Tăng tốc tìm kiếm gần đúng bằng thuật toán ScaNN

Nhóm

MỤC LỤC

CHỦ ĐỀ & MỤC TIÊU

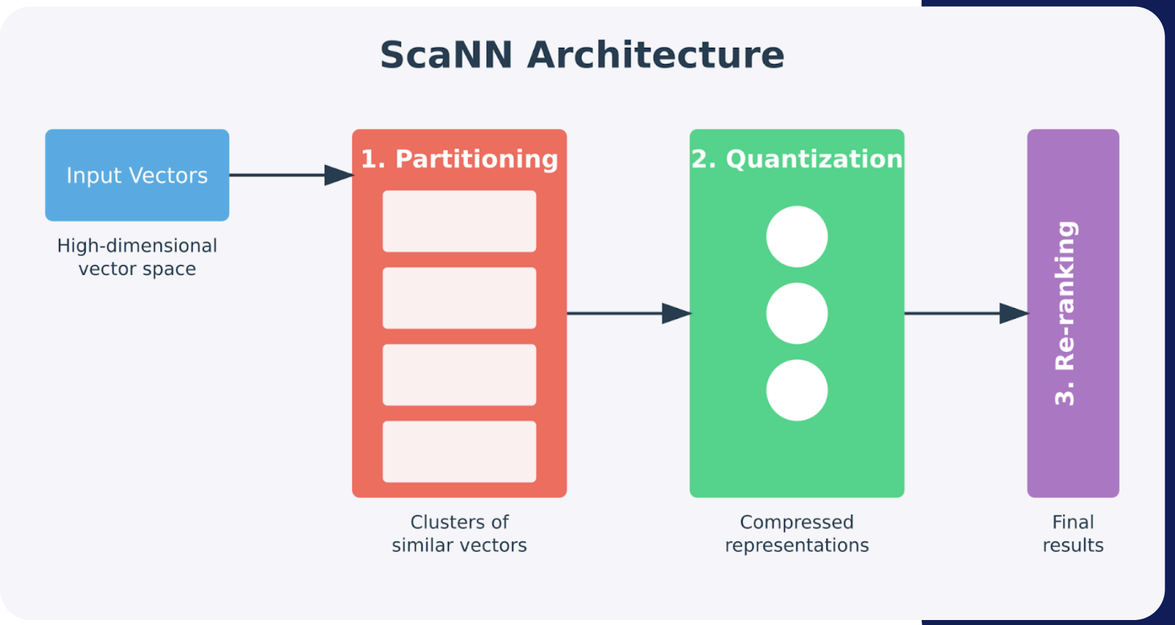
Ứng dụng thuật toán ScaNN để tăng tốc tìm kiếm lân cận gần nhất (Nearest Neighbors) trong không gian vector chiều cao.

MỤC TIÊU

- Nghiên cứu kiến trúc ScaNN: Partitioning – Scoring – Reordering.
- Hiện thực hệ thống tìm kiếm top-K gần đúng bằng ScaNN.
- Xây dựng baseline brute-force để so sánh.
- Đánh giá tốc độ, độ chính xác (Recall@K), bộ nhớ, khả năng mở rộng.
- Thực nghiệm trên ảnh và văn bản (ResNet50, SBERT).

CÁC BƯỚC TRIỂN KHAI

- Tìm hiểu lý thuyết ScaNN và so sánh ANN algorithms (Faiss, HNSW,…).
- Tạo tập vector thử nghiệm:
- Ảnh từ CIFAR-100 → ResNet50 → vector 2048 chiều.
- Văn bản AG News → SBERT → vector 384 chiều.
- Cài đặt ScaNN để tìm kiếm top-K.
- Cài đặt brute-force bằng cosine similarity.
- Chạy đánh giá hiệu năng: thời gian, recall, bộ nhớ.
- Trực quan hóa kết quả bằng bảng & biểu đồ.
- Chuẩn bị báo cáo, video demo và đưa mã nguồn lên GitHub.



CƠ SỞ LÝ THUYẾT SCANN

Tổng quan:

- ScaNN là thư viện ANN do Google Research phát triển (2020).
- Dùng cho tìm kiếm vector trong tìm kiếm ngữ nghĩa, gợi ý, Google Search…
- Mục tiêu: tốc độ nhanh, độ chính xác cao, tiết kiệm bộ nhớ.

PARTITIONING

- Phân cụm (k-means) tạo các vùng dữ liệu (leaves).
- Giảm không gian tìm kiếm → chỉ tìm trong vài cụm gần nhất.

SCORING

- Tính điểm tương tự (dot product / cosine).
- Dùng kỹ thuật Anisotropic Vector Quantization (AVQ) để tăng tốc.

REORDERING

- Tính toán lại chính xác top ứng viên (ví dụ top 100).
- Giữ thứ hạng gần với brute-force nhưng tốc độ nhanh hơn rất nhiều.

ỨNG DỤNG:

- Google Search & Google Lens

ScaNN giúp tìm kiếm ngữ nghĩa nhanh trên quy mô cực lớn bằng cách so khớp vector thay vì từ khóa. Nhờ đó, hệ thống truy xuất hình ảnh và văn bản gần nghĩa trong thời gian thực, kể cả khi không trùng từ khóa.

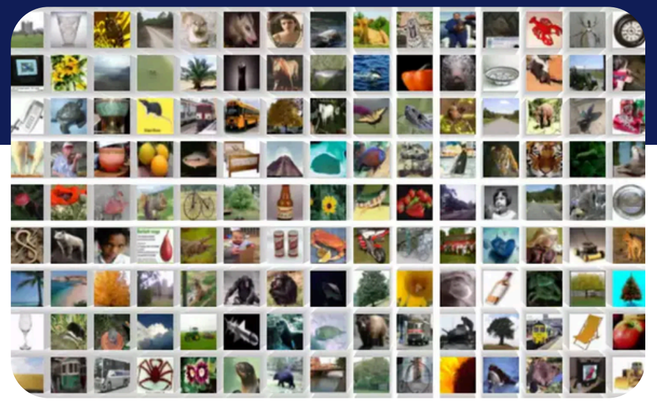
- Hệ thống gợi ý (Recommendations)

Trong mô hình hai-tower, ScaNN dùng để tìm các item có embedding gần người dùng. Điều này giúp gợi ý nội dung nhanh và chính xác hơn, giảm đáng kể chi phí tính toán so với brute-force.

- TensorFlow Recommenders

ScaNN được tích hợp làm backend mặc định cho lớp tìm kiếm top-K của TFRS, giúp mô hình deep learning truy vấn embedding nhanh hơn nhiều lần và hoạt động hiệu quả trên tập dữ liệu lớn.

XÂY DỰNG HỆ THỐNG



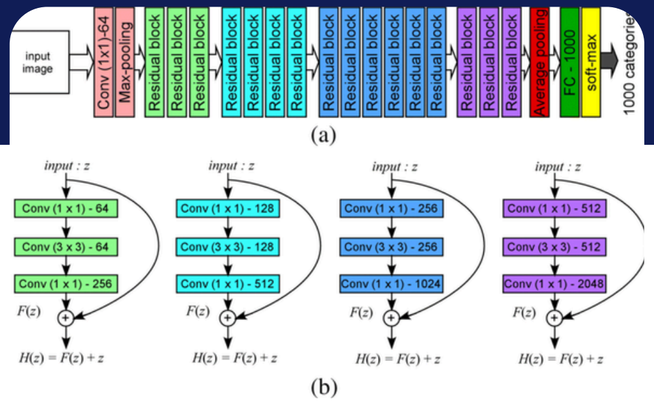
1. TẠO TẬP DỮ LIỆU VECTOR

Ảnh – CIFAR-100:

- Resize 224×224
- Chuẩn hóa ImageNet
- Trích đặc trưng bằng ResNet50 → vector 2048 chiều

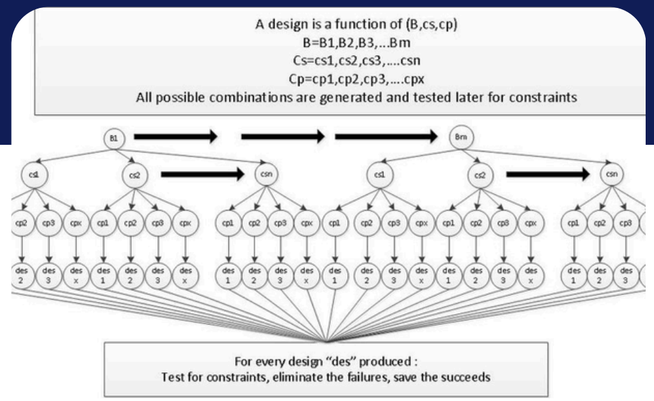
Văn bản – AG News:

- 2000 câu
- Dùng SBERT MiniLM-L6-v2 → vector 384 chiều



2. HIỆN THỰC SCANN TOP-K

- Dùng scann.scann_ops_pybind.builder()
- Thiết lập: num_leaves=100, num_leaves_to_search=10, training sample = 2000
- Quy trình: Query → Embedding → ScaNN search → Top-K.



3. BASELINE BRUTE-FORCE

- Dùng cosine similarity duyệt toàn bộ dataset.
- Kết quả chính xác tuyệt đối (100%).
- Thời gian tăng nhanh khi số vector lớn.

HIỆN THỰC BRUTE-FORCE LÀM CHUẨN THAM CHIẾU

MỤC TIÊU

- Phương pháp brute-force được xây dựng nhằm tạo ra một chuẩn tham chiếu (baseline) cho việc đánh giá hiệu quả của ScaNN.
- Thuật toán này tìm kiếm các vector gần nhất bằng cách duyệt toàn bộ tập dữ liệu và tính độ tương tự giữa từng phần tử và vector truy vấn mà không sử dụng bất kỳ tối ưu hóa hay chỉ mục nào.

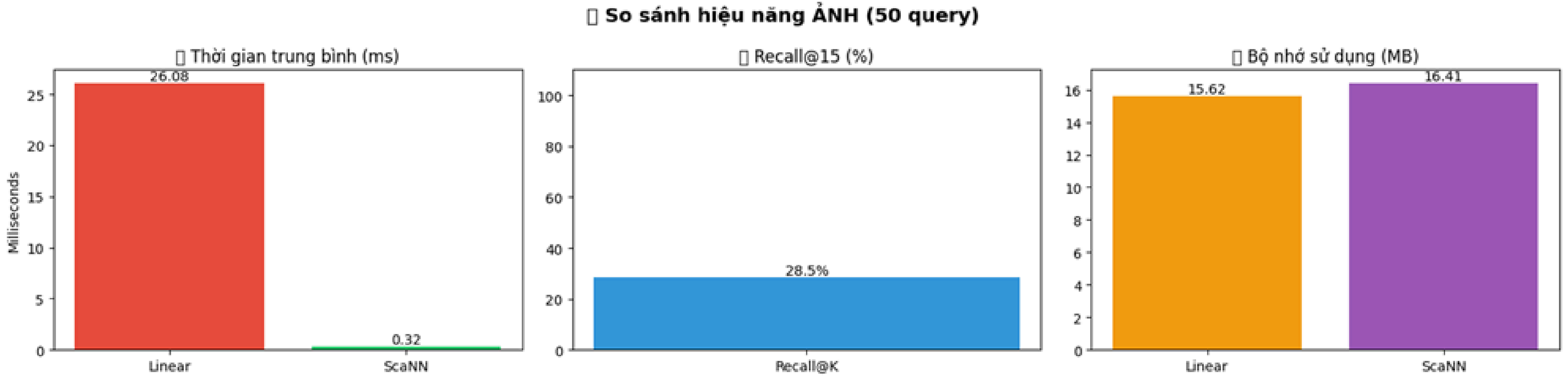
CÁCH HIỆN THỰC

Thuật toán được cài đặt bằng Python sử dụng thư viện scikit-learn (hàm cosine_similarity).

1. Nạp tập vector dữ liệu đã được trích đặc trưng:
2. Nhập vector truy vấn query_vector (được tạo từ ảnh hoặc câu nhập vào).
3. Tính độ tương tự cosine giữa query_vector và tất cả các vector trong tập dữ liệu.
4. Sắp xếp các giá trị độ tương tự theo thứ tự giảm dần.
5. Lấy ra Top-K phần tử có độ tương tự cao nhất.
6. Hiển thị kết quả:
    - Với ảnh: hiển thị K ảnh tương tự nhất kèm độ tương tự.
    - Với văn bản: in ra K câu gần nghĩa nhất.

KẾT QUẢ THỰC NGHIỆM

Phương pháp	Thời gian truy vấn (ms)	Bộ nhớ (MB)	Recall@15 (%)
Brute-force (Linear Search)	17.05	15.62	100
ScaNN (Approximate Search)	0.61	16.41	13.3



BIỂU ĐỒ MINH HỌA

KẾT LUẬN PHẦN ĐÁNH GIÁ

- Tốc độ: ScaNN nhanh hơn brute-force khoảng 28 lần, chứng minh hiệu quả của thuật toán Approximate Nearest Neighbor Search.
- Độ chính xác: Hiện chỉ đạt 13.3% do cấu hình chưa được tối ưu. Việc tăng num_leaves_to_search hoặc thêm bước reordering có thể nâng Recall lên trên 90%.
- Bộ nhớ: Mức tăng nhẹ (≈5%) là hợp lý, phản ánh chi phí lưu chỉ mục phân vùng.

Như vậy, ScaNN cho tốc độ vượt trội so với brute-force, đặc biệt phù hợp khi dữ liệu tăng lên hàng chục nghìn vector. Tuy nhiên, độ chính xác phụ thuộc mạnh vào cấu hình tham số, và cần tinh chỉnh thêm để đạt hiệu quả tối ưu.

THANK YOU !