In this document, I will provide insights into my thought process during the Miros homework.

Background Removal

Initially, I considered using an existing REST API for background removal, as various tools are available for this task. Implementing an API would be a practical approach during prototyping, given its speed. However, since the purpose of this exercise was to demonstrate my abilities, I opted against using an existing API.

This decision led to a new challenge. Most background removal services/models are made to remove the background surrounding a person and not everything except for a single piece of clothing. This made the problem significantly more complex, with no ready-made solutions available.

Given this challenge, I explored the possibility of fine-tuning an existing semantic segmentation model. With models like YOLO (bounding boxes only) and Mask R-CNN, as well as datasets like DeepFashionV2, this approach seemed feasible if time wasn’t of the essence. I found a repo (https://github.com/simaiden/Clothing-Detection) targeting this goal, but it no longer had weight files. Implementing this method would be time-consuming and thus beyond the scope of the first prototype.

Instead, I opted to begin by implementing a background removal model that retains the person in the image, as this would partially tackle the problem and enable me to advance to the search service.

For future development, I would fine-tune a semantic segmentation model, ideally one that identifies the precise pixels of an object, rather than just the bounding box.

Problems and areas of improvement:

- Models (persons) are not removed, the solution is detailed above.
- Some products are misclassified as part of the background and thus removed. This is especially prominent in close-ups. This problem is counterbalanced by the fact that every product has multiple images. Fine-tuning models is a potential long-term solution.

NLP Search Engine

The prototyping phase for this task was more straightforward due to the existence of models that rank similarity between text and images. I decided to use OpenAI's CLIP because it was easy to implement and demonstrated its potential as a powerful tool. Although CLIP is not fine-tuned for fashion data, its zero-shot nature still makes it highly valuable for our purposes.

Following model testing, I determined it was suitable for a prototype. The model performed well with prompts like "pink dress" and "stripes." However, due to the imperfections in our background removal service, prompts such as "jeans" yielded false positives. To resolve this, I would improve the background removal model, as I have identified a possible solution already.

Problems and areas of improvement:

- The “jeans” problem, as discussed above.
- Confidence scores for most searches are relatively low, suggesting the model is not fine-tuned to the data. Despite this, the prototype remains functional, as the relative difference between search results is sufficient to prioritize relevant products. This might also be solved by better background removal.
- The HTTP request takes 5+ seconds, primarily due to model inference time. This issue could escalate as the number of images/products increases. Potential solutions include using fewer images per product, lower-resolution images (compression), or considering an alternative model for larger loads.
- For a full-fledged product, the model should incorporate non-visual data such as product names and descriptions. Certain attributes, like material, may not be discernible from images alone.

Randal Annus