Optimizing Facial Recognition with Vector Databases for Deep Embeddings

Abstract

Deep learning–based facial recognition generates high-dimensional vector representations, or *deep embeddings*, which encode unique facial features. Efficient storage and retrieval of these embeddings is critical for large-scale systems. This project proposes a prototype system that uses **vector databases** to store and query facial embeddings. The study also includes a comparison with **NoSQL databases** to evaluate their performance in storing and retrieving embeddings. Metrics for evaluation include retrieval time, precision, recall, and identity separability. The methodology involves embedding extraction from a facial dataset, storage in both database types, implementation of similarity queries, and performance analysis. The expected outcome is a comprehensive understanding of the advantages and limitations of vector versus NoSQL databases in large-scale facial recognition.

1. Introduction

Facial recognition systems based on deep learning map faces into high-dimensional embeddings, which allow for quantitative similarity comparisons. Storing and querying these embeddings efficiently becomes a critical issue as datasets grow into millions of vectors. Vector databases (e.g., Milvus, FAISS, Weaviate) offer specialized indexing and approximate nearest-neighbor (ANN) search to reduce query time for high-dimensional vectors. This project investigates **the performance of vector databases for embedding storage and retrieval**, including a comparison with NoSQL databases.

2. Problem Statement

Deep learning facial recognition systems produce vast numbers of embeddings. Challenges include slow similarity searches in high-dimensional space, scaling to millions of vectors, real-time performance, and the trade-offs between specialized vector databases and general-purpose NoSQL databases. Research Question: How can vector databases optimize storage and retrieval of deep embeddings while maintaining identity separability, and how do they compare to NoSQL databases in performance?

3. Justification

Studying vector databases in facial recognition is timely due to the need for scalable and efficient similarity search. Including a comparison with NoSQL databases provides practical insights into database selection, highlighting strengths and limitations of each approach. Understanding this trade-off helps design AI systems that balance retrieval speed, scalability, and flexibility.

4. General Objective

To evaluate the use of vector databases for storing and querying deep embeddings in facial recognition, focusing on retrieval performance and identity separability.

5. Specific Objectives

1. Extract facial embeddings using a deep learning model.
2. Store embeddings in a vector database with appropriate indexing.
3. Implement similarity search using nearest-neighbor algorithms.
4. Measure retrieval performance: query time, precision, and recall.
5. Analyze intra-class and inter-class distances for identity separability.
6. Compare performance of the vector database with a NoSQL database in terms of storage, retrieval, and query efficiency.

6. Background

Vector databases are specialized systems for storing and querying high-dimensional vectors. They provide efficient similarity search via ANN algorithms, scalability for millions of vectors, and integration with AI pipelines. NoSQL databases, such as MongoDB or Cassandra, provide flexible document storage and distributed scalability but are less optimized for similarity search. Recent studies highlight the efficiency of vector databases for fast and accurate retrieval of embeddings, while NoSQL databases are suitable for flexibility and heterogeneous data management.

7. Theoretical Framework

7.1 Vector Databases

Vector databases store embeddings as vectors and allow similarity queries using algorithms such as HNSW or IVF, reducing query time while maintaining accuracy.

7.2 Deep Embeddings

Embeddings are high-dimensional vectors representing facial features generated by neural networks.

- Euclidean Distance: $d(x,y) = \sqrt{\Sigma(x_i - y_i)^2}$
- Cosine Similarity: $\cos(x,y) = (x \cdot y) / (||x||\ ||y||)$

7.3 NoSQL Databases

NoSQL databases store embeddings as documents or arrays. They allow horizontal scaling and flexible schema but require approximate methods for similarity search.

8. Methodology

1. Select a facial image dataset.
2. Extract embeddings using a deep learning model.
3. Store embeddings in a vector database with indexing for nearest-neighbor search.
4. Store the same embeddings in a NoSQL database using document storage.
5. Execute similarity queries in both database types.
6. Measure retrieval performance (query time, precision, recall).
7. Analyze intra-class and inter-class distances to assess identity separability.
8. Compare vector database and NoSQL database in terms of performance and scalability.

9. Expected Results

Vector databases are expected to provide faster query times and better identity separability. NoSQL databases are expected to demonstrate flexible storage and scalability but slower similarity search. The study will provide practical recommendations for database selection based on system requirements and dataset size.

10. References

- Deng, C. (2024). *A Review of Face Recognition Technologies Based on Deep Learning.*
- Li, M. (2025). *Research and Analysis of Facial Recognition Based on FaceNet, DeepFace and OpenFace.*
- Milvus Documentation (2024). *Vector Database for High-Dimensional Embeddings.*
- FAISS Documentation (2024). *Similarity Search Library for High-Dimensional Vectors.*
- MongoDB Documentation (2024). *NoSQL Database for Flexible and Scalable Storage.*