# Similar Item Retrieval for Auction Items

John AOUSSOU
freelance ML engineer for a private client
Feb.-Apr. 2024

## 1 Introduction and Problem Statement

A client company operates an online auction platform featuring a diverse range of items. A common user need is to find similar items to one they are viewing or have previously seen. This could be the exact same model (perhaps previously sold), the same model at a lower price, or the same model in better condition.

While some items have registered model numbers allowing for simple database lookups, many items lack this information and only have associated images. The task was to develop a prototype solution capable of retrieving similar items based solely on images, specifically for items without model numbers. A key constraint was that the final retrieval code must run efficiently on a CPU.

## 2 Dataset Examination

An initial examination of the client's dataset revealed that a significant portion of the data lacked consistent labeling or sufficient image quality for robust model training across all item types. However, one particular item type constituted the plurality of the data (over 20%) and possessed adequate labeling and image quality. It was therefore agreed to focus the initial research and prototype development on this specific item category.

## 3 Evaluation System

Before proceeding with development, a clear evaluation methodology was established to measure the success of the similarity retrieval system. The following metrics were agreed upon:

- **Top-1 Rank Accuracy:** The percentage of queries where the exact same item model was returned as the single best match.
- **Top-5 Rank Accuracy:** The percentage of queries where the exact same item model was present within the top 5 returned matches.

## 4 Solution Proposal: Initial Approach

The proposed solution involved using image embeddings extracted from a Convolutional Neural Network (CNN) combined with a k-Nearest Neighbors (k-NN) algorithm for retrieval. Cosine similarity was chosen as the distance metric for comparing embeddings in the k-NN search, as it is effective for high-dimensional feature vectors like image embeddings.

EfficientNet B0 was selected as the CNN architecture due to its favorable balance of accuracy and computational efficiency (light weight), making it suitable for potential CPU deployment. Using the pre-trained weights of EfficientNet B0 without any task-specific fine-tuning, the initial performance was evaluated:

- Top-1 Rank Accuracy: 53%
- Top-5 Rank Accuracy: 69%

These baseline results were considered reasonable, demonstrating the potential of the approach.

## 5 Fine-tuning the Embedding Model

To enhance the accuracy of the similarity model, fine-tuning the CNN specifically for this retrieval task was explored. A Siamese network architecture trained with Triplet Margin Loss was chosen for this purpose. This approach learns to pull embeddings of similar items (same model) closer together while pushing embeddings of dissimilar items further apart in the feature space.

The dataset preparation for fine-tuning involved the following steps:

1. Organize the images into folders named according to their corresponding item model number. Within each folder, images were saved using their unique item ID number.
2. Split the available item models into training and validation sets. Models with 3 or more distinct item images were allocated to the training set. Models with exactly 2 distinct item images were designated for the validation set.

Fine-tuning deep learning models can be susceptible to overfitting, especially with limited data. To mitigate this, the model was trained conservatively:

- Training was conducted for only one epoch.
- A low learning rate was used.

**Note:** Given the limited data and single-epoch training strategy, the distinction between the training and a separate validation set for hyperparameter tuning or early stopping was deemed less critical for this prototype phase. The primary goal was to demonstrate improvement over the baseline.

The fine-tuning process resulted in a notable increase in accuracy (approximately 10% absolute improvement in Top-1 and more in Top-5):

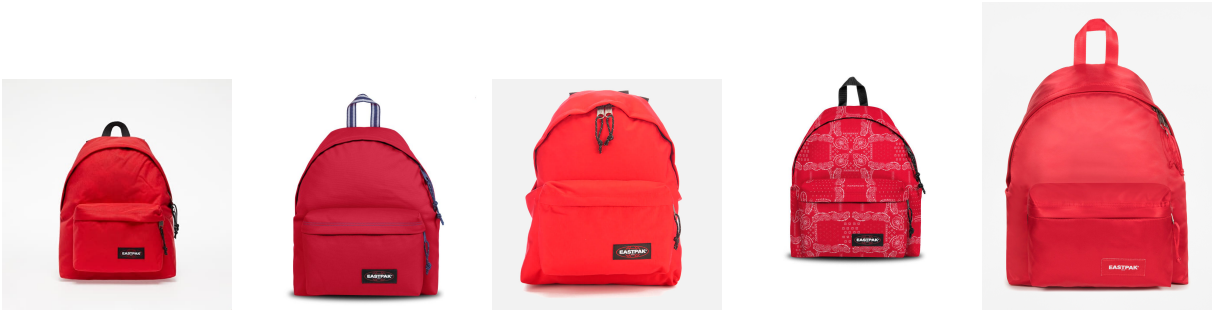
- Top-1 Rank Accuracy: 62%
- Top-5 Rank Accuracy: 81%

## 6 Other Considerations and Observations

Manual inspection of the results and model behavior yielded further critical insights:

- **Impact of Data Labeling Errors:** Visual inspection confirmed that in numerous instances where the top-1 result did not match the database label, the discrepancy was due to the database label itself being incorrect for the query item. The model frequently identified the *visually* correct item as the top result, despite the label mismatch penalizing it in the formal evaluation. This strongly indicates that the actual accuracy of the model in retrieving identical items is significantly underestimated by the reported metrics (62% Top-1, 81% Top-5, see Figure 1) due to inherent errors in the ground truth labels used for evaluation. The model's practical performance is therefore expected to be much higher.


Query Item



Rank 1 | Rank 2 | Rank 3 | Rank 4 | Rank 5

Figure 1: An illustrative example using placeholder images: A query item (top) and the corresponding top 5 similar items retrieved by the fine-tuned model (bottom row, ranked left-to-right). This visualizes the performance level reflected in the 81% Top-5 accuracy.

- **Shift in Feature Importance (Shape vs. Color):** A qualitative analysis suggested a shift in the features used for discrimination before and after fine-tuning. The initial model (pre-trained EfficientNet B0) appeared to rely more heavily on overall item *shape* for similarity. After fine-tuning with the Siamese network on items of the same category (and thus often similar shape), the model became noticeably better at discriminating based on subtle *color* differences. This aligns with expectations: the pre-training dataset for EfficientNet (ImageNet) contains categories largely distinguishable by shape, whereas fine-tuning on items with highly similar shapes necessitated learning finer-grained features like color for effective discrimination within the category.
- **Generalization Potential:** Preliminary tests suggested that the fine-tuned embedding model, although trained on one primary item type, showed promising generalization capabilities when applied to different types of items not included in the fine-tuning dataset.

These observations highlight the robustness of the learned visual features, underscore the critical impact of data quality on performance evaluation, and illustrate how fine-tuning adapted the model's feature focus to the specific task requirements.