**Autoencoder-Based Image Similarity Search: Methodology and Evaluation**

**Objective**

The goal is to develop and evaluate an autoencoder-based image similarity search system. This involves extracting compact latent representations of images and retrieving the most similar images based on these representations. The performance of the similarity search system is assessed using metrics such as Precision, Recall, and Retrieval Accuracy.

**Methodology**

**1. Dataset Preparation**

The CIFAR-10 dataset is used for this task. This dataset consists of 60,000 images (32x32 pixels) divided into 10 classes, with 50,000 images for training and 10,000 for testing. Each image belongs to one of the 10 categories (e.g., airplane, car, bird).

* **Training Set:** Used to train the autoencoder model.
* **Test Set:** Used for similarity search and evaluation.

**2. Model Development**

**Autoencoder Architecture:** The autoencoder model consists of two main components:

1. **Encoder:** Compresses the input image into a lower-dimensional latent representation (vector).
2. **Decoder:** Reconstructs the image from the latent representation, ensuring meaningful feature extraction.

The architecture used:

* Input: 32x32 RGB images
* Encoder: Convolutional layers with progressively reducing dimensions
* Latent Space: Compact representation of size 128
* Decoder: Deconvolutional layers to reconstruct the image

**3. Training the Autoencoder**

* **Loss Function:** Mean Squared Error (MSE) to minimize the reconstruction error.
* **Optimizer:** Adam optimizer with a learning rate of 0.001.
* **Training:** Model trained for 20 epochs with a batch size of 128.

**Output:** A trained autoencoder model capable of generating latent vectors for each image.

**4. Similarity Search**

**Latent Vector Extraction:** For each image in the test dataset, the encoder generates a 128-dimensional latent vector. These vectors represent the image features in a compressed format.

**Similarity Computation:** Cosine similarity is used to measure the similarity between latent vectors. The top-k most similar images are identified for each query image.

**Query Process:**

1. Select a query image from the test dataset.
2. Compute its latent vector using the encoder.
3. Compare the query latent vector with all other latent vectors in the dataset using cosine similarity.
4. Retrieve the indices of the top-k most similar images.

**5. Performance Evaluation**

**Metrics:**

1. **Precision:**
   * Fraction of relevant images among the top-k retrieved images.
2. **Recall:**
   * Fraction of relevant images retrieved out of all relevant images.
3. **Retrieval Accuracy@k:**
   * Proportion of queries where at least one relevant image is retrieved in the top-k results.

**Relevance Definition:** Relevant images are defined as those belonging to the same class as the query image.

**Results:** Using this approach, the following metrics are computed:

* Precision
* Recall
* Retrieval Accuracy

**Key Findings**

* **Precision:** Indicates the fraction of correctly retrieved images in the top-k results.
* **Recall:** Measures the system's ability to retrieve all relevant images.
* **Retrieval Accuracy:** Checks if at least one relevant image is retrieved.

These metrics provide a comprehensive evaluation of the similarity search performance and allow comparisons with other approaches.

**Conclusion**

The autoencoder-based similarity search system demonstrates effective retrieval capabilities with competitive precision and recall. Future work could involve:

* Fine-tuning the latent vector dimensions for improved performance.
* Exploring alternative similarity measures (e.g., Euclidean distance).
* Comparing performance with CNN-based and pre-trained models.