Outfit Compatibility with Body-Shape

M1- Intrenational Track in Electrical Engineering

Carine Farhat
Department of Electrical Engineering,
Université d'Evry,
Université Paris-Saclay
Evry, France
carine_farhat@outlook.com

***Abstract*— The fashion industry has made remarkable progress in integrating artificial intelligence (AI) and computer vision technologies to enhance outfit recommendations. AI models now possess the capability to analyze images of individuals and extract valuable information regarding their body shape and proportions. These models, trained on vast and diverse datasets encompassing various body types and fashion trends, offer accurate and personalized suggestions. In this paper, we present an overview of strategies employed in developing AI-based outfit recommendations, and introduce our own methodology focused on the compatibility task. Our methodology involves predicting body shape using Openpose pose estimation and the SMPLify-X model, separating outfit garments through object detection or segmentation using the YOLOv8 model, and finally computing the compatibility at the garments level and at the body-shape level. We evaluate the compatibility of garments using graph encoder-decoder neural networks. Furthermore, we implement clothes parsing and pose estimation tasks, discussing their results, and present a user interface to provide a glimpse of the final product. Notably, our experimentation reveals promising results for YOLOv8 in garment detection, while indicating the potential for further improvements in pose estimation. The rest of the experiments are to be conducted in the future. This paper primarily focuses on the body-shape compatibility task, while acknowledging that the fashion compatibility domain is constantly evolving and possesses ample room for improvement. Future considerations may include incorporating hair, skin, and age-related factors to enhance the accuracy and relevance of recommendations and incorporating active learning for personalization.**

Keywords—Fashion, Compatibility, Body-Shape, AI, computer vision.

I. INTRODUCTION

The fashion industry has witnessed significant advancements in its use of artificial intelligence (AI) and computer vision technologies. In the ever-evolving world of fashion and personal style, finding the perfect outfit that compliments an individual's body shape can be a challenging task (Fig.1). However, AI models can analyze an image of a person and extract valuable information about their body shape and proportions. These models are trained on vast amounts of data, incorporating diverse body types and fashion trends, enabling them to provide accurate and personalized recommendations. In the rising world of 3D shops and online shopping, AI advising applications can find themselves useful for both the businesses and the customers, serving as a marketing tool for the businesses and as guarantee for successful online purchase decisions for the customers. In this paper, we overview some of the strategies used to develop AI-based outfit recommendations and then develop our own methodology to tackle this problem. Our developed methodology allows us to compute a compatibility score and to achieve the fill-in the blank task, i.e., choose a garment from a given set to complete the outfit. However, for the sake of simplicity, we focus on the compatibility task in this paper. In our workflow, starting from one image, we predict the body shape using the Openpose pose estimation and the SMPLify – X (Skinned Multi-Person Linear – Expressive) model. We also separate the different outfit garments via object detection or segmentation using the YOLOv8 model. In the next step, we evaluate the compatibility of the garments between themselves and their compatibility with the body via graph encoder-decoder neural networks. We implement the clothes parsing task and the pose estimation task and discuss their results and create a user interface to have a basic idea of how the final product would look like.

Fig. 1. Example of outfit compatibility with different body-shapes [3]

II. RELATED WORKS

Fashion applications have experienced a significant rise in popularity alongside the increasing prevalence of 3D shops and online shopping. They leverage technology to provide users with immersive and convenient fashion experiences. These applications include outfit planning apps such as GetWardrobe [6], and Pureple [13], managing user's closet based on the season, the style, the occasion, etc. However, they do not study the compatibility of the items but rather take it as input and do not consider the body shape.

On the theoretical level, although many works have tackled the compatibility of the outfit garments, very little prior works have tackled the compatibility of the outfit with the body shape using computer vision. Many non-academic works have tackled the problem using simple linear regression of non-visual data. However, these methods are not user friendly as they require the user to input a lot of body dimensions such as shoulder width, hips' width etc. Among the few literature works that use vision for body shape compatibility, we cite Sattar et al.'s "Fashion is Taking Shape" [16] which originally introduced the idea, used a multi-photo approach with the SMPL (Skinned Multi-Person Linear) model to predict the body shape, and inferred the outfit/body-shape compatibility by splitting the outfits into two groups Ga corresponding to "average size" and Gp corresponding to "plus size". Although this paper made a first step towards solving the problem, it did not completely leverage the learning possibilities and limited the compatibility predictions to the 2 classes Ga and Gp. Furthermore, its multi-photo approach is not really suited for our goal of creating a user-friendly workflow. Hsiao et al. then introduced the idea of Visual Body-aware Embedding (ViBE) [9] which captures close affinities between the clothes and the body type. However, they did not address the compatibility of the outfit garments. A more complete approach by which we were highly inspired in the development of our methodology is the one introduced by Chong et al. [3]. They use Open Pose Detection and SMPLify-X to predict the body shape and YOLOv3 to detect the outfit garments. They then develop the Body-Shape Aware Object-Level Model (BOCTP) which aim is to fill in the blanks of a missing item in the image without requiring the user to specify which garment it is. It also tries to match the item with the overall image scenery. It is composed of five components, including image-level distance, region-level distance, object-level distance, body-shape compatibility, and category matching. In our case, using such a big model is unnecessary as we are only interested in predicting the compatibility of the outfit garments and their compatibility with the body shape. We find that for our practical applications, predicting the missing item in the outfit is not useful.

We notice that there isn't a single proper approach to tackle the compatibility problem. Although the compatibility with the body shape task is relatively new and does not have a lot of literature work, other tasks in the project have been extensively explored. We now look at each one of them:

A. *Detecting the outfit garments*

A common practice in clothes parsing is to use semantic segmentation for the purpose of separating and labeling the clothes. State-of-the-art methods using Graph LSTMs [12] or Extended U-nets [20] have achieved really good results for the clothes parsing task; however, not without imperfections in the results. Given the promising results of the newly released YOLOv8 which is capable of doing both object detection and instance segmentation, we decided that testing YOLOv8 for the clothes' parsing task is worth trying.

B. *Estimating 3D body shape*

3D generative body models capture anthropometric constraints of the population and facilitate the task of predicting 3D bodies from 2D images. Older works such as Chen et al.'s [2] have leveraged these generative models to estimate 3D shape from single images using shading cues, silhouettes, and appearance. However, newer state-of-the-art methods almost all use the SMPL model and try to fit it to the 2D image's respective joints [3][9][16]. Similarly to Chong et al. [3], we use the SMPLify-X model which is an improved version of the SMPL model with additional expressive features and better modeling of the hands.

C. *Compatibility of the outfit garments*

Several approaches have been adopted to evaluate the compatibility score of outfit garments. They all focus on achieving the 2 tasks of computing the compatibility score and filling in the blanks, i.e., determining what garment would fit best in place of the fashion item. Each one of these approaches has some intuition to it and specific hypotheses. For example, Han et al. [8], treat the outfit as a sequence (whereby we pick the top first then the pants then the shoes, etc..) and employ a Bi-directional LSTM to conditionally predict each item based on the previous one. Vasileva et al. [19] develop a type-aware embedding model whereby embeddings of each the outfit garments are made while specifically respecting the outfit type. This model was useful in making sure that 2 garments of the same type would not be matched in the fill-in the blanks task. Both previously discussed models use large datasets such as the *Polyvore* and *Modanet* datasets, assume that the outfits in the dataset are compatible and use them for their inference problem. Balim et al. [1] on the other hand build a fashion outfit compatibility model that uses image captioning techniques to describe what are the outfit positive and negative aspects for training. It therefore takes both compatible and incompatible outfits for training. The method that we found would most properly fit our project is the Context-aware compatibility model developed by Curcull et al. [5] which treats the outfit as a graph whereby the nodes are the garments, and the edges represent whether the items are compatible or not. During training, it uses both real edges (from the dataset) representing the compatible items and non-existent edges (randomly generated) representing the non-compatible outfits to infer the compatibility using embeddings and an encoder-decoder structure.

D. *Personalized Compatibility*

One of the challenges of using AI to predict outfit compatibility is the fact that fashion might be considered by some as subjective and varying between different populations. For this reason, some works have implemented active learning methods to refine the compatibility predictions based on the user's previous feedback. Song et al. [17] present a personalized compatibility modeling scheme GP-BPR, comprising of two essential components: *general compatibility modeling* and *personal preference modeling*, which characterize the item-item and user-item interactions, respectively. Li et al. approach the problem by developing a new framework, Hierarchical Fashion Graph Network (HFGN) [11], to model relationships among users, items, and outfits simultaneously. Although we did not consider the personalization in this first version of the project, it is a good perspective to keep in mind.

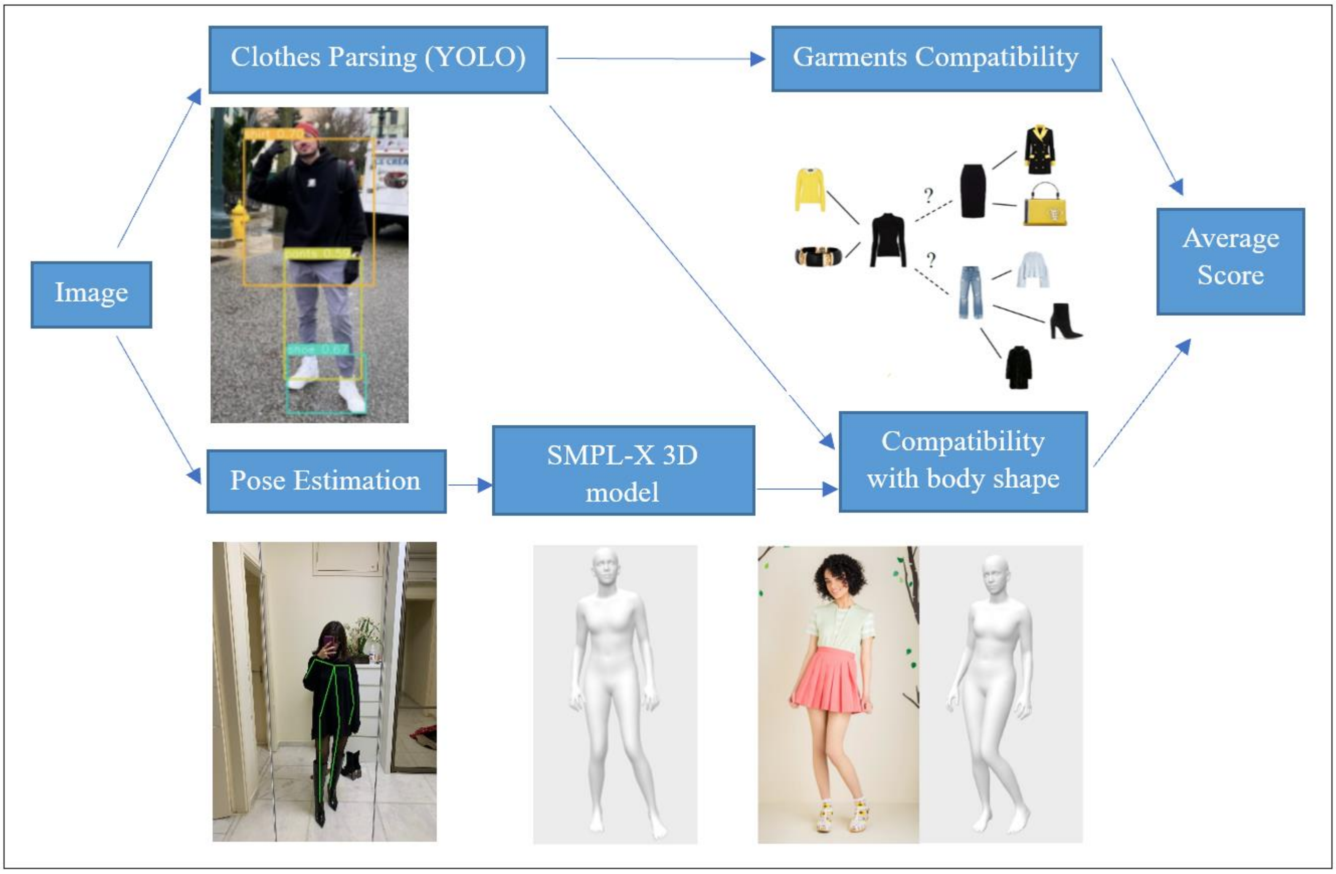

Fig.2 Schematic of the Developed Methodology

E. *Virtual Try-On*

Although the virtual try-on task is also not considered in the current scope, it could be considered as a great way to allow the users to visualize by themselves how the outfit would look on their body and give a more complete compatibility result. Some fashion-design software such as Clo3D [4] have already implemented virtual try-on experiences. However, they have targeted designers to help them visualize their creations and have not been optimized for the compatibility task targeting the regular shopper. Virtual try-on boils down to a clothing re-targeting problem. It entails adapting clothing to different body shapes and poses. Pons-Moll et al. [15] propose a method that tackles this challenge by simultaneously capturing body shape and 3D clothing geometry. This enables the re-targeting of clothing items to new bodies and poses. Han et al. [7] adopt a different strategy by using simplified body proxies instead of inferring 3D information. Their method involves directly re-targeting clothing to individuals in the image space using an encoder-decoder generator. Another notable approach by Lassner et al. [10] uses a Variational Auto-Encoder (VAE) and the SMPL body model to generate images of people wearing various clothing items.

III. METHODOLOGY

In this section, we discuss the different modules that allow us to develop the fashion advising task. The developed workflow is really a basic one, focusing only on the compatibility of the outfit with the body shape. Many developments can be made to improve it. It is composed of 4 modules: Clothes parsing, Fashion Compatibility, 3D Body Modeling, Compatibility with Body Shape. The clothes are first recognized and segmented using YOLOv8, their compatibility is then evaluated using a context-aware approach. From the 2D image, we develop the 3D model of the body of the person using the SMPL-X algorithm and then evaluate the compatibility of the outfit to the body shape. The outfit compatibility and body compatibility scores are then weighed together to give a final outfit score. A schematic of the workflow is shown in Fig.2.

A. *Clothes Parsing*

The first step of our workflow consists of detecting and labeling the clothes from the input picture, i.e. the outfit that the person is already wearing. To do this, we use YOLOv8 algorithm. YOLO, which stands for “You Only Look Once” is a state-of-the-art real-time object detection algorithm that performs detection and localization of objects in images and video frames. The YOLO algorithm is innovative in its way of predicting the bounding boxes and their respective class with a single pass through the neural network, i.e., it unifies the separate components of object detection into a single neural network. YOLO employs a pre-trained Convolutional Neural Network and functions as follows: The input image is divided into a grid of cells. The size of the grid depends on the network architecture with varying grid sizes in each layer. Each cell in the grid has a set of predefined anchor boxes.

For a 2-class problem for example, one anchor box would look like in Eq. 1:

$$anchor\ box = [Pc;\ Bx;\ By;\ Bw;\ Bh;\ C1;\ C2] \tag{1}$$

Where Pc is the probability of finding an object in the grid cell, Bx, By, are the coordinates of the box' center, Bw and By are the box dimensions and C1, and C2 define the class i.e. 1 for the corresponding class, 0 for the other. For such a problem, we would typically use 2 anchor boxes, one for each class.

During the training and inference process, the network weights are to be refined to correctly predict the anchor boxes, defining both the bounding boxes dimensions and center and the class of the box. To eliminate overlapping bounding box predictions, YOLO employs non-maximum suppression which eliminates the boxes with low confidence scores and high overlap measured by the Intersection over Union (IoU) score [18]. We choose YOLOv8 as this newly developed version surpasses the earlier ones in terms of speed and accuracy. It is also more flexible as it also includes instance segmentation, which can be highly useful later in our project.

B. *Garments Compatibility*

A second step of the fashion compatibility task is to evaluate the compatibility of the garments. To do this, one of the best approaches found in literature and that we would adopt in our project is the Context-Aware Visual Compatibility prediction method developed by Guillem Cucurull et al. [5]. Instead of addressing outfit compatibility as a pairwise distance between 2 garments, the compatibility problem is addressed using a graph neural network that learns to generate product embeddings conditioned on their context. Embeddings refer to representations of nodes or graphs that capture important information about their structural and relational characteristics. These embeddings are learned through the encoding process of a graph auto-encoder which is the basis framework of Curcull et al.'s model. The way it works is that the graph auto-encoder first takes an incomplete graph; the encoder consisting of a Graph Convolutional Neural Network (CNN), produces an embedding for each node; then the decoder learns a metric to predict the compatibility score between pairs of embeddings (Fig. 3).

For further details, we note:

- A graph is characterized by N nodes and edges. Each node is represented with a vector of F features $\overrightarrow{x_i} \in \mathbb{R}^F$ and constitutes a row of the features matrix $X \in \mathbb{R}^{N*F}$. Edges are represented inside the adjacency matrix $A \in \mathbb{R}^{N*N}$ whereby $A_{ij} = 1$ if there exists an edge between nodes i and j and 0 otherwise. $\overrightarrow{x_i}$ can be computed with a CNN feature extractor.
- The encoder transforms the initial features X into a new representation $H = f_{enc}(X, A) \in \mathbb{R}^{N*F}$ function of the feature and adjacency matrices. In our case, the initial features $\overrightarrow{x_i}$ contain information about the garment itself (e.g. shape, color, size). In order to achieve the context-aware learning, the encoder gives a new representation capturing not only the garment properties but also combining them with information about the other items the garment in question is compatible with. The function f_{enc} is found using a Deep Graph Convolutional Network with multiple hidden layers.
- Based on the new representations of the nodes $\overrightarrow{h_i}$ and $\overrightarrow{h_j}$, the decoder logistically computes the probability that 2 nodes are connected based on notion of compatibility:

$$p = \sigma(|\overrightarrow{h_i} - \overrightarrow{h_j}|\overrightarrow{w^T} + b) \tag{2}$$

Where w^T and b are learnable parameters. The sigmoid function σ maps the scalar value obtained to a probability value in the range [0,1].
- For the training, we remove a subset of the edges inside the original adjacency matrix to generate an incomplete adjacency matrix $\hat{A}$. This subset is noted ε^+ and is made of positive edges (existing edges between nodes). We randomly sample a subset of negative edges ε^- (non-existing edges between pairs of nodes). We use both positive and negative subsets for the training process. The model predicts the status of the edges found in $\varepsilon_{train} = [\varepsilon^+, \varepsilon^-]$ and is optimized by minimizing the cross-entropy loss between the predicted edges and their true status found in ε_{train}.
- The outfit compatibility task can finally be solved as a probability problem whereby the model computes the probability of an edge between each pair of nodes making up a total of $\frac{N(N-1)}{2}$ edges. The compatibility score (Eq. 3) of an outfit becomes the average of all pairwise edge probabilities:

$$\frac{2}{N(N-1)} \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} p_{i,j} \tag{3}$$

A schematic of the process is found in Fig. 3 below.

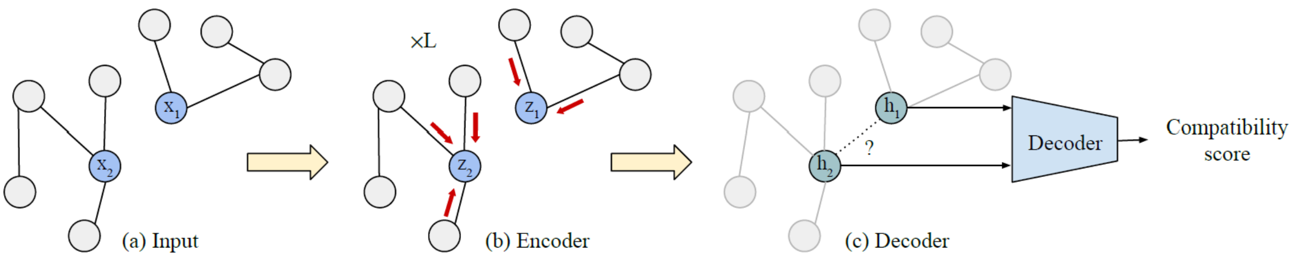

Fig.3 Context-aware compatibility, Graph Encoder, Decoder Network

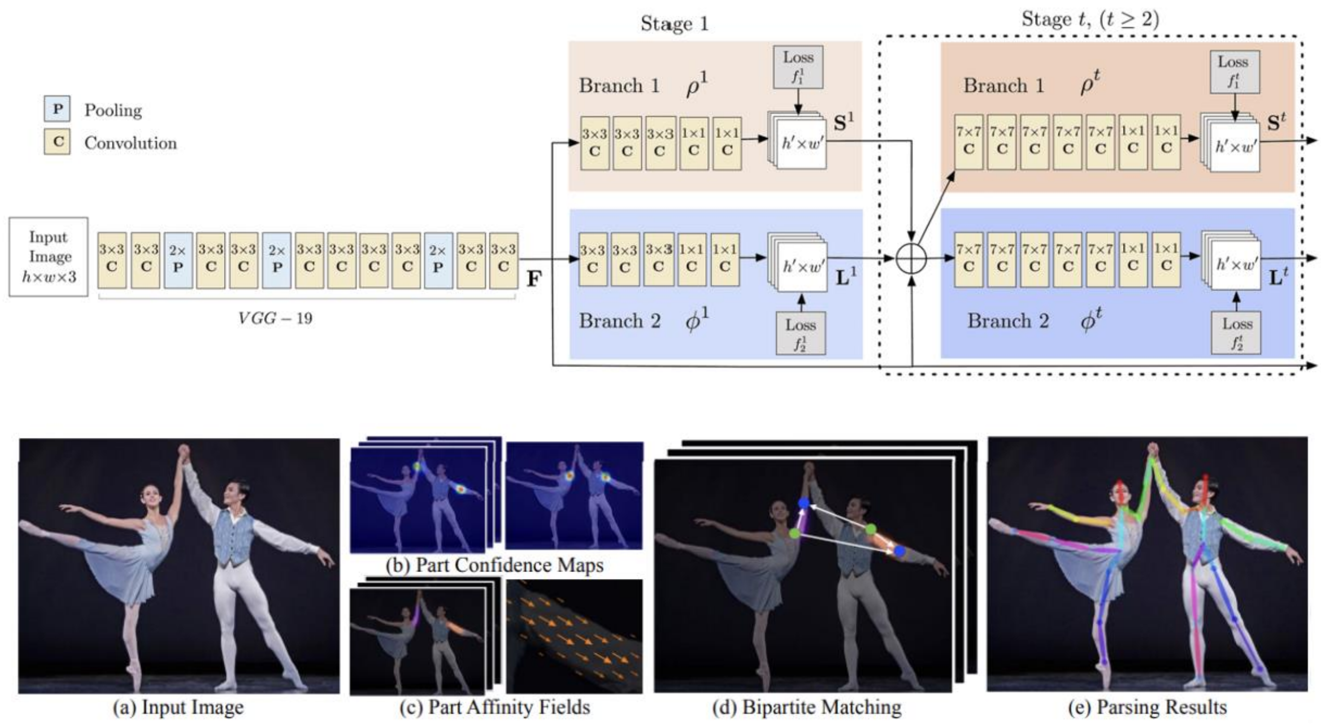

Fig.4 Pose Estimation Network and Workflow [21]

C. 3D Model of the Human Body

To evaluate the compatibility of the outfit to the body shape, we adopt Chong et al.'s method and first create a 3D model of the human body [3]. We do this by using the SMPLify-X method which estimates 3D shape and pose of the human body from a single RGB image. SMPLify-X is an extension of the original SMPL (Skinned Multi-Person Linear) model that improves its robustness and adds to it the possibility of modeling the hands and facial expressions. Although these do not really matter for the fashion task, they are more visually friendly to the user. SMPLify-X first takes as input the OpenPose 2D pose estimation as a json file.

1) *Pose Estimation*

OpenPose pose estimation is a model which allows the localization of human body key points such as joints and body parts. It can allow the estimation of multiple bodies in a single picture, which is not necessarily important for our current application, but can be useful later for its development. To estimate the pose, the model first employs a pretrained Convolutional Neural Network (VGG-19) which is responsible for the extraction of the features with high importance (Stage 0). Next, in stage 1, the Neural Network is branched into two different branches. The first branch is responsible for finding the different keypoints and creates confidence maps (heatmaps) to measure the likelihood of a particular keypoint being present at different spatial locations. Multiple estimations can be made for one single keypoint; in order to choose the correct estimation, non-maximum suppression is performed. After non-maximum suppression, multiple estimations can still exist for a single keypoint; the different estimations of pairs of keypoints are then connected to form a bipartite graph. The second branch of the neural network is responsible for finding the degree of association between 2 different keypoints and generating the part affinity fields, i.e. understanding which of the different parts can be combined to form a pair. Both the affinity fields and the bipartite graph are then used to know which one of the connections between a pair of keypoints represents best the positioning and linking of the body parts. This process is first initiated in stage 1 and then fine-tuned in later stages whereby the part affinity fields from the previous layers are used to refine the prediction of the confidence maps [21].

2) *3D Human Body Model*

SMPLify-X formulates the problem of estimating 3D pose and shape as an optimization task. It employs the SMPL human body parametric model which is a collection of 3D points that are connected to form a surface. The model focuses on optimizing an objective function combining:

- The reprojection error term consisting of the distance between the 2D joints and their respective projected 3D points using an estimated camera.
- A penalizing self-penetration term
- A pose prior term using the Gaussian Mixture Model (GMM)
- Regularization terms

The objective function is minimized using gradient descent. The optimization procedure continues until convergence, or a predefined stopping criterion is met. The final estimated 3D body model provides the reconstructed shape and pose of the person in the input image [14].

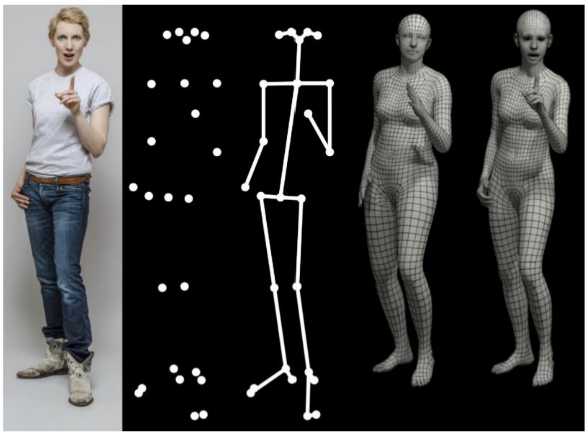

Fig.5 SMPL-X model [14]

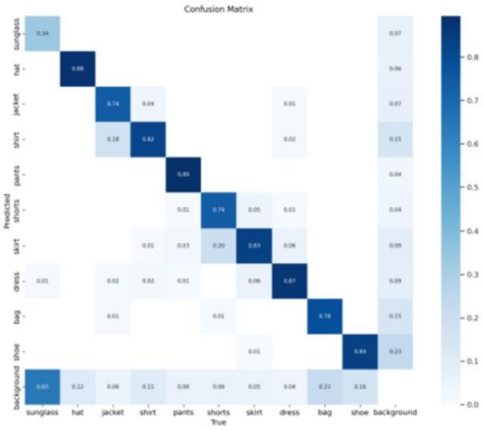

Fig.6 Confusion matrix – YOLOv8m results

D. *Outfit Compatibility with Body Shape*

Second, also to compute the compatibility of the outfit with the body shape while inspired Chong et al.'s method, we project the outfit into the body space using an encoder similarly to what was done previously for the garment compatibilities whereby the newly defined feature vectors would also contain information about the body shape designating their compatibility [3]. Therefore, we define a function h that adopts a linear projection for dimension reduction, such as the new low-level feature vector b_i is equal to:

$$b_i = h(\varphi, \sigma_i) = P\sigma_i \tag{4}$$

Where P is the embedding matrix.

Matrix factorization is then used to predict the compatibility between an item and the human body. The model would be trained with real-life pictures whereby the outfits would be compatible with the body shape. The final outfit-body compatibility score would be calculated as an average the compatibility of each garment with the body shape. To train this model we would use a large dataset containing real images of people of different body shapes wearing compatible outfits.

IV. Experiments and Results

In this section, we discuss the experiments we did in the process of building the project.

A. *Clothes Parsing*

We tested Yolov8 for the object detection task and refine it using the Kaggle "Colorful Fashion Dataset" which is composed of 2682 annotated fashion images consisting mainly of women. Annotations contain both information about the bounding boxes and the 10 garment classes: sunglasses, hat, jacket, shirt, pants, shorts, skirt, dress, bag, and shoes. Although larger YOLO models would have given us better results, we used YOLOv8m for the sake of testing while minimizing the training time.
The object detection results were really promising as the model was able to reach around 80% accuracy with its medium size. Class accuracy was 80% and higher for all classes except for the sunglasses which were most of the time classified as background (Fig.5), (Fig.6). Next, we will test the YOLOv8 segmentation and evaluate its results.

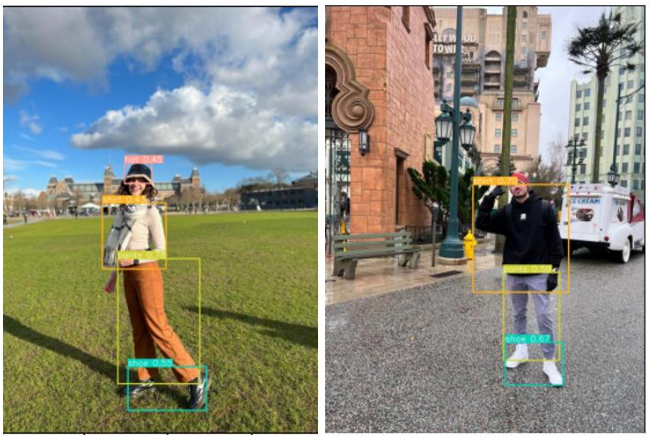

Fig.7 Bounding Boxes – YOLOv8m results.

B. *Pose Estimation*

We test the pose estimation algorithm and notice that the joints are accurately predicted when the person is facing the camera (Fig. 8). However, not very accurate estimations were made when the person was rotated. As this might cause issues for the SMPL-X model, we might need to make some modifications to the code or find an alternative pose estimation method in the future.

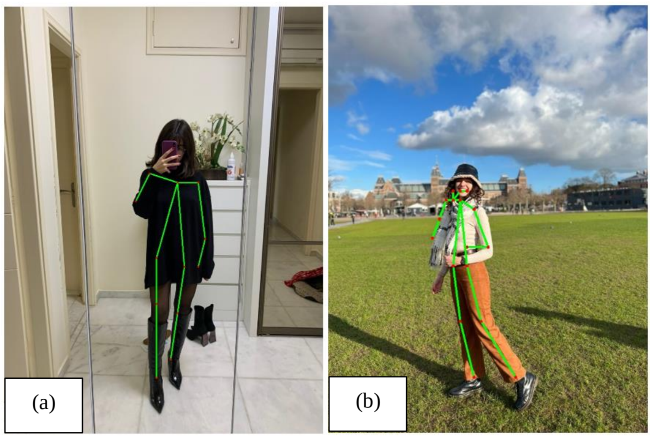

Fig.8 Pose Estimation – (a) Good results in the frontal case – (b) Bad results in the rotated case.

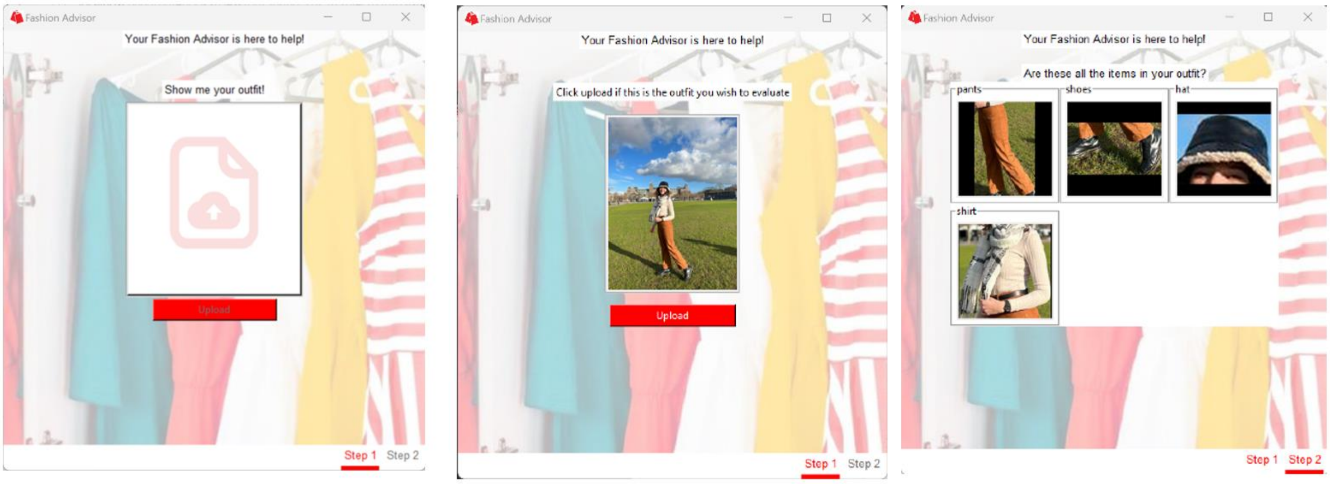

Fig.9 Steps 1 and 2 of User Interface

C. User Interface

We started implementing a user interface (Fig.9) to imagine how the workflow would look like when completely implemented. Our user interface would be sequential in nature, consisting of the following steps: Step 1 – The user uploads a picture and confirms their choice, Step 2 – Clothes parsing YOLO results are displayed and require user confirmation, Step 3 – Results is displayed. As we have already tested the garment detection, we were able to implement the user interface up until Step 2. We implement using the Tkinter Python library.

To create a sequential user interface, we create a function for each one of the steps which would contain all the widgets and functions required for the step. We manage the transitions between the steps using a steps bar by enabling and disabling the buttons in the steps bar. We create a delete page function that would remove all widgets from the window once a step is completed to make room to the new widgets.

Step 1 contains instruction text and a select button with its respective select function allowing to access the filing system and choose a local picture. By clicking the upload button after selecting the picture, the picture is converted to PIL to be inputted into the YOLO model and the window transitions to Step 2.

Step 2 contains a grid system to be able to display the detected garments. It contains cropping and resizing functions to be able to crop the detected bounding boxes and fit them into the grid while respecting their aspect ratio.

V. Limtations and Future Works

We note that the developed framework is really a basis for outfit compatibility with body shape. It currently holds many limitations whereby fashion subjectivity is not taken into consideration. Physical properties such as skin-color, hair, age, were not taken into account. In fact, the fashion compatibility task can be approached in so many ways which always leaves room for improvement. Therefore, after implementing our basis methodology, it would be a good idea to start taking the user's opinion and other physical properties into consideration.

VI. Conclusion

The newly developed AI solutions are promising for the fashion field. With the ever-evolving fashion industry, AI solutions could offer great new perspectives. We now have a idea of how to predict fashion compatibility with body shape. After implementing our 4 modules workflow, garment detection / segmentation, garment compatibility, 3D body shape estimation and compatibility with body shape, we would try to take personalization and other physical properties such as hair, skin, age etc. into consideration and even try adding a virtual try-on experience. We note that through the implementation, the workflow may change depending on the experimentation results. The experiments we already did and mentioned in this paper already told us that YOLOv8 is a good candidate for the clothes parsing task. However, the pose estimation does not give accurate positioning of the 2D joints. It therefore might need to be improved if it causes problems with the SMPL-X model. In fact, the complexity of the fashion compatibility task is that there is always something more to take into consideration. The nice thing about it is that it always has room for improvement.

References

[1] Balim, C., & Özkan, K. (2023). Diagnosing fashion outfit compatibility with Deep Learning Techniques. *Expert Systems with Applications*, *215*, 119305. https://doi.org/10.1016/j.eswa.2022.119305

[2] Chen, Y., Kim, T.-K., & Cipolla, R. (2010). Inferring 3D shapes and deformations from single views. *Computer Vision – ECCV 2010*, 300–313. https://doi.org/10.1007/978-3-642-15558-1_22

[3] Chong, X., & Leung, H. (2023). Body shape-aware object-level outfit completion for full-body portrait images. *Applied Sciences*, *13*(5), 3214. https://doi.org/10.3390/app13053214

[4] *CLO: 3D fashion design software*. CLO Official Site. (n.d.). https://www.clo3d.com/en/

[5] Cucurull, G., Taslakian, P., & Vazquez, D. (2019). Context-aware visual compatibility prediction. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.01290

[6] Get the most out of Wardrobe. Getwardrobe. (n.d.-a). https://getwardrobe.com/

[7] Han, X., Wu, Z., Wu, Z., Yu, R., & Davis, L. S. (2018). Viton: An image-based Virtual Try-On Network. *2018 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2018.00787

[8] Han, X., Wu, Z., Jiang, Y.-G., & Davis, L. S. (2017). Learning fashion compatibility with bidirectional lstms. *Proceedings of the 25th ACM International Conference on Multimedia*. https://doi.org/10.1145/3123266.3123394

[9] Hsiao, W.-L., & Grauman, K. (2020). Vibe: Dressing for diverse body shapes. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr42600.2020.01107

[10] Lassner, C., Pons-Moll, G., & Gehler, P. V. (2017). A generative model of people in Clothing. *2017 IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2017.98

[11] Li, X., Wang, X., He, X., Chen, L., Xiao, J., & Chua, T.-S. (2020). Hierarchical Fashion Graph Network for personalized outfit recommendation. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3397271.3401080

[12] Liang, X., Shen, X., Feng, J., Lin, L., & Yan, S. (2016). Semantic Object Parsing with graph LSTM. *Computer Vision – ECCV 2016*, 125–143. https://doi.org/10.1007/978-3-319-46448-0_8

[13] Outfit organizer App. Pureple. (n.d.). https://pureple.com/

[14] Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A., Tzionas, D., & Black, M. J. (2019). Expressive body capture: 3D hands, face, and body from a single image. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.01123

[15] Pons-Moll, G., Pujades, S., Hu, S., & Black, M. J. (2017). Clothcap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics*, *36*(4), 1–15. https://doi.org/10.1145/3072959.3073711

[16] Sattar, H., Pons-Moll, G., & Fritz, M. (2019). Fashion is taking shape: Understanding clothing preference based on body shape from online sources. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. https://doi.org/10.1109/wacv.2019.00108

[17] Song, X., Han, X., Li, Y., Chen, J., Xu, X.-S., & Nie, L. (2019). GP-BPR: Personalized Compatibility Modeling for Clothing Matching. *Proceedings of the 27th ACM International Conference on Multimedia*. https://doi.org/10.1145/3343031.3350956

[18] Terven, J. R., & Cordova-Esparaza, D. M. (2023). A COMPREHENSIVE REVIEW OF YOLO: FROM YOLOV1 TO YOLOV8 AND BEYOND.

[19] Vasileva, M. I., Plummer, B. A., Dusad, K., Rajpal, S., Kumar, R., & Forsyth, D. (2018). Learning type-aware embeddings for fashion compatibility. *Computer Vision – ECCV 2018*, 405–421. https://doi.org/10.1007/978-3-030-01270-0_24

[20] Vozáriková, G., Staňa, R., & Semanišin, G. (2021). Clothing parsing using extended U-Net. Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. https://doi.org/10.5220/0010177700150024

[21] Zhao, Y., Yan, H., & Wang, Z. (2019). The advisable technology of key-point detection and expression recognition for an intelligent class system. *Journal of Physics: Conference Series*, *1187*(5), 052011. https://doi.org/10.1088/1742-6596/1187/5/052011