

## CLASSIFICATION OF FASHION AND APPAREL PRODUCTS

Our new task involves the classification of fashion and apparel products. With the vast variety of fashion and apparel items available, we aim to accurately detect these items and assign them to their respective classes. In our research, we found that the YOLOv8 pretrained model best suits our needs due to time constraints. After studying the YOLOv8 documentation from Ultralytics, we decided to proceed with this model.

We began by writing scripts and training the model using a dataset sourced from Kaggle, which comprises 12,000 images. Initially, we trained the YOLOv8 model for 100 epochs, but the results did not meet our expectations, with poor mAP (mean Average Precision) values. Undeterred, we extended the training to 200 epochs, although this significantly increased the training time. Despite this, the model's performance still fell short.

Upon further analysis, we identified issues with the dataset, notably its 88 classes, which led to poor prediction accuracy. To address this, we reduced the classes to four: top, bottom, shoes and one\_piece, using a Python script for mapping. However, even after retraining the model with the refined dataset, the results remained unsatisfactory.

To enhance the model's accuracy, we transitioned from the YOLOv8n to the YOLOv8m model, which has 25 million parameters compared to the 3 million of YOLOv8n. Training the YOLOv8m model for 100 epochs took approximately 10 to 15 hours, with each epoch lasting around 8 minutes. Despite the increased complexity of the model, the performance improvements were not significant.

Realizing that our dataset was unbalanced, with insufficient images of shirts and tops, we sought to rectify this issue. After further research, we found additional datasets, but they contained poor quality images, and their annotations were incompatible with YOLO. This unbalanced dataset posed a significant challenge, as obtaining a high-quality, balanced dataset was crucial for improving the model's performance.

After gathering images from external sources, we identified classes with fewer images and created bounding boxes for those images using scripts, saving their class labels and coordinates. This augmented our dataset to achieve better balance, and we proceeded to train the [YOLOv8m.pt](#) model for 200 epochs, aiming to improve its performance.

After initially transitioning to the YOLOv8m model due to dataset imbalance, we returned to training with the YOLOv8n model for 100 epochs, as the former proved too time-consuming. Despite extending training to 200 epochs, the results remained unsatisfactory. Recognizing that our model struggled with predicting multiple objects due to single bounding boxes in our dataset, we opted to create our own dataset instead of relying on open-source ones.

Using segmented images with masks, we developed Python code to generate bounding boxes for multiple classes and extract their coordinates. These coordinates, initially non-normalized, were converted using code, and the model underwent training on these augmented images and text files using the YOLOv8n architecture.