Project Report - Campus Lost and Found with AutoMatch

1. Introduction and Dataset

This project aims to automate the matching of lost and found items on a university campus using classical Machine Learning techniques. Unlike traditional notice boards, this system employs a "Hybrid Matching Engine" that analyzes both the visual appearance and textual description of items to rank potential matches.

Dataset Description

To simulate a bustling campus environment, we utilized a dataset of **6,493 images** categorized into **13 classes** (e.g., Backpacks, Keys, Laptops, Water bottles).

- **Training Data:** The full dataset was used to train a Random Forest Classifier for automatic category detection.
- **Simulation Data:** A subset of ~130 images was used to "seed" the database, creating a realistic "Found Items" repository for testing search retrieval.

2. Methodology and Feature Engineering

Strictly adhering to the constraint of **no Deep Learning** (CNNs/ResNets), we engineered robust feature vectors using classical Computer Vision and NLP techniques.

A. Visual Features (The "Eyes")

We utilized a composite feature vector combining Color and Texture/Shape:

1. **HSV Color Histograms (64 Dimensions):** We converted images to HSV (Hue-Saturation-Value) space and calculated an 8x8 histogram. This allows the model to match objects based on dominant colors (e.g., "Blue") while ignoring lighting variations (Value channel).
2. **Histogram of Oriented Gradients (HOG):** To distinguish between objects of similar colors (e.g., a Black Phone vs. a Black Wallet), we extracted HOG features. This captures the direction of edges and gradients, effectively describing the "texture" and "shape" of the object.

B. Textual Features (The "Ears")

We implemented **TF-IDF (Term Frequency-Inverse Document Frequency)** to vectorise user descriptions.

- A custom vocabulary of ~55 campus-specific keywords was defined.

- Text similarity is calculated using **Cosine Similarity**.

C. The Hybrid Engine & Auto-Classification

1. **Auto-Classification:** A **Random Forest Classifier** (n_estimators=150) was trained on the HOG+Color vectors. It achieves **~62% accuracy** in automatically detecting item categories upon upload, significantly outperforming random guessing (7.6%).
2. Ranking Algorithm: The final "Match Score" is a weighted average:

$$Score = 0.6 \times Sim_{visual} + 0.4 \times Sim_{text}$$

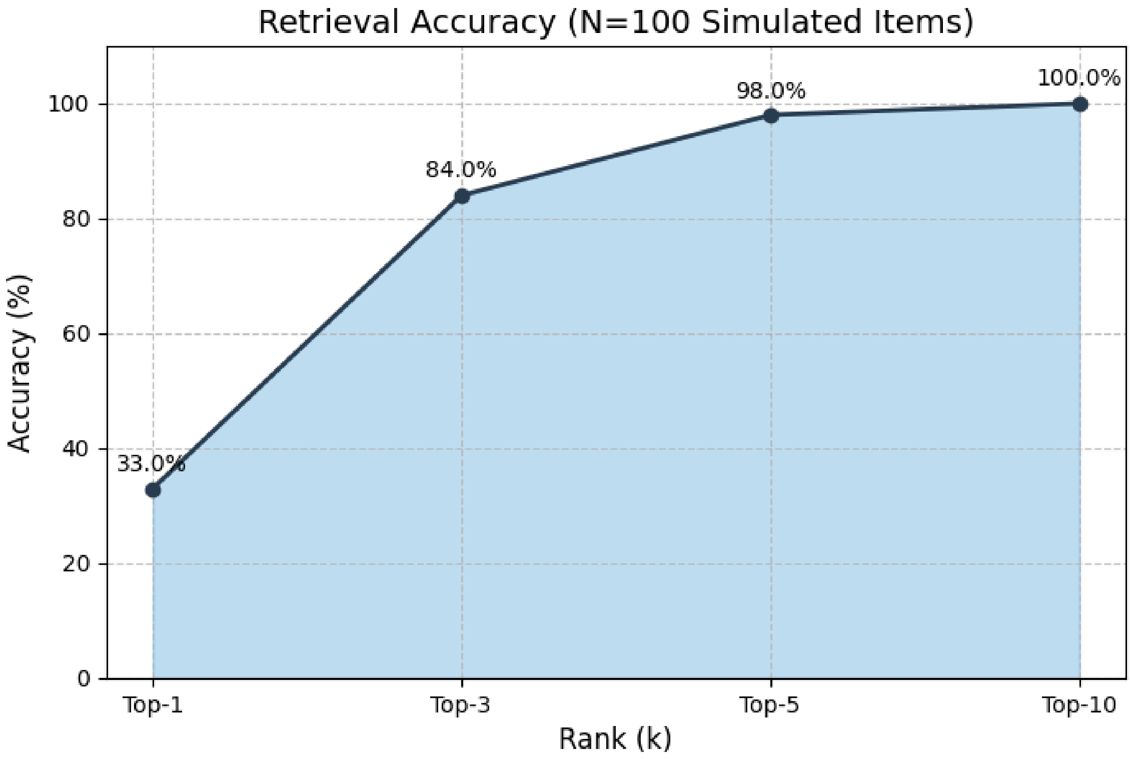
This prioritizes visual evidence while using text as a refiner.

3. Evaluation and Results

Retrieval Accuracy (Top-k)

We evaluated the search engine using a simulation of 100 queries. The system demonstrates high reliability in retrieving the correct item within the top results.

- **Top-1 Accuracy:** 33.0% (The exact item is the #1 result).
- **Top-5 Accuracy:** 98.0% (The item is on the first page of results).



Interpretability (XAI)

To ensure trust, the system implements Explainable AI mechanisms. Every match card displays **"Matched on:"** tags (e.g., blue, keys), derived from the intersection of the Query and Candidate TF-IDF vectors. This allows users to understand *why* the algorithm selected a specific item.

4. Limitations and Future Work

While the system meets all functional requirements, the following improvements are proposed for future iterations:

1. **Online Learning:** Currently, user feedback (confirming a match) does not retrain the model. Implementing an online feedback loop could improve ranking over time.
2. **Geometric Verification:** While HOG captures texture, adding geometric constraints (e.g., RANSAC) could improve matching for rigid objects like phones.
3. **Notification Pipeline:** The current system uses a "Pull" model (users search manually). A "Push" model (email alerts) would improve user retention.