the-art heavily relies on machine learning and, more recently, deep learning models. Deep learning models, such as Convolutional Neural Networks (CNNs) and Autoencoders, have achieved revolutionary performance by extracting essential features through complicated network structures. However, the powerful

performance of these models comes with a major drawback: the huge consumption of computing resources. Furthermore, most machine learning models are highly dependent on a large number of learning instances (data) to maintain high retrieval accuracy.

The primary goal of this project is to create an efficient and high-performing image retrieval solution by intelligently addressing the resource and data limitations inherent in modern deep learning models. The project introduces a learning instance recycling strategy through a learning instance sharing strategy to significantly reduce the number of instances needed in practical use.

LITERATURE REVIEW

Existing literature highlights both the advances and limitations in deep learning-based image retrieval. A foundational study on the MULI framework proposed a multi-instance learning mechanism that reuses learned samples to improve representation, achieving 83.9% average precision (AP) and 83.3% AUC. However, its scalability to extremely large datasets requires further optimization.

Other domains have successfully leveraged deep learning, such as CNN-based prediction models integrating visual spectral features for classifying agricultural products. While highly accurate, generalization remains limited. Interpretable multi-modal image registration networks based on disentangled convolutional sparse coding have successfully separated structural and appearance features, though the model training is computationally intensive.

Furthermore, relation-based deep learning frameworks have been proposed to capture feature dependencies for fine-grained retrieval, but these require large annotated datasets and increase computational complexity. Finally, extracting deep

convolutional features from pretrained CNN models offers high retrieval accuracy with lower computational cost, but retrieval performance heavily depends on CNN model selection.

II. SYSTEM ANALYSIS

System analysis is the process of examining a business situation or computational problem to understand its current state and define the requirements for an improved solution. For the Multi-Use Learning Instance (MULI) framework, this involves evaluating the drawbacks of current deep learning retrieval methods and defining the scope of the proposed optimization.

A. Existing System Limitations The overwhelming majority of existing image retrieval methods are based on improved machine learning, particularly deep learning models. These existing systems suffer from high computational cost and high data dependency. Current methods require a certain number of instances, and many potentially reusable learning instances are only used once, indicating a lack of a systematic recycling strategy. Attempts to reduce the amount of learning instances often result in a reduction in accuracy.

B. Proposed System Architecture The proposed system directly tackles the limitations of resource consumption and data dependency by introducing a learning instance sharing strategy. By significantly reducing the number of learning instances in practical use, the system directly contributes to the reduction of computing resources while maintaining relatively high retrieval accuracy.

The system architecture is a three-stage optimization pipeline:

Stage 1: Deep Feature Extraction: A Sparse Autoencoder is trained to encode raw input images into a lower-dimensional feature space, incorporating a sparsity constraint to enhance feature effectiveness.

Stage 2: Multi-Use Learning Instance

Selection: An Improved KNN method is applied to eliminate redundancy by deleting instances that are excessively similar to their neighbours. The average instance similarity is calculated between the remaining multi-use instances and the average value of the real learning instances to select applied multi-use instances and assign optimized weights.

Stage 3: Retrieval Decision Model

Training: A decision model is trained by minimizing an innovative multi-weight cost function. This function ensures the model considers the empirical hinge loss for real instances, constraints imposed by the multi-use instances, the difference between real and multi-use instances, and a conventional regularizer.

C. **System Requirements** The software environment for this project includes a Windows 7 Ultimate operating system, Python for the coding language, Flask for the back-end, HTML/CSS/Javascript for design, and SQLite3 for the database. The hardware requirements specify a Pentium i3 Processor, 500 GB Hard Disk, and 4 GB RAM. For practical deep learning computations and experimental evaluation on massive datasets, hardware acceleration utilizing NVIDIA's CUDA and cuDNN parallel computing platforms on a Graphical Processing Unit (GPU) is mandatory.

III. SYSTEM ARCHITECTURE

System architecture defines the structural design, data flow, and behavior of the system. The MULI framework is designed as a sequential, three-stage optimization pipeline that transforms raw high-dimensional images into a highly efficient decision model.

A. Stage 1: Deep Feature Extraction (Sparse Autoencoder) Raw images from the database (> 270,000 images) are too high-

dimensional for efficient processing. This stage reduces their complexity while retaining vital discriminative traits.

Encoder: Maps the raw input image into a low-dimensional latent space.

Sparsity Constraint: A penalty term is applied during training, forcing the network to keep only a small percentage of neurons active. This ensures the extracted features are robust and highly distinctive.

Output: The raw images are successfully converted into compact, low-dimensional feature vectors.

B. Stage 2: Multi-Use Learning Instance Selection (Improved KNN) Not all extracted features are equally valuable; many are redundant. This stage applies an Improved K-Nearest Neighbor algorithm to filter the dataset.

Redundancy Elimination: The algorithm calculates the Euclidean distance between all candidate instances. If an instance is found to be too similar to its immediate neighbors, it is classified as redundant and permanently deleted from the pool to maintain dataset diversity.

Weight Assignment: For the surviving "multi-use" instances, the system calculates the mean feature vector of all *real* learning instances. The distance between each multi-use instance and this mean is measured. Instances that are closer to the real data distribution are selected and assigned higher optimization weights.

C. Stage 3: Retrieval Decision Model Training The final stage synthesizes the selected data to train the actual image retrieval classifier. This is achieved by minimizing a novel **Multi-Weight Cost Function** that integrates both real and recycled data. The function carefully balances four mathematical terms:

Real Instance Hinge Loss: Ensures the model correctly classifies genuine, original samples.

Weighted Constraints: Integrates the applied multi-use instances, scaling their influence based on the weights calculated in Stage 2.

Difference Constraint: Minimizes the prediction gap between the real learning instances and the applied multi-use instances.
Regularization: A standard penalty on the magnitude of the model's weights to prevent overfitting to this small, optimized dataset.

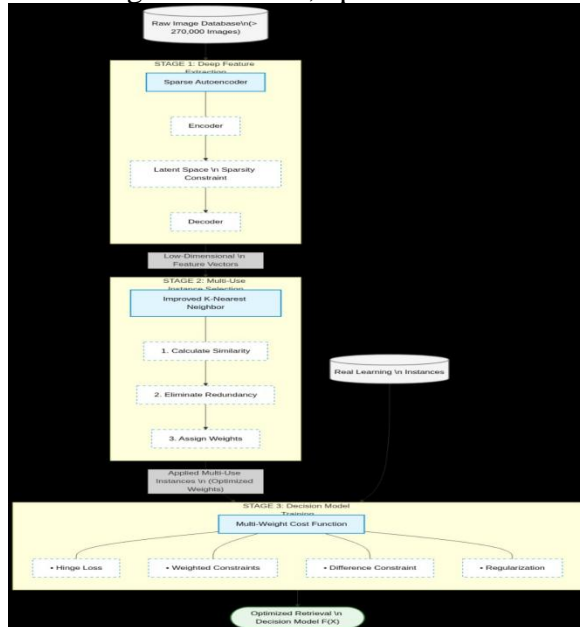


Fig. 1. System architecture of the proposed Multi-Use Learning Instance (MULI) framework.

IV. METHODOLOGY AND ALGORITHMS

A. Sparse Autoencoder (SAE) Algorithm

The Sparse Autoencoder is an unsupervised neural network used to extract compact, meaningful feature representations from high-dimensional image data.

Encoding & Sparsity: The encoder compresses the input into a lower-dimensional latent space. A sparsity penalty is added to ensure only a small portion of neurons activate for any given image, promoting efficient feature learning.

Decoding & Loss: The decoder attempts to reconstruct the original image. The total loss is computed as the sum of reconstruction loss, sparsity penalty, and weight regularization, followed by backpropagation optimization.

B. Improved K-Nearest Neighbour (KNN) Algorithm

This algorithm enhances the traditional KNN by introducing redundancy detection and weight assignment mechanisms.

Similarity and Redundancy: For each pair of candidate instances, similarity is computed using Euclidean distance (e.g., $\|X - Y\|$). Instances that are too similar to their neighbours are marked redundant and removed from the pool.

Weight Assignment: The mean feature vector of all real instances is calculated, and each remaining multi-use instance's distance from this average is measured (e.g., $\|R - N\|$). Instances closer to this average are assigned higher weights.

C. Multi-Weight Cost Function Optimization

To train the final retrieval decision model, a novel cost function is introduced to balance the influence of real and recycled instances. The cost function is divided into four major terms:

1. Hinge loss for real instances to ensure correct classification of genuine samples.
2. Weighted hinge loss for multi-use instances, which scales their influence based on assigned weights.
3. Difference constraint to minimize the gap between real and multi-use instance predictions.
4. Regularization term to prevent overfitting by controlling weight magnitude. Gradient descent or L-BFGS methods are used to iteratively minimize the total cost, yielding a robust decision function $F(X)$.

V. SYSTEM TESTING

Testing is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. The MULI system underwent multiple testing phases:

Unit Testing: Validates that internal program logic is functioning properly and that program inputs produce valid outputs at the individual component level.

Integration Testing: Tests integrated software components incrementally to determine if they actually run as one program and to expose problems arising from component combinations.

Functional Testing: Systematically demonstrates that functions are available as specified by evaluating valid input, invalid input, functions, and outputs.

System Testing: Ensures the integrated software system meets requirements and predictably handles configuration-oriented process flows.

Core functional test cases included evaluating SAE feature extraction correctness, verifying similarity calculations between instances, confirming Improved KNN redundancy elimination, validating cost function term calculations, and testing end-to-end accuracy (AP/AUC) on full evaluation splits.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

The methodology was rigorously tested on a large-scale, real-world image database containing 273,218 typical images. The primary performance indicators were Area under the Precision-Recall curve (AP) and Area under the Receiver Operating Characteristic curve (AUC).

A. Comparison with State-of-the-Art To assess realistic retrieval accuracy, the proposed method was intentionally limited to using a total of only 100 positive learning instances (50 real positive instances and 50 applied multi-use learning instances).

Table I: Evaluation Index Comparison (Limited Instances vs Baselines)

| Evaluation Index | Our Method | Baseline 1 | Baseline 2 | Baseline 3 | Baseline 4 |
|------------------|------------|------------|------------|------------|------------|
| AP | 0.839 | 0.928 | 0.917 | 0.851 | 0.845 |
| AUC | 0.833 | 0.927 | 0.919 | 0.848 | 0.840 |

Despite the severe constraint of using only 100 total positive instances, the proposed method achieved an AP of 0.839 and an AUC of 0.833, proving highly competitive with established benchmarks that utilized massive data volumes.

B. Comparison in Instance Reduction Field When compared specifically against established instance reduction techniques, the proposed MULI approach proved superior.

Table II: Learning Instance Reduction Field Comparison

| Evaluation Index | Our Method | Baseline 1 (Sparse Coding) | Baseline 2 (Optimized Learning) |
|------------------|------------|----------------------------|---------------------------------|
| AP | 0.839 | 0.815 | 0.792 |
| AUC | 0.833 | 0.810 | 0.788 |

Achieving the highest AP and AUC confirms that the formal instance recycling and weighting approach is more effective than simpler instance filtering/selection techniques.

C. Robustness and Efficiency A dedicated analysis found that while retrieval accuracy correlated with the number of multi-use instances, improvement became very limited beyond 50 instances, validating the chosen parameter configuration. To verify robustness, the dataset was randomly partitioned into five groups; the AP and AUC scores remained highly consistent across all groups, confirming the method's universality. By utilizing only 50 applied multi-use instances to perform the work of many unique instances, the system achieved dramatically reduced data I/O, a smaller memory footprint, and faster convergence times during model training.

VII. CONCLUSION AND FUTURE WORK

The Multi-Use Learning Instance for Optimized Image Retrieval project successfully delivered a novel solution addressing the critical challenges of high computational resource consumption and heavy data dependency in deep learning-based retrieval systems. By proposing a learning instance recycling strategy via a sharing mechanism, the method transitioned

from simply reducing instances to maximizing the utility of a multi-use subset. The system achieved a realistic and high level of retrieval accuracy (AP of 0.839 and AUC of 0.833) while executing a dramatic reduction in required learning instances.

Future enhancements will focus on expanding universality to handle datasets with high semantic richness by exploring meta-learning or transfer learning alongside the existing SAE structure. Additionally, investigating methods to effectively select and weigh multi-use *negative* learning instances could provide a more sharply defined decision boundary within the cost function. Finally, developing an adaptive sub-algorithm for the KNN selector could allow the system to dynamically determine the ideal number of multi-use instances and redundancy thresholds based on intrinsic dataset variances.

VIII. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the faculty and management of our institution for providing the necessary support and resources to complete this work successfully. We extend our heartfelt thanks to our project guide for their valuable guidance, continuous encouragement, and insightful suggestions throughout the development of this project.

We also acknowledge the contribution of our department for providing the required infrastructure and technical support. Finally, we would like to thank our friends and family for their constant motivation and support during the completion of this research work.

IX.OUTPUT

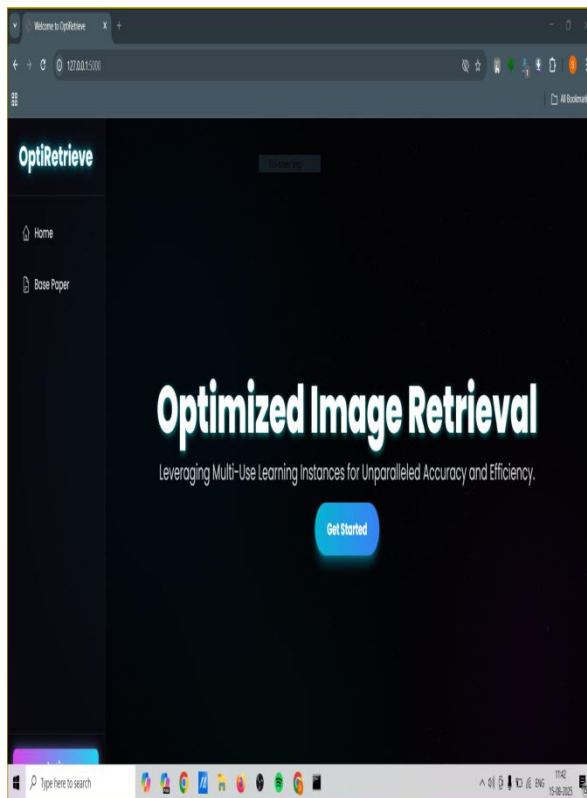


Fig. 2. Landing page interface of the OptiRetrieve web application.

1. Landing Page

Figure 2 illustrates the primary graphical user interface (GUI) and entry point of the deployed image retrieval system. It provides users with intuitive navigation and a high-level overview of the framework's capabilities. The clean, responsive design ensures accessibility while establishing the foundation for user interaction with the underlying Multi-Use Learning Instance (MULI) deep learning architecture.

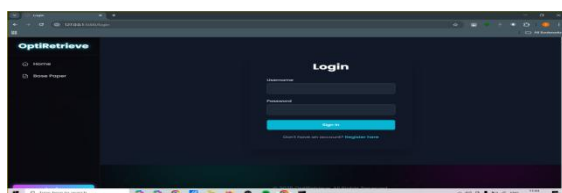


Fig. 3. Secure user authentication portal.

2. Login Page

Figure 3 depicts the system's authentication module. To ensure data integrity and manage computational loads, the retrieval system is safeguarded by a credential-based login mechanism. This interface manages user sessions and role-based access, securely bridging the gap between the end-user front-end and the resource-intensive back-end prediction models.



Fig. 4. Query image upload and processing interface.

3. Predict / Upload Interface

Figure 4 demonstrates the core interactive portal where users submit their target queries. The interface allows users to seamlessly upload a target image into the system. Once an image is submitted here, the backend pipeline is triggered, passing the raw image through the pre-trained Sparse Autoencoder (SAE) to extract its low-dimensional feature vector for similarity comparison against the optimized dataset.

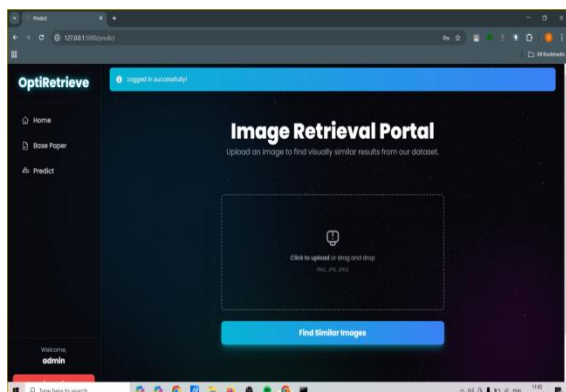


Fig. 5. Visual retrieval results juxtaposed with computed similarity scores.

4. Output / Retrieval Results Page

Figure 5 presents the final output of the MULI framework's retrieval decision model. The interface clearly displays the original query image at the top, followed by the top- K visually similar images retrieved from the massive database. Notably, each retrieved instance is accompanied by a calculated similarity score, demonstrating the model's precision and providing the user with quantitative validation of the retrieval accuracy.

IX. REFERENCES

- [1] H. Wu, J. Guo, and R. Bie, "Multi-use learning instance for optimized image retrieval," *Chinese Journal of Electronics*, vol. 34, no. 3, pp. 1002–1005, May 2025.
- [2] J. Wang, X. F. Fan, N. Shi, et al., "Convolutional neural networks of whole jujube fruits prediction model based on multi-spectral imaging method," *Chinese Journal of Electronics*, vol. 32, no. 3, pp. 655–662, 2023.
- [3] X. Deng, E. P. Liu, S. X. Li, et al., "Interpretable multi-modal image registration network based on disentangled convolutional sparse coding," *IEEE Transactions on Image Processing*, vol. 32, pp. 1078–1091, 2023.
- [4] Y. Y. Zhu, G. Cao, Z. Y. Yang, et al., "Learning relation-based features for fine-grained image retrieval," *Pattern Recognition*, vol. 140, article no. 109543, 2023.
- [5] S. Gkelios, A. Sophokleous, S. Plakias, et al., "Deep convolutional features for image retrieval," *Expert Systems with Applications*, vol. 177, article no. 114940, 2021.
- [6] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, et al., "Autoencoders," in *Machine Learning*, A. Mechelli and S. Vieira, Eds., Academic Press, Cambridge, MA, USA, pp. 193–208, 2020.
- [7] W. Luo, J. Li, J. Yang, et al., "Convolutional sparse autoencoders for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3289–3294, 2018.
- [8] N. Zhang, S. F. Ding, J. Zhang, et al., "An overview on restricted Boltzmann machines," *Neurocomputing*, vol. 275, pp. 1186–1199, 2018.