Image Embedding for Product Labeling

Couture Vision

January 22, 2025

Project Objective

Goal:

- Based on a photo of an article, match the reference image of this article.

Example:

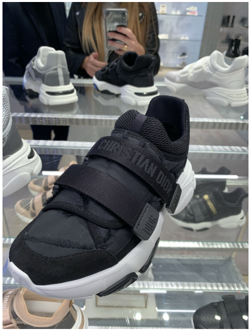

Figure: Test Image

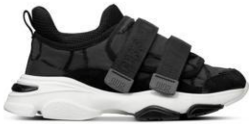

Figure: Reference Image

Test set labeling

- We labeled the test set manually, using embeddings to suggest potential matches.
- The test set contains 80 images.
- Some items don't have a corresponding image in the DAM.
- Some items are very similar to others, making classification challenging.

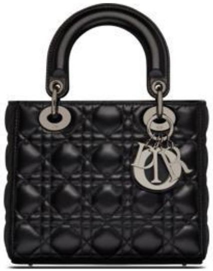
Figure: Black bag

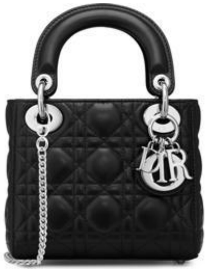
Figure: Another black bag

Project Overview

- We tried a lot of models to extract embeddings: Microsoft's ResNet50, Google's ViT Patch-16, OpenAI's CLIP, Meta's DINOv2 and ViTMSN.
- Cosine similarity is used to find the best matches between images. Better than Euclidean distance. We also tried a mix of both.
- We replaced `rembg` background removal by `RMBG 2.0`.
- All images were cropped and zoomed to the object.

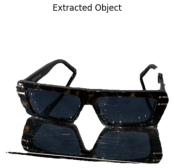

Figure: Background removal using `rembg`

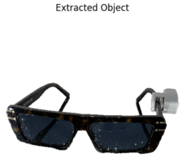

Figure: Background removal using `RMBG 2.0`

Data augmentation with a 3D generation model

- We used **TRELLIS** for generating 3D models for the 2700 DAM assets.
- We used **Blender** to render the 3D models in 8 different angles each.

Figure: Input for TRELLIS

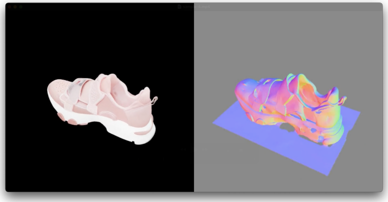
Figure: TRELLIS generated 3D model

Data augmentation with a 3D generation model

Figure: 3D renders of a bag

The performance improvements

- Adding embeddings of the 3D renders improved the Top-1 performance from 36% to 0.42%.
- Averaging 2D and 3D embeddings improved the Top-1 performance from 0.42% to 0.52%.
- We then removed all 2D embeddings and only used 3D embeddings, which improved the Top-1 performance from 0.52% to 0.56%.
- We used gradient descent on 25% of the test set to tune the weights for merging 2D and 3D embeddings, as well as the feature selection weights, which improved the Top-1 performance from 0.56% to 0.61%.
- Final results: Top-1: 0.61%, Top-3: 0.79%, Top-5: 0.84%.

Web Interface with Gradio

- Develop a simple UI using Gradio.
- Users can upload a picture and get top-X matches.
- The UI will display the reference DAM image with its ID, the top-X matches, and a 3D render of the item.

Using NLP for Image-to-Text and Matching

- **Image-to-Text Conversion**:
    - Generate descriptive captions for images to capture key attributes like color, material, and unique features.
- **Text Embedding and Matching**:
    - Convert captions into embeddings and store them in a vector database.
    - Perform similarity search to find the closest matches to client-uploaded images.
- **Combination with Image Embeddings**:
    - This approach can be enhanced by combining text-based embeddings with image-based embeddings for improved accuracy.

Improvement Attempt Using NLP

There are some challenging cases, with **color** or **texture**:

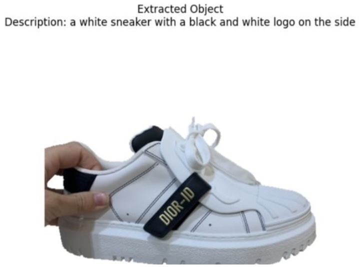
Extracted Object
Description: a white sneaker with a black and white logo on the side

Figure: Example: Color mismatch

Blip Description: *"A white sneaker with a black and white logo on the side"*

Using NLP for Image-to-Text and Matching

Steps :

1. Generate descriptions using the **BLIP model**.
2. Convert descriptions into **text embeddings**.
3. Combine **text embeddings** with 2D image embeddings and 3D-rendered image embeddings
4. Use the **mean embedding** for improved matching.

Improving Top-1 result

- Use **more recent BLIP models** for better descriptions.
- Question prompt:

  What is the object in this image? What is its color? What is its texture? What is a specific detail about it?
  Answer in the format: [object, color, texture, specificity].

Proposed Method

- Select the top-5 DAM candidates selected after applying the cosine similarity on the embedding (without the BLIP embedding).
- Apply BLIP to the top-5 and the test image to generate a structured description.
- Keep the candidate with the **highest overlap** in responses with the test image.