Ci task GDAŃSK UNIVERSITY OF TECHNOLOGY

TASK Quarterly 27 (2) 2023

Object detection and multimodal learning for product recommendations

Karolina Selwon[1]
Faculty of Electronics
Telecommunications and Informatics
Gdańsk University of Technology
ul. Narutowicza 11/12, Gdańsk, Poland

Paweł Wnuk
Shopai sp. z o.o
ul. Roździeńskiego 2A, Piekary Śląskie, Poland

https://doi.org/10.34808/1ayy-v106

Abstract

This study showcases how deep learning can be applied to automated information extraction in fashion data to create a recommendation system. The proposed approach is an algorithm for recommending multiple products based on visual and textual features, ensuring compatibility with query items. The object detection model can detect many products across different garment categories. The study utilized public e-commerce datasets and trained models using deep learning methods. The compatibility model has shown promising results in automating recommendations of compatible products based on user interests. The study experimented with multiple pre-trained feature extraction models and successfully trained the object detection model for fashion article detection and localization tasks. Overall, the goal is to deploy the method to enhance its effectiveness and usefulness in providing a satisfying shopping experience for e-commerce users.

Keywords:

object detection, multimodal learning, features extraction

[1] Corresponding author. E-mail: karselwo@pg.edu.pl

1. Introduction

E-commerce stores meet the expectations of users by offering advanced product search tools. Suppose someone spends significant time scouring various fashion websites searching for the perfect outfit. In that case, they may be interested in exploring a cutting-edge platform that leverages advanced technology to offer personalized fashion recommendations based on deep learning and visual search. Modern platforms can completely transform the approach to shopping for clothes, providing highly accurate and helpful results. Recent research advances in artificial intelligence can improve automated information extraction in applications such as e-commerce. We can get advanced product search tools by combining the results of many techniques. A tailored fashion recommendation system can lead to increased revenue and better customer engagement and satisfaction. This work shows that recent deep learning advances can be employed in applications for automated information extraction using properly selected fashion data. This paper aims to create an application that recommends multiple products to users based on product visual and textual features. The user can get recommendations for clothes collections from the product catalog by providing an input photo of clothes or just a particular item. In addition, the application ensures the compatibility of recommended products with the query items, which aims to ensure that clothing products fit each other. What makes this approach innovative is the ability to detect many products in a photo and recommend similar and compatible products to the user for different garment categories. To assess this study's contribution, we first highlight relevant datasets and methods for data extraction. In this approach, public e-commerce datasets are used to extract product information automatically. Next, the deep learning methods are used to train the models, and the evaluation results are described. Finally, the proposed system for recommending sets of clothing products for user search is delivered, and results are provided.

2. Related work

Much work on fashion domain data has been carried out already. Thus for this article, the focus is on the selection and optimization of the most applicable techniques needed to build the system. This section discusses related works that form the background for free approaches proposed in this article: fashion object detection, fashion compatibility learning, and similar product recommendations.

Fashion object detection

The first essential element of the system is the recognition of a piece of clothing from a photo. The article [1] considers an image detection algorithm for recognizing clothing styles. Experimental results on the DeepFashion [2] dataset show that the YOLO [3] model achieves tempting results regarding recognition accuracy and detection speed better than other machine learning algorithms. YOLO is an object detection system that applies a single neural network to the entire image, dividing the image into regions and finally predicting bounding boxes and probabilities for each region.

Fashion compatibility learning

The crucial component of the system is an algorithm that can recognize the features of a part of clothing and, based on these features, decide whether the elements are compatible with each other, i.e., they fit each other in terms of features and style. Previous works have used a variety of photo and text-based learning techniques to train a compatibility model. The authors [4] used feature extraction from pre-trained visual models and text representations, e.g. bag of words, to learn embeddings space that respects item type and notions of items similarity and compatibility. Han et al. [5] learn the visual representation of each item in an outfit by feeding features into an LSTM (long short-term memory) to jointly reason about the outfit as a whole. Authors [6] learn nonlinear embeddings that recognize multiple notions of similarity within a shared embedding space using shared feature extractors. Authors [7] use knowledge graph representation for compatibility learning. Moreover, multiple works show the advantage of multi-modal embedding learning over modeling a single modality, i.e., visual or text [8].

Fashion retrieval systems

In addition to the techniques of extracting information from data, it is essential to use them in practice, where the end users of the e-commerce store can use them for product search. Therefore, it is crucial to implement an end-to-end retrieval system. Authors [9] discuss a novel technique to assist consumers in finding similar fashion products. The system uses deep learning techniques for human key point detection, pose classification, object localization, and detection. Finally, the proposed retrieval system recognizes similar products in the product catalog. The authors [10] proposed an industrial-scale solution for compatibility-based fashion recommendation. They introduce a fashion outfit dataset and learn the item's style recognition and recommendation model. They deployed and evaluated the model in an end-to-end recommendation system that performs retrieval from a diverse corpus of shopping products.

3. Proposed approach

The goal of this article is to present an end-to-end system that uses the existing achievements in the field of fashion recommendation and suggests their extension by employing the latest data extraction models. The proposed solution is a content-based recommendation system that includes modules to perform specific tasks that help achieve and improve the overall system performance. The system consists of the following main components: object detection module, product compatibility model, and product catalog search, as shown in Figure 1.

Detection and location of fashion items from the query image

This module aims to identify products in an image and localize them in bounding boxes. The image can include multiple fashion items and accessories. The model identifies the correct locations of one or more objects with bounding boxes that correspond to a rectangular shape around the objects and assigns an appropriate category label.

Fashion items compatibility modeling

This module aims to model the compatibility of clothing products from different categories based on features extracted from text descriptions and visual features. Products on shopping websites most often contain textual and visual data. Therefore, a recommendation model for complete sets of clothing products was created to implement this system, considering the multimodality of data. In the case of the system proposed in the article, the model was trained for objects for the upper and lower clothing, footwear, bags, and accessories categories. The metrics used to evaluate the system are similar to other compatibility modeling studies [4]: the outfit compatibility prediction (AUC) and the Fill-in-the-blank (FITB) metrics.

Generating product embeddings and searching in the product catalog

The last module combines the capabilities of two models: detection of clothing items and recommendations of compatible clothing items. Once the bounding boxes for a particular product are obtained, the part of the image corresponding to that product is cropped, and the encoder creates embedding for the compatibility model. Then, the product compatibility model is used to find products from multiple categories compatible with the selected item. Based on the calculated similarity (based on visual features) between the query's fashion article embeddings and the products in the database, we can get a set of multiple most compatible products for the query from categories different from the query item's category. An embedding module was built to embed

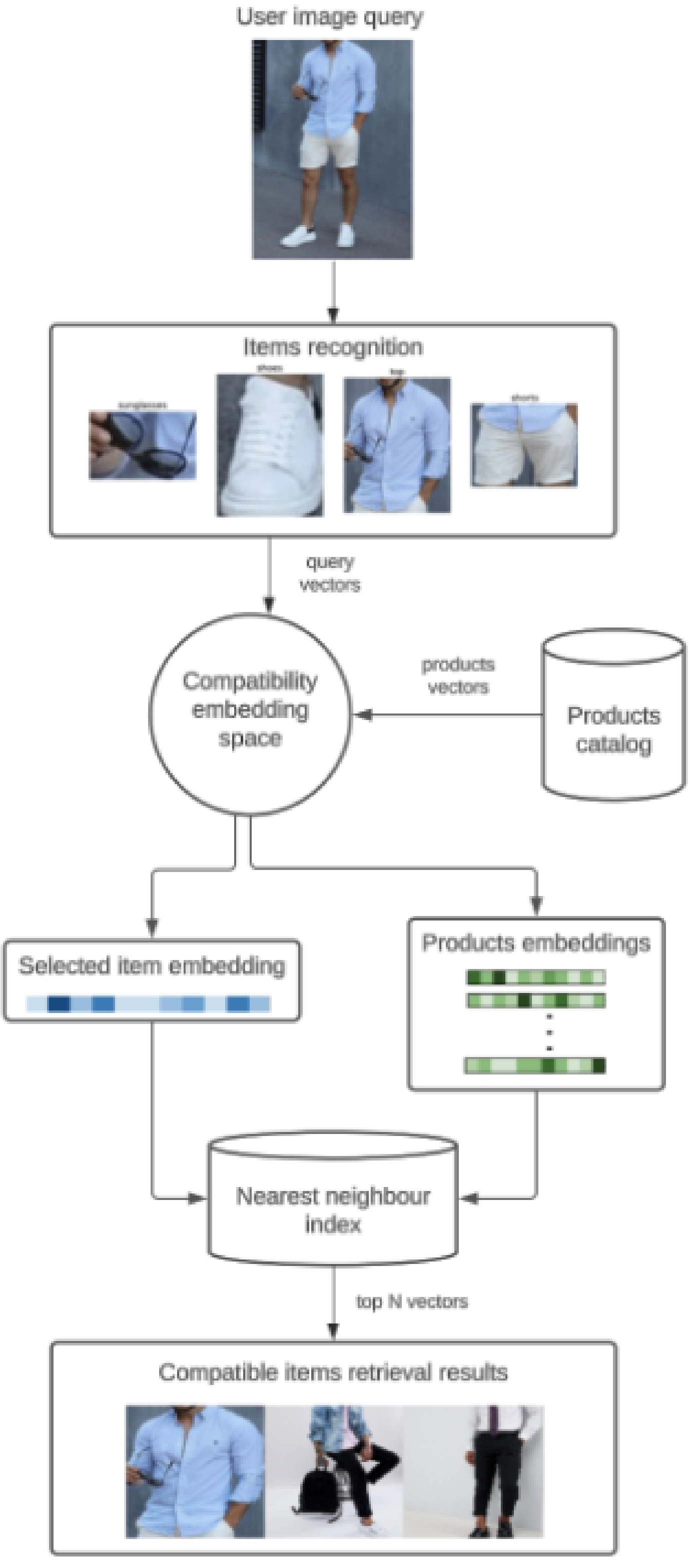


Figure 1: System overview. The user of the system sends a photo of an image with an outfit of clothing. The system detects each object in the picture and its category. For a piece of clothing selected by the user, a specified type, a vector is created using the compatibility model encoder. Next, the most compatible clothing items of different categories in the product catalog are searched. As a result, the user receives recommendations for a full set of clothes and a complete outfit.

product images into a compatibility embedding space that can be used to retrieve compatible products with the k-nearest neighbors search algorithm. In this manner, a complete set of products is obtained as a result of the recommendation.

The advances beyond existing works in this study lie in several key aspects. Unlike previous research that primarily focused on either object recognition or compatibility, this study integrates both elements to develop a comprehensive fashion recommendation system. While many recommendation systems focus on suggesting products based on individual features, this study emphasizes product compatibility based on multimodal features, which is especially important in fashion, ensuring that the recommended items align with the user's preferences. The study proposes a recommendation system that combines visual and textual features to ensure compatibility between query items and recommended products. Using an object detection model capable of recognizing and localizing fashion articles across diverse garment categories is a significant step toward automating the processing and recognition of fashion data. Fine-tuning the detection model for the e-commerce fashion dataset enhances accuracy in identifying and localizing domain-specific items. The study demonstrates the applicability of its models using publicly available e-commerce datasets, thereby providing a practical and reproducible benchmark for future research in this field. Unlike theoretical studies, the focus on deployment highlights the practical utility of this work.

4. Experimental results

4.1. Fashion object detection model

The fashion items detection module serves the purpose of identifying and categorizing different articles of clothing and accessories in images.

Dataset

For the purpose of this work, ModaNet [11] and DeepFashion2 [2] datasets were used, which contain the data about the clothing products in the photos and their locations. ModaNet is a street fashion image dataset consisting of object annotations in images. Each tagged object is associated with a tag from 13 fashion categories. DeepFashion2 features images of clothing, both from commercial retail stores and from consumers. The DeepFashion2 set consists of 20,000 images in the training set and 10,948 in the test set. The ModaNet set consists of 41,803 photos in the training set and 5,225 in the test set. Both datasets have been unified for class naming consistency. After unifying the nomenclature of the categories for these sets, the location of the envelope and the classification for 12 types of clothing were obtained.

Experimental details

For this article, the Yolov5 model was successfully trained for the fashion article detection and localization task. The baseline model was fine-tuned for 50 epochs with an AdamW optimizer.

Table 1: Detection evaluation metrics. Mean average precision results by category calculated for the validation set.

Label	Modanet	DeepFashion2	DeepFashion2 + Modanet
bag	0.87		0.87
belt	0.74		0.74
dress	0.84	0.85	0.85
hat	0.94		0.94
outerwear	0.83	0.77	0.83
pants	0.96	0.94	0.96
scarf/tie	0.60		0.63
shoes	0.87		0.87
shorts	0.87	0.86	0.88
skirt	0.86	0.80	0.86
sunglasses	0.91		0.91
top	0.80	0.92	0.80
all classes	0.84	0.86	0.84

Evaluation

Table 1 provides a comprehensive list of fashion article categories that were used in the training sets. The table also shows the mean average precision results for each set separately and after combining the sets. The lack of empty cells in Table 1 is due to the lack of specific categories in the DeepFashion2 dataset. Additionally, Figure 2 showcases the detected article types for the image in question.

Observations

During the analysis of the object detection module analysis, a few observations were noticed that are worth mentioning. First, the model identifies some products correctly with a low threshold, such as shoes at 0.02, as shown in Figure 2. However, there are cases where the same product is identified multiple times with different classes, and the identified class with a higher score may be incorrect. This highlights the need for human re-labeling of detected images to improve classification quality. After verifying the selected images, it was found that lowering the threshold positively affected detection accuracy. It is important to note that a product classification is considered correct if it includes only one class. Unfortunately, there are instances where many labels are still present, but not all of them were marked correctly. Additionally, there are cases where a higher score is associated with the wrong class. Despite these challenges, the object detection model's overall performance was satisfactory for the pipeline.

4.2. Fashion compatibility model

The baseline model is trained to map the product image and description into a common multimodal "complementary" embedding space in which compatible products are close to each other and non-complementary products are far apart. For inference, the distances between the outfit products and the candidates are used to select the most compatible candidate.

Dataset

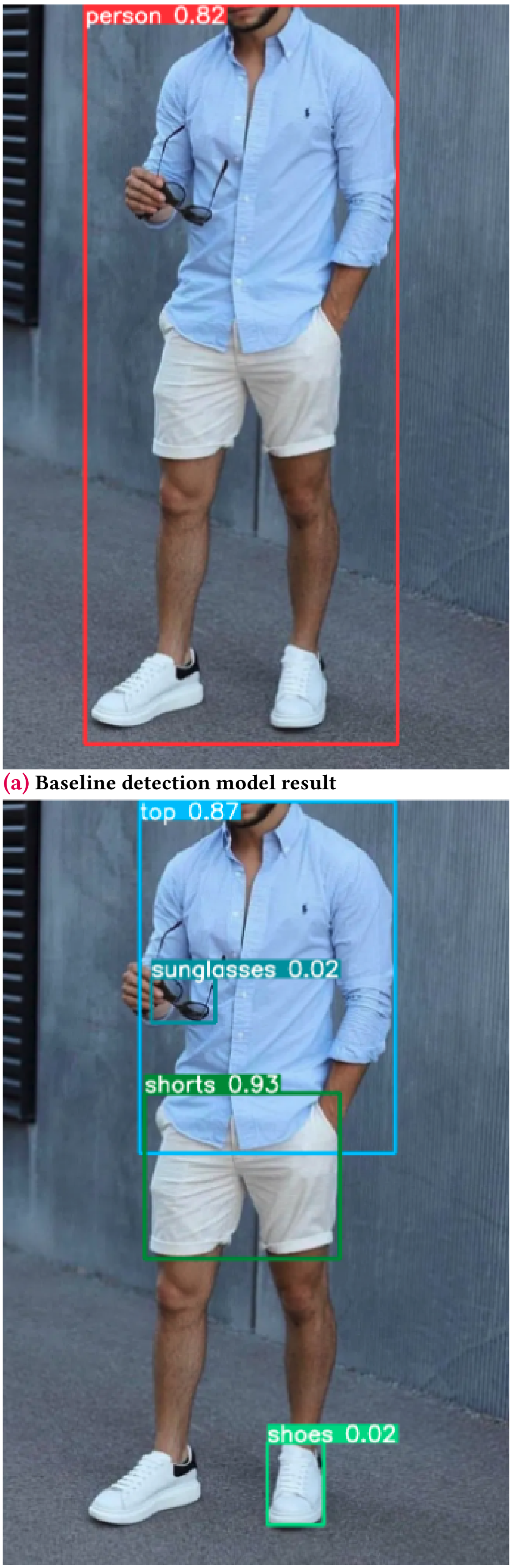
Polyvore type-labeled fashion outfit dataset [12] was prepared for the experiments with five types of clothing classification: top, bottom, shoes, bag, and accessory. For this study, there was a focus on outfits with elements from all categories. Outfits in the dataset with less than five types were filtered out to ensure each outfit in the dataset has the same number of and from different categories. The dataset of outfits was split into 6046 for training, 2015 for testing, and 2015 for validation.

Triplets sampling

This approach proposes a procedure for minimizing the distance of vector representations of compatible products through triplet data sampling. Thus, triples of images are needed, of which the first two are semantically similar and the third dissimilar to the first two. Triplet sampling of the dataset was defined as a set of products with the following relationships. Let first denote the embeddings of the three images as (x_a, x_p, x_n). The anchor product x_a belongs to one category, while the positive x_p and negative x_n products are of another category. The pair of anchor products and positive products are compatible with each other, they exist in a compatible set. On the other hand, the pair of anchor and negative products do not exist in a common set, so they are incompatible. In addition, an average pooling for anchor x_a embeddings was proposed, where the anchor contains $n-1$ elements from the outfit. Whereas x_p is the missing element, x_n is an element that does not match the outfit but is in the same category as x_p.

Multimodal Encoder

Regarding compatibility modeling, this work uses transfer learning to train a model for extracting pairs of visual and textual features. A model takes normalized feature encoders and combines text and image representations into a joint embedding space. The model learns a complementary embedding space, and the implemented baseline



(a) Baseline detection model result

(b) Fine-tuned model result

Figure 2: Examples of the detection model performance before and after training on fashion sets.

model comprises three main modules of sub-networks: an image encoder, a text encoder, and a multimodal encoder. The multimodal encoder concatenates text and image representations to a multimodal feature space, followed by a projection block to fuse both modalities into a shared embedding space.

Optimization

Training the network aims to bring the embeddings of x_a and x_p closer together while moving x_n away. The model optimizes the Margin Ranking Loss to maximize the embedding distance of the incompatible products so that the distance of the incompatible products is greater than the distance of the compatible products. The distance between the non-complementary products (anchor and negative samples) is greater than the distance of complementary products (anchor and positive samples) by a margin. This relationship is shown by the equation in Figure 4.

Experimental details

Model experiments were conducted with various feature encoders. Text descriptions were tokenized using a tokenizer from the Hugging Face library. Pretrained text models on the English language BERT model [13] and CLIP (Contrastive Language-Image Pre-Training) [14] were selected. Furthermore, this article tested an image encoder architecture for the visual encoder, including a pre-trained ResNet-50 [15], Vision Transformer (ViT) [16], EfficientNetV2 [17], and MLP-Mixer [18] for image classification models. All baseline versions were trained for 30 epochs with a batch size of 64, and the L2 regularization technique was used. As for the model's architecture, appropriate embedding dimensions were used according to the parameters of the pre-trained image encoder for the text encoder and the multimodal encoder. Lastly, the pairwise distance between input vectors was computed. Table 2 shows the results for modeling feature encoder pairs for metrics for the validation set.

Table 2: Outfit compatibility prediction AUC and FITB accuracy on Polyvore dataset.

Model	AUC	FITB
Bert + MLP-Mixer	0.65	0.51
Bert + ViT	0.65	0.50
Bert + EfficientNetV2	0.65	0.53
Bert + ResNet-50	0.67	0.56
CLIP + MLP-Mixer	0.70	0.57
CLIP + ViT	0.71	0.59
CLIP + EfficientNetV2	0.71	0.60
CLIP + ResNet-50	0.72	0.61

Evaluation

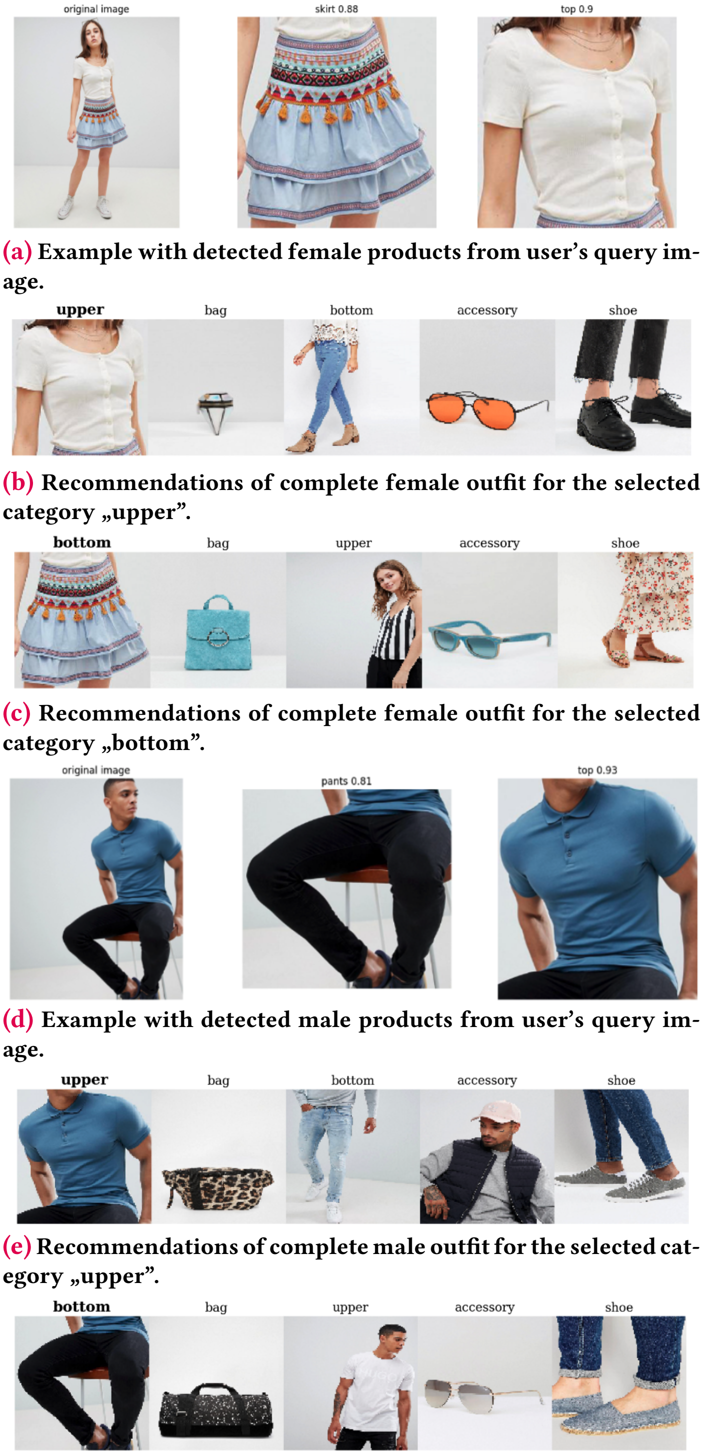
The approach was evaluated to perform two tasks in the fashion recommendation experiment, following metrics from similar studies [4, 5, 7, 12]. The fashion compatibility score was computed as overall compatibility, where a candidate outfit is scored as all its parts are compatible. Performance is evaluated using the average under a receiver operating characteristic curve (AUC). On the other hand, the Fill-in-the-blank (FITB) metric aims to select the most consistent item with the rest of the outfit as a question. In the evaluation outfit, one item is missing. The aim is to choose as the answer the product with the highest score out of four options to choose from.

Observations

According to this study and a review of similar works, results are improved when using transfer learning with the latest vision and language models. However, the analysis of the obtained results indicates potential biases of the dataset used in the study. This issue may be related to overrepresenting some characteristics, such as women's gender.

4.3. Retrieval results in the products catalog

This task is focused on creating a model that can recommend complementary clothing items that are compatible with a particular piece of clothing. A dataset was collected from an e-commerce store to achieve this goal, which was then used to assess and visualize the approach. The final step involved combining all the modules into a pipeline that could retrieve the query image and provide appropriate recommendations. The detection module was responsible for identifying the products in the input image and generating embeddings for these objects using a trained compatibility model encoder. Once the relevant fashion items were extracted from the user's picture, similar products from the catalog database were fetched. To accomplish this, similarly, the products from the database were encoded with a compatibility model encoder. The retrieval module was designed to embed product images in a compatibility-aware embedding space and to retrieve compatible products using the k-nearest neighbors index. The k-nearest elements from the embedding catalog products belonging to different categories were selected for embedding of the input photo. Figure 3 displays some search results obtained using this methodology.



(a) Example with detected female products from user's query image.

(b) Recommendations of complete female outfit for the selected category „upper".

(c) Recommendations of complete female outfit for the selected category „bottom".

(d) Example with detected male products from user's query image.

(e) Recommendations of complete male outfit for the selected category „upper".

(f) Recommendations of complete male outfit for the selected category „bottom".

Figure 3: Visualizations of the compatible product retrieval results using the proposed approach with user query image and e-commerce products catalog.

5. Conclusion and further works

In conclusion, the proposed system has demonstrated promising results in the automation recommendation of compatible products from multiple categories based on user interests. By combining rich datasets and cutting-edge machine learning techniques, we were able to extract detailed and standardized product information and provide effective recommendations. This study experimented with multiple pre-trained feature extraction models. We have successfully trained the object detection model for the fashion article detection and localization task. This significantly improves the ability to accurately identify and locate fashion items in images, which is especially important for e-commerce platforms.

Looking ahead, the following steps are planned for further refinements to the study. Firstly, a newer and faster version of the detection model will be used, hyperparameters will be optimized, and the results will be verified by humans. Additionally, training the compatibility model on an extended dataset with catalog products and human evaluation of results to improve the accuracy of recommendations. By adding specific fine-grained classes instead of general categories, we can lead to improved recommendations for more niche categories. Additionally, providing more context to the dataset and incorporating real-world user data can help resolve potential biases and improve the accuracy of recommendations. Overall, the goal of the proposed approach is to implement such a system and expose it to users to try it out and receive feedback, which will further enhance the effectiveness and usefulness of the presented recommendation system. We believe that the system can provide a more satisfying shopping experience for users.

Acknowledgements

This project was created in collaboration with the Shopai sp. z o.o company.

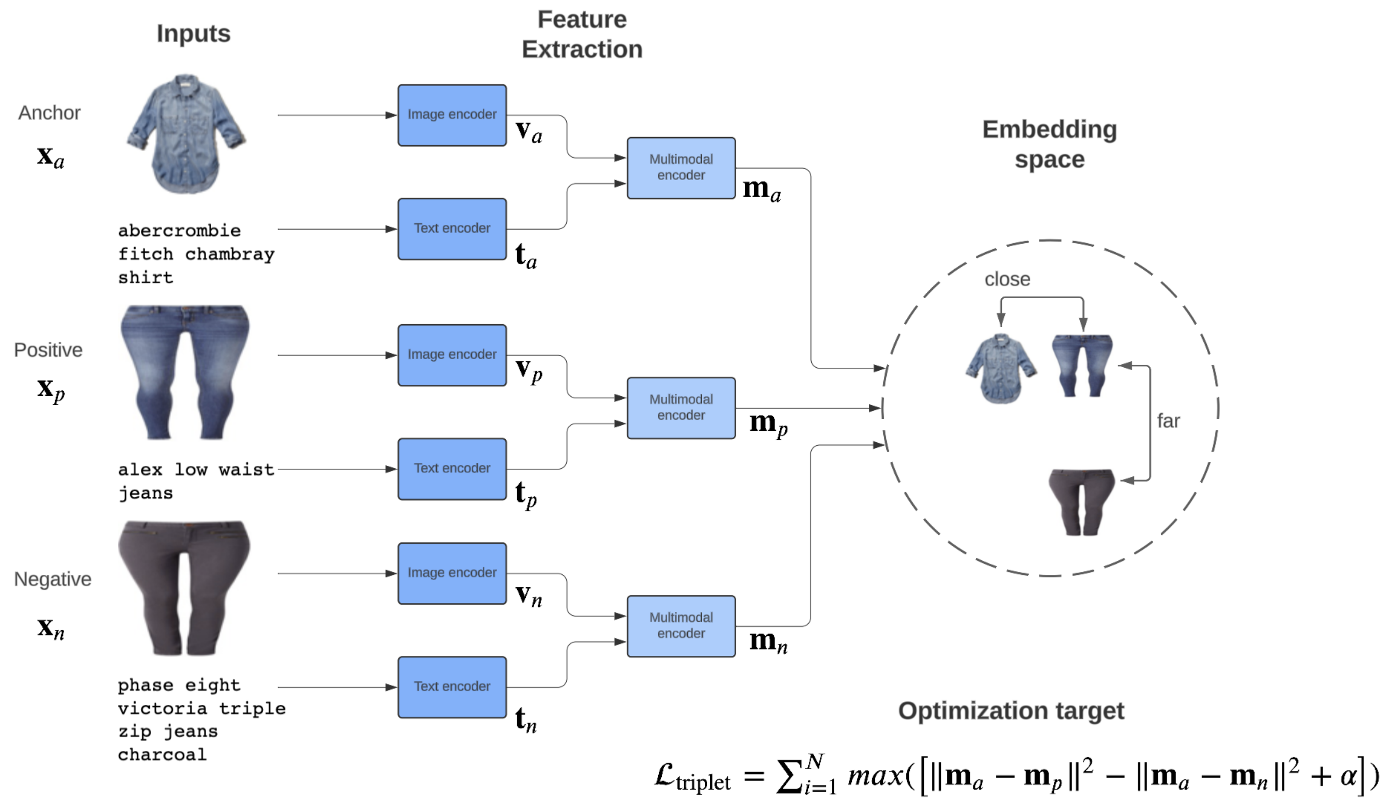


Figure 4: Compatibility model. A model takes normalized feature encoders and combines text t and image v representations into a joint embedding space. The model learns a complementary embedding space to optimize the Margin Ranking Loss to maximize the embedding distance of the incompatible products. The loss is computed across all triplets in the batch. m_a, m_p, m_n are embeddings of the anchor, positive, and negative examples respectively. The *max* function ensures that the loss is non-negative. A α hyperparameter defines a minimum margin by which the anchor-positive distance must be closer than the anchor-negative distance.

References

[1] Y.-H. Chang and Y.-Y. Zhang, "Deep learning for clothing style recognition using yolov5," *Micromachines*, vol. 13, no. 10, p. 1678, 2022.

[2] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5337–5345, 2019.

[3] "Yolov5." `https://github.com/ultralytics/yolov5`. Accessed: 2023-07-01.

[4] M. I. Vasileva, B. A. Plummer, K. Dusad, S. Rajpal, R. Kumar, and D. Forsyth, "Learning type-aware embeddings for fashion compatibility," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 390–405, 2018.

[5] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1078–1086, 2017.

[6] A. Veit, S. Belongie, and T. Karaletsos, "Conditional similarity networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 830–838, 2017.

[7] Z. Cui, Z. Li, S. Wu, X.-Y. Zhang, and L. Wang, "Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks," in *The world wide web conference*, pp. 307–317, 2019.

[8] Y. Li, L. Cao, J. Zhu, and J. Luo, "Mining fashion outfit composition using an end-to-end deep learning approach on set data," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1946–1955, 2017.

[9] A. Ravi, S. Repakula, U. K. Dutta, and M. Parmar, "Buy me that look: An approach for recommending similar fashion products," in *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 97–103, IEEE, 2021.

[10] E. Li, E. Kim, A. Zhai, J. Beal, and K. Gu, "Bootstrapping complete the look at pinterest," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3299–3307, 2020.

[11] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "Modanet: A large-scale street fashion dataset with polygon annotations," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1670–1678, 2018.

[12] X. Wang, B. Wu, and Y. Zhong, "Outfit compatibility prediction and diagnosis with multi-layered comparison network," in *Proceedings of the 27th ACM international conference on multimedia*, pp. 329–337, 2019.

[13] "Bert base model." `https://huggingface.co/bert-base-uncased`. Accessed: 2023-07-01.

[14] "Clip (contrastive language-image pre-training)." `https://github.com/openai/CLIP`. Accessed: 2023-07-01.

[15] "Resnet-50." `https://huggingface.co/microsoft/resnet-50`. Accessed: 2023-07-01.

[16] "Vision transformer." `https://huggingface.co/google/vit-base-patch32-224-in21k`. Accessed: 2023-07-01.

[17] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*, pp. 10096–10106, PMLR, 2021.

[18] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in neural information processing systems*, vol. 34, pp. 24261–24272, 2021.