VICTOR: VISUAL INCOMPATIBILITY DETECTION WITH TRANSFORMERS AND FASHION-SPECIFIC CONTRASTIVE PRE-TRAINING

A PREPRINT

Stefanos-Iordanis Papadopoulos
CERTH-ITI
stefpapad@iti.gr

Christos Koutlis
CERTH-ITI
ckoutlis@iti.gr

Symeon Papadopoulos
CERTH-ITI
papadop@iti.gr

Ioannis Kompatsiaris
CERTH-ITI
ikom@iti.gr

ABSTRACT

In order to consider fashion outfits as aesthetically pleasing, the garments that constitute them need to be compatible in terms of visual aspects, such as style, category and color. With the advent and omnipresence of computer vision deep learning models, increased interest has also emerged for the task of visual compatibility detection with the aim to develop quality fashion outfit recommendation systems. Previous works have defined visual compatibility as a binary classification task with items in a garment being considered as fully compatible or fully incompatible. However, this is not applicable to Outfit Maker applications where users create their own outfits and need to know which specific items may be incompatible with the rest of the outfit. To address this, we propose the Visual InCompatibility TransfORmer (VICTOR) that is optimized for two tasks: 1) overall compatibility as regression and 2) the detection of mismatching items. Unlike previous works that either rely on feature extraction from ImageNet-pretrained models or by end-to-end fine tuning, we utilize fashion-specific contrastive language-image pre-training for fine tuning computer vision neural networks on fashion imagery. Moreover, we build upon the Polyvore outfit benchmark to generate partially mismatching outfits, creating a new dataset termed Polyvore-MISFITs, that is used to train VICTOR. A series of ablation and comparative analyses show that the proposed architecture can compete and even surpass the current state-of-the-art on Polyvore datasets while reducing the instance-wise floating operations by 88%, striking a balance between high performance and efficiency.

Keywords Recommendation System · Outfit Matching · Visual Compatibility · Computer Vision · Deep Learning

1 Introduction

Fashion products do not exist in a vacuum. When customers consider buying a new garment they may contemplate its subjective appeal, price, quality or trendiness but also think of ways to match it with other pieces and how compatible it is with other items in their wardrobe. To help customers in their endeavours, contemporary e-commerce applications usually provide outfit recommendations and suggestions of how to "complete the look" based on an item of interest. Outfit compatibility is a rather challenging task: not only is it highly subjective but it also involves numerous variables such as the style, color, fit, patterns, proportions, textures of numerous garments and how these aspects interrelate. To this end, researchers have recently utilized computer vision neural networks, that learn to produce informative representations from fashion images, along with pairwise-based [Tan et al., 2019, Vasileva et al., 2018], graph-based [Cucurull et al., 2019, Cui et al., 2019] or attention-based neural networks [Zhan and Lin, 2021, Zhan et al., 2021, Chen et al., 2019] that learn to predict the compatibility of outfits.

However, previous studies define outfit compatibility prediction as a binary (OC_b) classification task. An outfit is either fully compatible or fully incompatible. This is a reasonable assumption for e-commerce applications that recommend fully compatible outfits to their customers. It is not as applicable to *Outfit Maker* applications[1], where users combine

[1]Examples of outfit maker applications include: ShopLook, Smart Closet, Stylebook, Pureple and Combyne

garments to create their own outfits. Instead, it would be more useful to offer an overall compatibility score and detect specific mismatching garments in order to inform users which items are not compatible with the rest of the outfit. This would give a sense of how aesthetically pleasing an outfit is and help users identify garments with clashing colors or patterns, select more suitable alternatives and generally fine-tune their outfits.

In this study we define outfit compatibility as a regression (OC_r) problem and also address the task of mismatching item detection (MID) in fashion outfits. We use the Polyvore outfit dataset [Vasileva et al., 2018] which consists of fully compatible and incompatible outfits to generate partially mismatching outfits (MISFITs). We propose the **V**isual **I**n**C**ompatibility **T**ransf**OR**mer, or VICTOR, a multi-tasking, Transformer-based architecture that is trained to predict the overall OC_r score and detect mismatching garments in an outfit. Previous works on OC_b either rely on feature extraction from computer vision models pre-trained on ImageNet[Chen et al., 2019, Lorbert et al., 2021] or end-to-end fine-tuning [Han et al., 2017, Vasileva et al., 2018, Lin et al., 2020, Tan et al., 2019, Sarkar et al., 2022]. When using visual features from ImageNet-pretrained models, VICTOR outperforms other methods by 4.87% in terms of AUC on the Polyvore dataset. On the other hand, E2E fine-tuning tends to significantly outperform feature extraction but is notably more resource intensive. Instead, we utilize *fashion-specific contrastive language image pre-training* (FLIP) to fine-tune computer vision models for fashion imagery and then use the extracted visual features for OC_r and MID. Empirical results show that VICTOR with FLIP are capable of competing and even surpassing, the current state-of-the-art on Polyvore datasets for OC_b while reducing instance-wise floating point operations (FLOPs) by an impressive 88%.

The main contributions of our work are:

- We define two new sub-tasks for visual compatibility, namely: outfit compatibility prediction as regression (OC_r) and mismatching item detection (MID) and examine them in the domain of Fashion with the proposed multi-tasking Transformer-based neural network that is optimized for both tasks.
- We propose a methodology for generating partially mismatching outfits and create a new dataset called Polyvore-MISFITs. We provide the code[2] that generates the Polyvore-MISFITs dataset in order to encourage further research in the field.
- We utilize fashion-specific contrastive language image pre-training (FLIP) for fine-tuning computer vision neural networks on fashion imagery. We experiment with four computer vision backbones and perform an extensive ablation and comparative analysis that shows VICTOR with FLIP to be capable of competing and even surpassing the current state-of-the-art on Polyvore datasets while reducing instance-wise floating point operations by 88% and total study-wise operations by up to 98%.

2 Related Work

In recent years, researchers have shown increased interest in applying deep learning and computer vision neural networks [Cheng et al., 2021] in order to address numerous tasks relevant for the Fashion domain including category and attribute classification [Liu et al., 2016, Papadopoulos et al., 2022a], trend forecasting [Al-Halah et al., 2017, Mall et al., 2019], popularity prediction [Skenderi et al., 2021, Papadopoulos et al., 2022b], fashion recommendations systems [Hwangbo et al., 2018, Stefani et al., 2019] and among them, the task of outfit recommendations. In order to recommend complete outfits it is first necessary to understand which garments go well together and can create compatible and cohesive outfits.

The first studies to address the task, considered outfit compatibility as a series of pairwise comparisons between all comprising garments [Tan et al., 2019, Vasileva et al., 2018]. Pairwise-based approaches have utilized Siamese [Veit et al., 2015] and triplet loss networks with either type-aware embeddings [Vasileva et al., 2018] or similarity-aware embeddings [Tan et al., 2019]. Other works, instead of aggregating garment-level relations attempted to capture global outfit-level representations with the use of bidirectional LSTMs [Han et al., 2017] or graph neural networks [Cucurull et al., 2019, Cui et al., 2019]. In practice, outfits are not ordered sequences; the order of the garments should not affect the model's predictions. Thus, recurrent neural networks are not the most suitable architecture for the task. On the other hand, graph-based approaches tend to require large "neighborhoods" of compatible garment-nodes as input in order to reach optimal performance which is problematic for new items that lack neighbor information and may straggle from the cold start problem [Lin et al., 2020].

In order to address the aforementioned challenges, more recent works have employed attention-based methods [Zhan and Lin, 2021, Zhan et al., 2021, Chen et al., 2019]. Attention mechanisms have been used in pairwise-based approaches [Lin et al., 2020, Taraviya et al., 2021] but the Transformer architecture has been successfully used for personalised

[2] The GitHub repository will be provided upon acceptance of the paper.

outfit recommendations [Chen et al., 2019] and complementary item retrieval [Sarkar et al., 2022]. With the use of multi-head attention, the Transformer is suitable for learning relations between multiple items, in this case the compatibility between all garments in an outfit. Additionally, by removing the positional encoding [Vaswani et al., 2017, Dosovitskiy et al., 2020] it can capture unordered relations between all garments.

However, all aforementioned studies have defined outfit compatibility as a binary classification problem. An outfit is treated as either fully compatible or fully incompatible. To the best of our knowledge, this is the first study to tackle the task of mismatching item detection (MID) and treat compatibility prediction as a regression (OC_r) instead of a binary task (OC_b).

Previous works have relied on visual, textual information and fashion categories for creating representations of garments in outfits. Transfer learning is generally being used for extracting visual information from the garment's images, either with feature extraction (FX) from ImageNet pretrained models [Chen et al., 2019, Lorbert et al., 2021] or by end-to-end fine-tuning (E2E) for OC_b [Han et al., 2017, Vasileva et al., 2018, Lin et al., 2020, Tan et al., 2019, Sarkar et al., 2022]. E2E tends to outperform FX-ImageNet since the visual features are trained to specialize on the target domain and task. Nevertheless, E2E is a highly resource intensive process since the gradients of a - usually large - network backbone need to be updated on top of the outfit matching neural network. In this study, we attempt to find the middle ground between the efficiency of FX and the high accuracy of E2E by utilizing contrastive language-image pre-training - inspired by [Radford et al., 2021] - with a focus on fashion imagery.

3 Methodology

3.1 Problem Formulation

In this study, we address the task of mismatching item detection (MID) in fashion outfits. Moreover, we define visual outfit compatibility prediction as a regression task (OC_r) - allowing for partially mismatching outfits - in contrast to previous studies that define it as a binary classification task (OC_b). Let a fashion outfit $\mathcal{O} = \{g_1, g_2, \ldots, g_n\}$ consist of n garments g_i. Our architecture after processing the outfit images, $\mathcal{I} = \{I(g_1), I(g_2), \ldots, I(g_n)\}$, produces $n+1$ outputs, one for the OC_r task denoted $Y_{OC_r} \in (0,1) \subset \mathbb{R}$ and n for the MID task denoted $Y_{MID} \in (0,1)^n \subset \mathbb{R}^n$, which are optimized to comply with the corresponding target variables, T_{OC_r} and T_{MID}. First, $T_{OC_r} \in (0,1) \subset \mathbb{R}$ denotes the compatibility of the garments, where 0 means that all garments are incompatible, 1 that all are compatible and in-between values denote partial compatibility. Second, a list of binary values $T_{MID} = [x_{g_1}, x_{g_2}, \ldots, x_{g_n}]$, with $x_{g_i} \in \{0,1\}$ and $i = 1, \ldots, n$, where 1 denotes the mismatching garments in outfit $\mathcal{O}$. OC_r is defined as a regression task and MID as a multi-label classification task.

3.2 Generating Mismatching Outfits

Existing outfit datasets, e.g. Polyvore [Vasileva et al., 2018], provide annotations for fully compatible or fully incompatible outfits. In this study we attempt to address partial incompatibility and the detection of specific mismatching items within an outfit. To this end, we generate partially mismatching outfits (MISFITs) with the following method. For every matching outfit $\mathcal{O}$, with $n > 2$ we generate m number of MISFITs by randomly selecting (i) the number of garments $1 \leq r \leq n-2$ that will be replaced and (ii) their positions $\mathcal{P}$. The garments in positions $\mathcal{P}$ are then replaced with randomly selected items of the same category, thus generating hard negative samples. Hard negative sampling forces the model to recognize and focus on fine-grained characteristics of the garments and their interrelations. In contrast, sampling garments from different categories would be easier for the model to recognize but it would learn less useful relations [Vasileva et al., 2018], for example that an outfit can not consist of two dresses or two pairs of shoes. For $\mathcal{O}$ with $n = 3$ we only allow $r = 1$ because having outfits with only 1 compatible item is invalid. The target compatibility score is calculated as $T_{OC_r} = 1 - r/n$ and the mismatching items target is defined as a list T_{MID} of binary values with 1 in $\mathcal{P}$ positions denoting the incompatible garments and 0 in other positions denoting the compatible garments. The fully compatible outfits retain $T_{OC_r} = 1$ and $T_{MID} = [0, 0, \ldots, 0]$, while the fully incompatible ones $T_{OC_r} = 0$ and $T_{MID} = [1, 1, \ldots, 1]$, respectively.

3.3 VICTOR

The proposed pipeline of the Visual InCompatibility TransfORmer (VICTOR) is illustrated in Fig. 1. First, the images $\mathcal{I} = \{I(g_1), I(g_2), \ldots, I(g_n)\}$ of all garments in an outfit $\mathcal{O}$ are passed through a visual encoder $\mathrm{E_V}(\cdot)$ that produces the corresponding vector representations $\mathbf{v}_{g_i} \in \mathbb{R}^{e \times 1}$, where e is the encoder's embedding dimension. Then, following Dosovitskiy et al. [2020] that makes use of a classification token (CLS) we similarly consider a regression token

(REG)[3] and pass $\{\mathbf{v}_{g_i}\}_{i=1}^{n} \cup \{< \text{REG} >\}$ through a Transformer decoder[4] D($\cdot$). Outfits are not sequential objects thus we do not make use of positional encodings [Vaswani et al., 2017, Dosovitskiy et al., 2020] so as to capture the unordered relations between garments. The Transformer decoder D($\cdot$) consists of L layers that have h attention heads of embedding dimension d. Finally, the OC_r score Y_{OC_r} and the MID scores list Y_{MID} of the outfit are calculated as:

$$\begin{aligned}
\mathbf{v}_{g_i} &= \mathrm{E_V}(I(g_i)) \\
\mathbf{d}_{g_i} &= \mathrm{D}(\mathbf{v}_{g_i}) \\
\mathbf{d}_{<\mathrm{REG}>} &= \mathrm{D}(< \mathrm{REG} >) \\
Y_{OC_r} &= \mathbf{W}_1 \cdot \mathrm{GELU}(\mathbf{W}_0 \cdot \mathrm{LN}(\mathbf{d}_{<\mathrm{REG}>})) \\
Y_{MID}[i] &= \mathbf{W}_i \cdot \mathrm{GELU}(\mathrm{LN}(\mathbf{d}_{g_i}))
\end{aligned}$$

where $\mathbf{d}_{g_i} \in \mathbb{R}^{d\times 1}$ and $\mathbf{d}_{<\mathrm{REG}>} \in \mathbb{R}^{d\times 1}$ are the Transformer's outputs, $\mathbf{W}_0 \in \mathbb{R}^{\frac{e}{2}\times d}$, $\mathbf{W}_1 \in \mathbb{R}^{1\times \frac{e}{2}}$ and $\mathbf{W}_i \in \mathbb{R}^{1\times d}$ are sigmoid activated dense projection layers (learnable bias terms are considered but omitted here for clarity), LN stands for Layer Normalization and GELU is the activation function. Zero padding is also considered for D($\cdot$) input in outfits with less than 19 items being the largest outfit size in Polyvore.

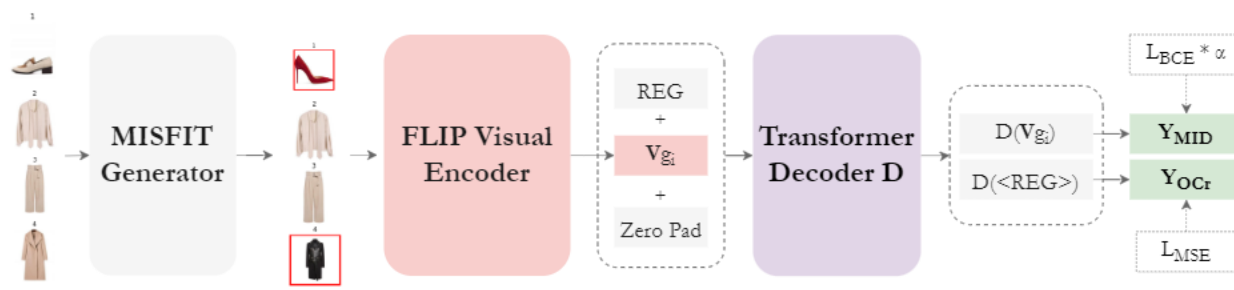


Figure 1: Workflow of the VICTOR architecture.

D($\cdot$) utilizes multi-head attention, thus each token contains information about a garment's interrelations with all other garments. In our case this translates to an item being mismatching with the rest of the items in the outfit. Sarkar et al. [2022] proposed the use of the CLS token for predicting the overall compatibility of the outfit. However, after experimentation, we found this to be sub-optimal for the MID task and our architectural approach to perform consistently better. VICTOR is optimized based on two different loss functions. Y_{OC_r} - being a regression task - is optimized based on the mean squared error loss function (L_{MSE}), while Y_{MID} - being a multi-label classification task - is optimized based on the binary cross entropy (L_{BCE}) loss, ignoring the zero padded items. However, the two loss functions do not necessarily have balanced values. We therefore introduce α, a hyper-parameter for weighted combination of the two loss functions as a standard multi-objective optimization practice. The final loss for VICTOR is calculated as $L = L_{MSE} + L_{BCE} \cdot \alpha$.

3.4 Fashion-specific language image pre-training (FLIP)

Analysing the visual compatibility of fashion items requires the use of computer vision neural networks for producing informative representations of said items. Unlike previous works that have utilized feature extraction from ImageNet-pretrained models or end-to-end fine-tuning, we propose the use of contrastive language-image pre-training for fashion imagery (FLIP). FLIP's workflow is illustrated in Fig. 2 and is following the training procedure proposed by Radford et al. [2021]. FLIP consists of one visual $\mathrm{E_V}(\cdot)$ and one textual $\mathrm{E_T}(\cdot)$ encoder. Image-text pairs $(I(g_i), T(g_i))$ are passed through their respective encoders and the resulting embeddings are projected onto the same embedding space with the use of two fully connected layers of the same size one for each encoder, as shown below:

$$\begin{aligned}
\mathrm{F_V(i)} &= \mathbf{W_V} \cdot \mathrm{E_V}(I(g_i)) \\
\mathrm{F_T(i)} &= \mathbf{W_T} \cdot \mathrm{E_T}(T(g_i))
\end{aligned}$$

The dot product between image and text projection embeddings are calculated and the loss function is defined as the mean cross entropy between the predicted and the target image-text pairs, the latter being reflected by the main diagonal. Our rationale for utilizing FLIP is that it balances performance and efficiency. Training computer vision

[3] A trainable vector that learns a global representation incorporating information about the relations of all garments in an outfit.

[4] D($\cdot$) is actually structured as the encoder part of the original Transformer architecture but we use it to decode the image embeddings in our model thus we call it a decoder herein.

models end-to-end for outfit compatibility can yield a high performance but is a rather resource intensive process. On the other hand, ImageNet-pretrained models do not specialise on fashion imagery and can only produce a general visual representation. In contrast, the visual encoder of FLIP will learn to produce fashion-specific features. FLIP does not require annotated fashion datasets, which are expensive and time consuming to produce, instead it relies on image-texts pairs of existing fashion products which are easier to attain. Moreover, we may train a single FLIP model, extract the visual features from fashion imagery and re-use them for numerous experiments on outfit compatibility, such as hyper-parameter tuning and ablation analyses, without requiring e2e fine-tuning. Thus, significantly reducing floating point operations (FLOPs) and by extension computational costs and training time.

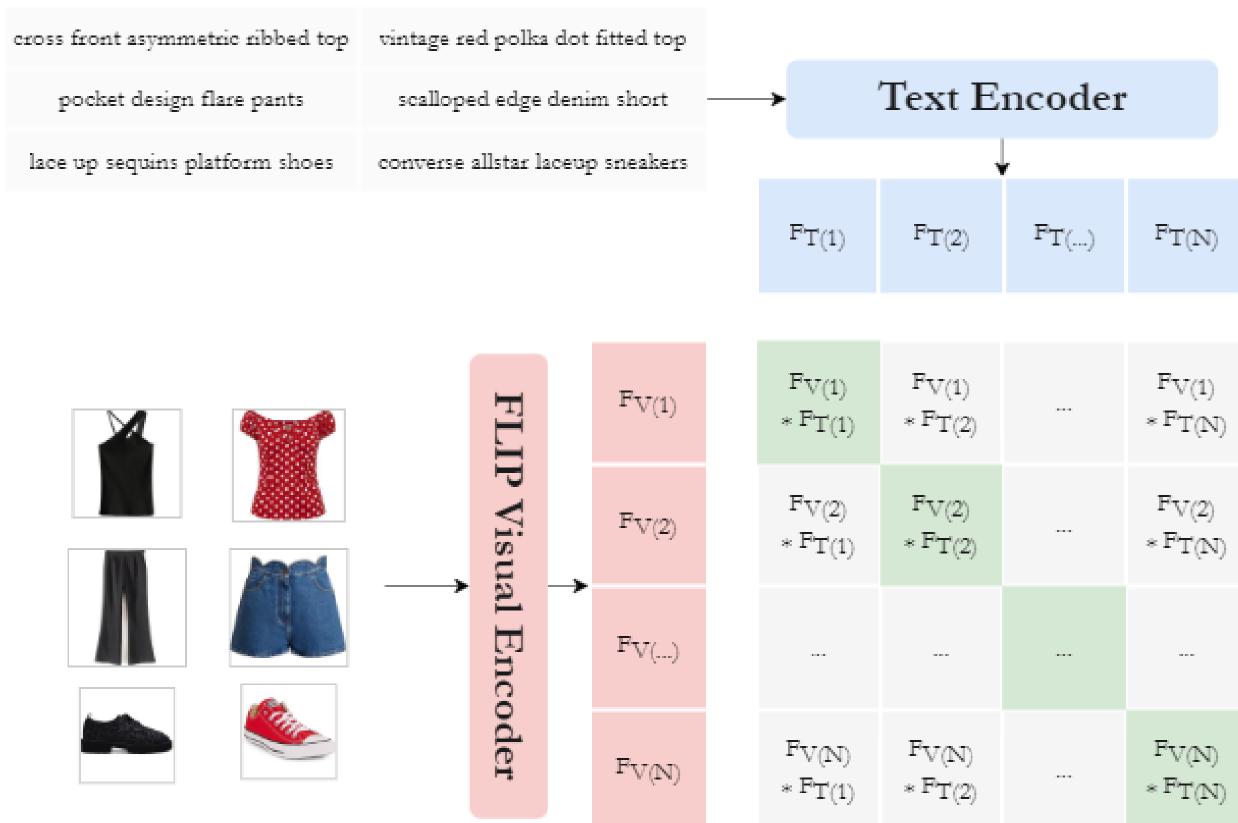


Figure 2: Workflow of fashion language-image pre-training (FLIP). FLIP consists of a visual and a textual encoder that are trained contrastively to predict the correct image-text pair which are placed in the main diagonal. Images and texts are selected with in-batch sampling.

4 Experimental Setup

4.1 Polyvore Dataset

The Polyvore dataset is a widely used benchmark dataset for outfit recommendation that was collected by Vasileva et al. [2018]. The dataset provides 68,306 matching outfits comprising 251,008 unique garments. Each garment comes with multi-modal information including an image, product name, description and associated fashion categories consisting of 14 *types* and 142 categories including *bottoms*: “skirt”, “long skirt”, *tops*: “sweater”, “turtleneck sweater”, *shoes*: “boots”, “flat sandals” but also *hats*, *jewelry* and other *accessories*. For every matching outfit the authors have generated an equal amount of fully incompatible outfits by randomly replacing each garments with items of the same category. The dataset comes in two versions that have fixed training, validation and testing splits. The first version of Polyvore consists of 106,612, 10,000, 20,000 outfits for training, validation and testing respectively. There are no overlapping outfits between the different splits but garments can overlap between the splits. The second version, Polyvore-Disjoint, consists of 33,990, 6,000, 30,290 outfits for training, validation and testing but there are no overlapping garments between the splits. Each outfit has at least 2, a maximum of 19 and a median value of 5 garments. As the target variable T_{OC_b}, fully compatible outfits have a score of 1 while fully incompatible have 0.

Outfits with more than 10 garments make up less than 0.5% of all outfits and could therefore be considered outliers. Moreover, we found that outfits with more than 10 garments often have more than one garment of the same category e.g. two pants or two jackets, which is infeasible. However, we do not filter anything out so as to ensure comparability with previous works.

4.2 Polyvore-MISFITs Dataset

We apply the MISFIT generation process described in section 3.2 on the Polyvore dataset for $m = 2$ and $m = 4$. $m = 2$ creates a balanced dataset between the initial and the generated outfits, with 133,944 MISFITs out of the 270,556 in total which are distributed into 104,498, 9,794, 19,652 for training, validation and testing while $m = 4$ generates 267,888 MISFITs with a total of 404,500 outfits which are split into 315,608, 29,588, 59,304.

Figure 3: Examples of generated MISFITs from fully compatible outfits. Red frames denote the mismatching items.

Fig. 3 presents two indicative examples of generated MISFITs. On top there are two women's outfits of different styles which are annotated as matching. On the left, a classic monochromatic look with a loose fit and on the right a casual look with black pieces and blue jeans. The MISFIT generation process has randomly replaced certain garments of the original outfit with items of the same category. For example, the beige pair of wide-fitting pants is replaced with leopard-print leggings (item 3, row 1) and the leather jacket (right outfit) is replaced with a colorful Aztec-pattern jacket (item 6, row 1). These, like most replaced garments, are not matching the aesthetic and style of the initial outfit. Thus, they are correctly categorised as mismatching items. However, the generation process is not perfect. For example, the pair of beige loafers is replaced with a beige pair of heels (item 1, row 4, left outfit) which some would not consider it to be mismatching with the rest of the outfit. The replacement process is random, thus, replacement items may match by chance. However, since each category contains thousands of items, we expect that random selections will more often than not lead to incompatible combinations. We provide the code[5] for generating the Polyvore-MISFITs for reproducibility and in order to encourage further research in the field.

4.3 Implementation Details

We perform an ablation and comparative analysis and in order to distinguish different versions of VICTOR, we denote the training task in square brackets. The proposed multi-tasking learning (MTL) model optimized both for OC_r and MID is referred to as VICTOR[MTL]. Furthermore, we define: (1) VICTOR[OC_b] trained only for binary outfit compatibility, optimized based on the binary cross entropy loss function, (2) VICTOR[OC_r] trained only for compatibility as regression optimized based on on the MSE loss function and (3) VICTOR[MID] trained only for mismatching item detection, optimized based on the multi-label binary cross entropy loss function. For all versions of VICTOR, we select $L = 8$ transformer layers of $d = 64$ dimensions, $h = 16$ attention heads, a dropout rate of 0.2 and a batch size of 512. We train VICTOR[MTL] four times with $\alpha \in [0.2, 0.5, 1, 2]$ and denote different values of α as VICTOR[MTL;α]. Wherever required, we also denote the version of Polyvore-MISFITs that was used to train VICTOR as VICTOR[MTL;α;m].

We use the image-text pairs from the Polyvore-Disjoint dataset for training FLIP since there is no overlap between training, validation and testing sets. For FLIP's visual encoder $\mathrm{E_V}$, we experiment with four models 1) ResNet18 [He et al., 2016], 2) EfficientNetV2-B3 [Tan and Le, 2021], 3) MLP-Mixer B/16 [Tolstikhin et al., 2021] and ViT B/32 [Dosovitskiy et al., 2020]. The aformentioned models are taken from the timm library[6] and are initially pre-trained on ImageNet. The input image sizes are 224 for all models expect EfficientNetV2-B3 which is 300. For FLIP's textual encoder $\mathrm{E_T}$, we use CLIP's Transformer text encoder and do not fine-tune it any further. We select a projection layer of 512 and a batch size of 32 for FLIP.

We train both FLIP and VICTOR for 20 epochs with the Adam optimizer and a learning rate scheduler with an initial learning rate of 1e-4 that reduces by a rate of 0.1 at 10 epochs.

Regarding the evaluation protocol, we follow all previous works that use the area under the roc curve (AUC) as the evaluation metrics for OC_b. For OC_r we report the mean absolute error (MAE) and for MID the binary accuracy and exact match. We use the training, validation and testing sets as provided by the Polyvore dataset in order to ensure fair comparability. We checkpoint the network's parameters with TOPSIS [Hwang and Yoon, 1981] based on the validation MAE, binary accuracy and exact match.

5 Results

5.1 FLIP and FLOPs

We fine-tune four computer vision neural networks for fashion imagery with the use of fashion language-image pre-training (FLIP). Their performance in terms of the cross entropy loss can be seen in Table 1. Lower values of cross entropy loss translates into fewer mistakes when matching the visual and textual projections of actual image-text pairs. However, lower cross entropy loss may not necessarily translate into better performance for VICTOR. Our rationale for employing FLIP was to fine-tune the models on fashion imagery while avoiding end-to-end (E2E) fine-tuning for outfit compatibility which can be considerably resource-intensive.

To measure the efficiency gains of FLIP, we calculate the number of floating point operations (FLOPs) using Facebook's *fvcore*[7]. Table 2 presents the FLOPs of each computer vision model for a single instance of training. We observe that employing FLIP and then utilizing the extracted visual features to train VICTOR reduces the number of FLOPs by an average of 88.14% compared to E2E training. Moreover, if we not only consider instance-wise FLOPs but also epoch-wise FLOPs there is an average decrease of up to 94.86%. This is due to FLIP being trained on the Polyvore-Disjoint dataset (86,624 training+validation instances) - but is then also used for Polyvore - compared to the Polyvore's 202,446 training+validation instances. Furthermore, we should also consider the re-usability of FLIP, meaning that a FLIP model can be trained once but its extracted features can then be re-used with no additional cost. In our study, we run 12 experiments per computer vision model, for the ablation study and the tuning of α. Compared to using standard E2E training within the same experimental setup, we have actually reduced the number of FLOPs by an impressive average of 98.81%. Utilizing FLIP proved to be significantly more efficient than conventional E2E training for outfit compatibility prediction.

[5]The GitHub repository will be provided upon acceptance of the paper.

[6]`https://github.com/rwightman/pytorch-image-models`

[7]`https://github.com/facebookresearch/fvcore`

Table 1: Performance of computer vision models fine-tuned with FLIP in terms of the cross entropy loss.

Model	Cross entropy loss (↓)
ViT B/32	1.27
ResNet18	1.23
MLP-Mixer B/16	1.21
EfficientNetV2-B3	**1.07**

Table 2: Floating-point operations (FLOPs) of VICTOR when trained with FLIP or end-to-end (E2E) fine-tuning with different computer vision models.

Model	**Model Parameters**	**FLIP**	**VICTOR**	**FLIP + VICTOR**	**VICTOR (E2E)**	**% ↓**
ResNet18	1.14E+07	5.36E+09	1.82E+08	5.54E+09	4.55E+10	87.8
EfficientNetV2-B3	1.30E+07	6.07E+09	1.55E+09	7.63E+09	6.02E+10	87.3
MLP-Mixer B/16	5.93E+07	7.31E+09	4.00E+08	7.71E+09	2.40E+11	96.8
ViT B/32	8.76E+07	1.56E+10	4.00E+08	1.60E+10	8.26E+10	80.62

5.2 Ablation Analysis

We perform an ablation analysis comparing the proposed multi-tasking VICTOR[MTL] with its two separate components, VICTOR[OC_r] and VICTOR[MID]. The results are shown in Table 3. For VICTOR we tune the α hyperparameter - the weight that combines the two loss functions - and report the best performing based on TOPSIS which takes into account *MAE*, exact match and binary accuracy. VICTOR[OC_r] is only trained for compatibility prediction as regression and can not detect specific mismatching items. Being specialised on OC_r, it yields an average *MAE* of 0.255 for $m = 2$ and 0.225 for $m = 4$. VICTOR[MTL] performs marginally better with 0.250 for $m = 2$ and 0.219 for $m = 4$. The overall lowest, hence better, MAE scores are reached by VICTOR[MTL] with MLP-Mixer B/16 and ViT B/32 for $m = 2$ and $m = 4$ respectively.

VICTOR[MID] is trained on predicting mismatching items in outfits and yields on average a *binary accuracy* of 71.46% for $m = 2$ and 70.19% for $m = 4$, closely followed by VICTOR[MTL] which has 70.14% and 68.8% respectively. In terms of the *exact match* evaluation metric, the strictest evaluation metric for the MID task, we observe that VICTOR[MTL] significantly outperform VICTOR[MID] with 40.65% compared to 37.80% for $m = 2$ while they

Table 3: Ablation analysis between VICTOR[OC_r], VICTOR[MID] and VICTOR[MTL] on Polyvore-MISFITs dataset with $m = 2$ and $m = 4$. For VICTOR[MTL] we report the best performing $\alpha = a|b$ based on TOPSIS with *a* for $m = 2$ and *b* for $m = 4$.

VICTOR	**FLIP Model**	**MAE (↓)**		**Exact Match (↑)**		**Accuracy (↑)**		OC_b **AUC (↑)**	
		m=2	m=4	m=2	m=4	m=2	m=4	m=2	m=4
VICTOR[OC_r]	ResNet18	0.254	0.221	-	-	-	-	0.90	0.90
	EfficientNetV2-B3	0.255	0.226	-	-	-	-	0.91	0.88
	MLP-Mixer B/16	0.255	0.229	-	-	-	-	0.89	0.86
	ViT B/32	0.254	0.225	-	-	-	-	**0.92**	**0.92**
	Average	0.254	0.225	-	-	-	-	0.91	0.89
VICTOR[MID]	ResNet18	-	-	38.30	26.29	68.64	69.44	0.89	0.90
	EfficientNetV2-B3	-	-	38.50	27.52	71.99	70.29	0.91	0.90
	MLP-Mixer B/16	-	-	36.70	27.03	72.42	70.44	0.90	0.90
	ViT B/32	-	-	37.69	27.85	**72.79**	70.59	0.91	0.90
	Average	-	-	37.80	27.17	71.46	70.19	0.90	0.90
VICTOR[MTL]	ResNet18 ($\alpha = 0.2\|0.2$)	0.257	0.224	40.70	26.02	69.35	65.75	0.90	0.90
	EfficientNetV2-B3 ($\alpha = 0.2\|1$)	0.248	0.216	39.70	26.98	70.24	69.79	0.91	0.91
	MLP-Mixer B/16 ($\alpha = 0.2\|0.2$)	**0.247**	0.222	**41.55**	26.15	70.57	68.98	**0.92**	0.91
	ViT B/32 ($\alpha = 0.2\|1$)	0.250	**0.214**	40.65	**27.91**	70.38	**70.68**	**0.92**	**0.92**
	Average	0.250	0.219	40.65	26.77	70.14	68.8	0.91	0.91

Table 4: Comparison with the state-of-the-art on binary outfit compatibility prediction (OC_b) in terms of AUC. For VICTOR, we report the best performing hyper-parameter combination.

Method	Input	Polyvore	Polyvore-D
BiLSTM + VSE [Han et al., 2017]	ResNet18 (E2E) + Text	0.65	0.62
GCN (k=1) [Li et al., 2019]	ResNet18 (E2E)	0.82	0.87
Li et al. [2019]	ResNet18 (E2E)	0.90	0.85
SiameseNet [Vasileva et al., 2018]	ResNet18 (E2E)	0.81	0.81
Type-aware [Vasileva et al., 2018]	ResNet18 (E2E) + Text	0.86	0.84
SCE-Net [Tan et al., 2019]	ResNet18 (E2E) + Text	0.91	-
CSA-Net [Lin et al., 2020]	ResNet18 (E2E)	0.91	0.87
OutfitTranformer [Sarkar et al., 2022]	ResNet18 (ImageNet)	0.82	-
OutfitTranformer [Sarkar et al., 2022]	ResNet18 (E2E)	0.91	-
VICTOR[OC_b]	ResNet18 (ImageNet)	0.86	0.78
	EfficientNetV2-B3 (ImageNet)	0.86	0.78
	MLP-Mixer B/16 (ImageNet)	0.84	0.73
	ViT B/32 (ImageNet)	0.86	0.80
	ResNet18 (FLIP)	0.90	0.85
	EfficientNetV2-B3 (FLIP)	0.91	0.86
	MLP-Mixer B/16 (FLIP)	0.91	0.86
	ViT B/32 (FLIP)	0.91	0.87
VICTOR[MTL]	ResNet18 (ImageNet)	0.86	0.77
	EfficientNetV2-B3 (ImageNet)	0.86	0.79
	MLP-Mixer B/16 (ImageNet)	0.84	0.71
	ViT B/32 (ImageNet)	0.86	0.78
	ResNet18 (FLIP)	0.90	0.85
	EfficientNetV2-B3 (FLIP)	0.91	0.87
	MLP-Mixer B/16 (FLIP)	**0.92**	0.87
	ViT B/32 (FLIP)	**0.92**	**0.88**

both perform similarly for $m = 2$, with 26.77% and 27.17% accordingly. The overall highest, hence better, exact match scores are reached by VICTOR[MTL] with MLP-Mixer B/16 and ViT B/32 for $m = 2$ and $m = 4$ respectively.

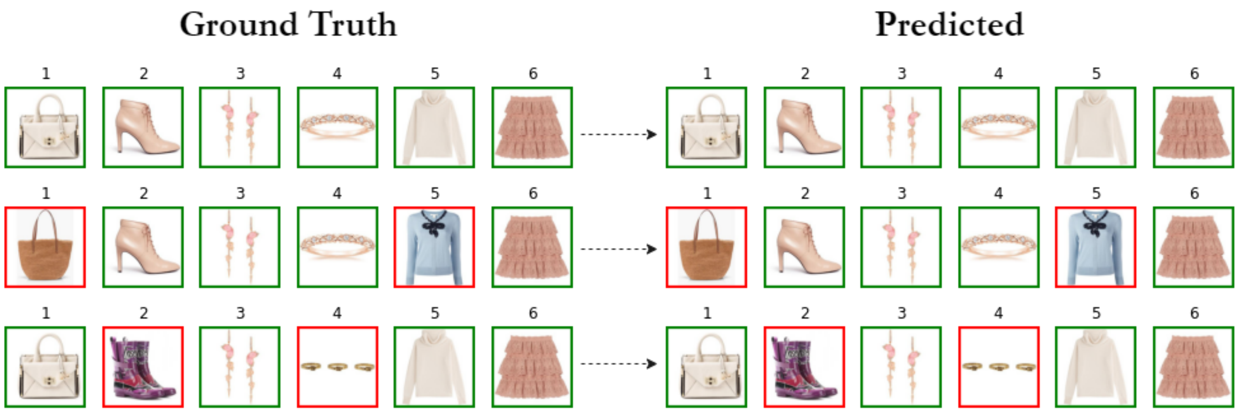
Finally, regarding binary outfit compatibility prediction (OC_b), which is evaluated in terms of *AUC*, we observe that VICTOR[MTL], with 0.91/0.91 AUC on average for $m = 2/m = 4$ respectively, slightly outperforming VICTOR[OC_r]: 0.91/0.89 and MID-only Transformer: 0.90/0.90. ViT B/32 reaches the highest OC_b AUC (0.92) for both m=2 and m=4 with either VICTOR[OC_r] or VICTOR[MTL].

By combining OC_r and MID in one model and tuning the hyper-parameter α, VICTOR[MTL] can perform consistently well on both tasks with all computer vision models. Based on TOPSIS, the overall best performance is reached by VICTOR[MTL;$\alpha = 0.2$;$m = 2$] with visual features from MLP-Mixer B/16.

5.3 Comparative Analysis

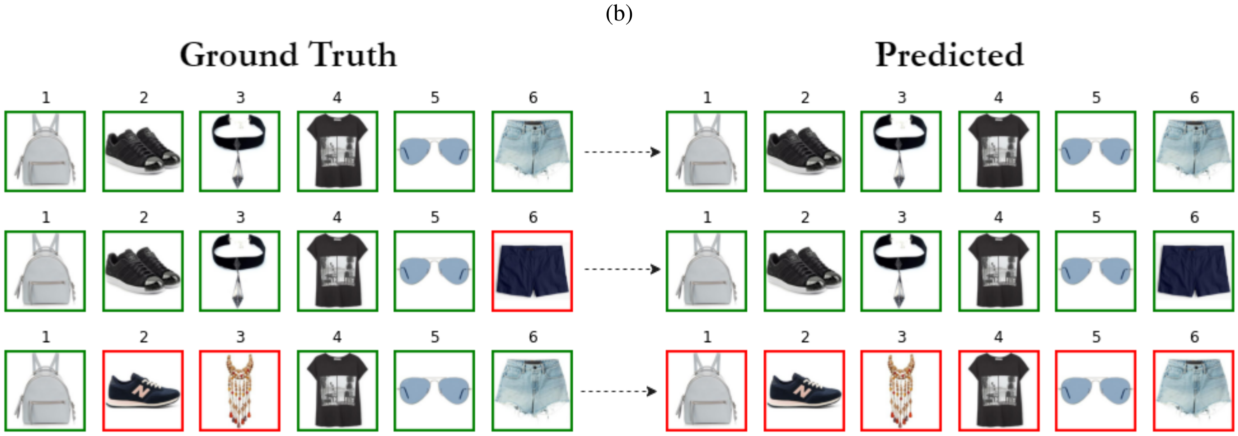
The central focus of this study is the detection of mismatching items in outfits which can be considered a sub-task of visual compatibility. We therefore compare the proposed VICTOR[MTL] with numerous state-of-the-art (SotA) models for binary outfit compatibility prediction (OC_b). The current SotA for visual-based OC_b on the Polyvore dataset is held by CSA-Net [Lin et al., 2020] and OutfitTransformer [Sarkar et al., 2022] with 0.91 AUC. When category information are added the performance of OutfitTransformer increases to 0.92 and when texts are also added it yields 0.93 AUC. However, these are not directly comparable with our work since we do not use category information nor texts.

Comparing the models that use pre-trained visual features on ImageNet, we observe that OutfitTransformer w/ ResNet18 (ImageNet) yields 0.82 AUC on Polyvore while our VICTOR[OC_b] w/ ResNet18 (ImageNet) outperforms it with 0.86 AUC. VICTOR[OC_b] exhibit a similar performance with the other three computer vision models, with an average AUC of 0.86. Furthermore, when employing $\mathbf{v}_{g_i}$ from FLIP, VICTOR[OC_b] w/ ResNet18 (FLIP) improves to 0.9 AUC similarly with all other computer vision models; that display an average AUC of 0.91 for Polyvore and 0.86 on Polyvore-D. The proposed VICTOR[MTL;$\alpha = 0.2$;$m = 2$] further improves upon VICTOR[OC_b] with MLP-Mixer B/16 and ViT B/32 FLIP models. This slight improvement may be due to Y_{OCr} forcing VICTOR[MTL] to learn deeper



(a)

(b)



(c)

Figure 4: Inference examples from VICTOR on fully compatible outfits and their generated partially mismatching versions. Green frames denotes compatible items while red frames denote incompatible items.

and more complicated relations compared to the simple OC_b-based model. Notably, VICTOR[MTL] w/ ResNet18 (FLIP) performs at the same level as the SotA while being significantly faster and less resource-intensive to train; requiring 94.8% fewer FLOPs. Finally, VICTOR[MTL] w/ MLP-Mixer B/16 (FLIP) or ViT B/32 (FLIP) surpasses the vision-based SotA with 0.92 AUC on Polyvore while VICTOR[MTL] with ViT/B32 (FLIP) surpasses the SotA on Polyvore-D with 0.88 AUC.

5.4 Qualitative Analysis

Fig.4 illustrates a few inference examples from VICTOR[MTL;$\alpha = 0.2$;$m = 2$] with MLP-Mixer B/16 since it exhibited the highest exact match score. We use samples from the Polyvore MISFITs $m = 2$ thus there are three fully compatible outfits and for each, there are two generated outfits containing at least one incompatible item. We observe that VICTOR is capable of correctly identifying the fully compatible outfits in all three cases (row=1 of each outfit). There are also cases that correctly identifies all mismatching items, row 2 and 3 of Fig. 4a and row 2 of Fig. 4b. VICTOR has presumably learned to "understand" which styles and colors of different garments can be matched together.

Expectedly, there are also some mistaken predictions. Row 3 of Fig.4b shows that $2/6$ items - while being annotated as compatible - they are predicted to be mismatching by the model. Similarly, in row 3 of Fig. 4c the whole outfit is predicted to be incompatible while $4/6$ items are annotated as compatible. On the other hand, row 2 of Fig.4c although the pair of dark navy shorts is annotated as mismatching with the rest of the outfit, some would consider this to be a mistaken annotation since grey, black and navy are often paired together. VICTOR seems to have generalized well enough so as to ignore the few rare cases of mistaken annotations.

One general challenge for the task of visual compatibility is that there is always the element of subjectivity. Moreover, what is considered compatible differs from culture to culture and is time dependent; since fashion trends are in constant flux. In our case, the ground truth compatible outfits reflect the subjective opinions and biases of fashion stylists from Polyvore, creating a data-driven bias in our models. Despite this caveat, overall, VICTOR seems to produce reasonable predictions and we believe that a larger and more diverse dataset would further improve its performance.

Finally, VICTOR does not only predict the mismatching items in an outfit but has also learned to predict the overall compatibility of an outfit. As a result it can be used for outfit recommendation. Fig. 5 illustrates an example where VICTOR detects two mismatching items in an outfit and given a set of candidate garments - all garments from the same category as the mismatching items - selects the better suited alternatives, resulting in a more cohesive and aesthetically pleasing outfit.

6 Conclusions

In this study we define two new sub-tasks within the general task of visual compatibility prediction, namely compatibility prediction as regression and mismatching item detection and examine both in the Fashion domain. We use the Polyvore outfits dataset to generate partially mismatching outfits (MISFITs) and create the Polyvore-MISFITs dataset where we perform a series of ablative and comparative experiments. We propose a multi-tasking Transformer-based architecture, named VICTOR, and utilize visual features from multiple computer vision neural networks fine-tuned with *fashion-specific contrastive language-image pre-training* (FLIP). The proposed architecture outperforms other methods by 4.87% in terms of AUC on the Polyvore dataset when using visual features extracted from ImageNet-pretrained models with no additional computational cost. Moreover, by using visual features from FLIP, VICTOR proves capable of competing and even surpassing state-of-the-art methods on Polyvore datasets that utilize end-to-end fine-tuning while reducing instance-wise floating point operations by 88%.

One limitation of the current study is that when generating the Polyvore-MISFITs dataset, we use random alternative sampling. More intricate methods could be implemented that take into account the rate of similarity between the ground truth and the selected mismatching garment. However, it is difficult to select the appropriate threshold of similarity without input from professional stylists. Selecting too similar items - e.g a black pair of dress shoes with another - would not result in actual mismatching outfits but selecting too dissimilar items - e.g the dress shoes with a pair of snow boots - would lead to numerous easy-to-predict MISFITs and as a result, VICTOR would not have learned to discern more subtle and useful cases of visual incompatibility. A second issue is that VICTOR has been trained on images from Polyvore dataset which depicts individual garments in a white background. This may limit its application to real-world fashion images as worn by people. However, VICTOR could very easily be integrated in a full system, similar to [Papadopoulos et al., 2022a], that applies garment detection to real world fashion imagery and then extract the visual features of individual garments; given that the garments are fully or mostly visible.

Our focus is centered around general visual compatibility in fashion. By relying on the Polyvore dataset VICTOR has learned to reflect the subjective opinions and biases of fashion stylists from Polyvore. It would be interesting for future works to re-create similar architectures that also take personalization into account [Zhan and Lin, 2021]. Finally, the proposed VICTOR and FLIP fine-tuning are not limited to applications within the Fashion domain. Future works could experiment with other visually-driven domains such as exterior and interior architecture design [Aggarwal et al., 2018].

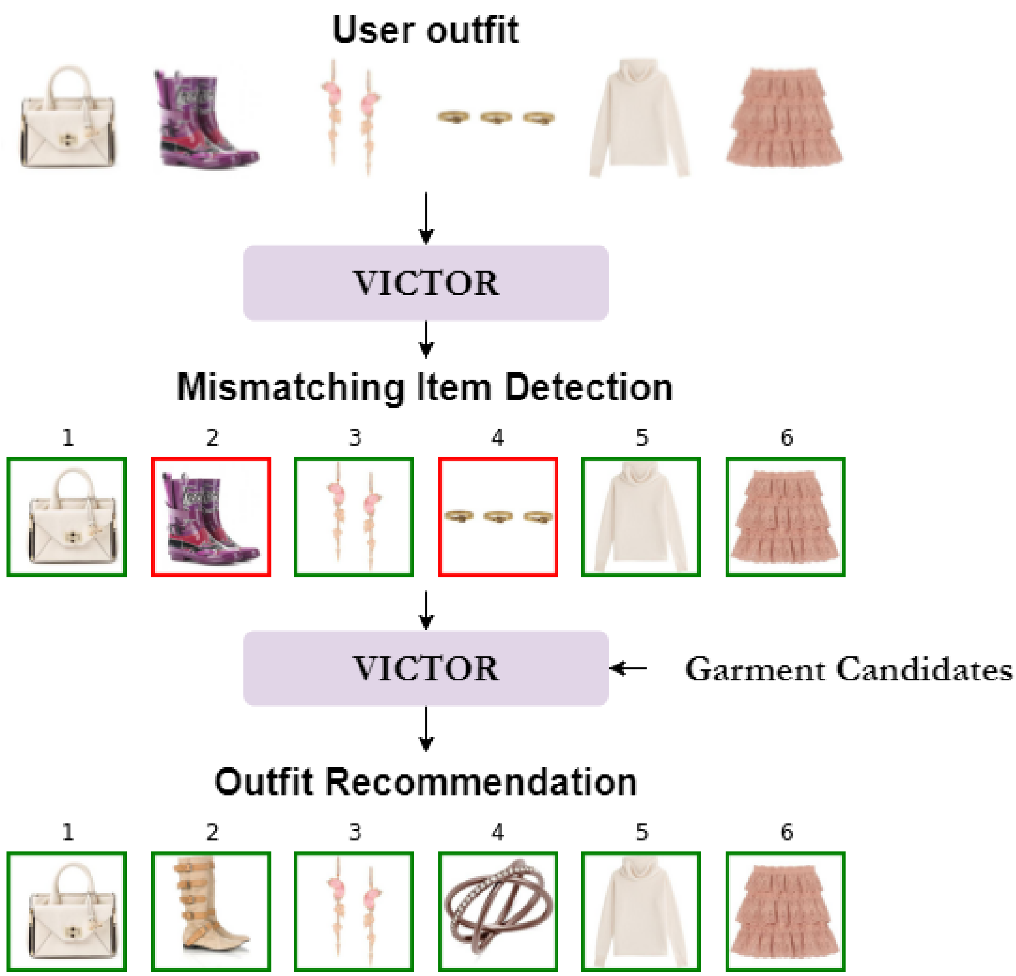


Figure 5: Example of VICTOR detecting the mismatching items in an outfit and recommending more compatible alternatives.

References

Reuben Tan, Mariya I Vasileva, Kate Saenko, and Bryan A Plummer. Learning similarity conditions without explicit supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10373–10382, 2019.

Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European conference on computer vision (ECCV)*, pages 390–405, 2018.

Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12617–12626, 2019.

Zeyu Cui, Zekun Li, Shu Wu, Xiao-Yu Zhang, and Liang Wang. Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks. In *The World Wide Web Conference*, pages 307–317, 2019.

Huijing Zhan and Jie Lin. Pan: Personalized attention network for outfit recommendation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2663–2667. IEEE, 2021.

Huijing Zhan, Jie Lin, Kenan Emir Ak, Boxin Shi, Ling-Yu Duan, and Alex C Kot. a^3-fkg: Attentive attribute-aware fashion knowledge graph for outfit preference prediction. *IEEE Transactions on Multimedia*, 24:819–831, 2021.

Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2662–2670, 2019.

Alexander Lorbert, David Neiman, Arik Poznanski, Eduard Oks, and Larry Davis. Scalable and explainable outfit generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3934, 2021.

Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1078–1086, 2017.

Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3311–3319, 2020.

Rohan Sarkar, Navaneeth Bodla, Mariya Vasileva, Yen-Liang Lin, Anurag Beniwal, Alan Lu, and Gerard Medioni. Outfittransformer: Outfit representations for fashion recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2263–2267, 2022.

Wen-Huang Cheng, Sijie Song, Chieh-Yun Chen, Shintami Chusnul Hidayati, and Jiaying Liu. Fashion meets computer vision: A survey. *ACM Computing Surveys (CSUR)*, 54(4):1–41, 2021. doi:10.1145/3447239.

Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1096–1104, 2016.

Stefanos-Iordanis Papadopoulos, Christos Koutlis, Manjunath Sudheer, Martina Pugliese, Delphine Rabiller, Symeon Papadopoulos, and Ioannis Kompatsiaris. Attentive hierarchical label sharing for enhanced garment and attribute classification of fashion imagery. In *Recommender Systems in Fashion and Retail*, pages 95–115. Springer, 2022a. doi:10.1007/978-3-030-94016-4_7.

Ziad Al-Halah, Rainer Stiefelhagen, and Kristen Grauman. Fashion forward: Forecasting visual style in fashion. In *Proceedings of the IEEE international conference on computer vision*, pages 388–397, 2017.

Utkarsh Mall, Kevin Matzen, Bharath Hariharan, Noah Snavely, and Kavita Bala. Geostyle: Discovering fashion trends and events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 411–420, 2019.

Geri Skenderi, Christian Joppi, Matteo Denitto, and Marco Cristani. Well googled is half done: Multimodal forecasting of new fashion product sales with image-based google trends. *arXiv preprint arXiv:2109.09824*, 2021. doi:10.48550/arXiv.2109.09824.

Stefanos I Papadopoulos, Christos Koutlis, Symeon Papadopoulos, and Ioannis Kompatsiaris. Multimodal quasi-autoregression: Forecasting the visual popularity of new fashion products. *arXiv preprint arXiv:2204.04014*, 2022b.

Hyunwoo Hwangbo, Yang Sok Kim, and Kyung Jin Cha. Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, 28:94–101, 2018.

Maria Anastassia Stefani, Vassilios Stefanis, and John Garofalakis. Cfrs: a trends-driven collaborative fashion recommendation system. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–4. IEEE, 2019.

Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015.

Meet Taraviya, Anurag Beniwal, Yen-Liang Lin, and Larry Davis. Personalized compatibility metric learning. In *KDD Workshop*, volume 1, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021.

Ching-Lai Hwang and Kwangsun Yoon. Methods for multiple attribute decision making. In *Multiple attribute decision making*, pages 58–191. Springer, 1981.

Kedan Li, Chen Liu, and David Forsyth. Coherent and controllable outfit generation. *arXiv preprint arXiv:1906.07273*, 2019.

Divyansh Aggarwal, Elchin Valiyev, Fadime Sener, and Angela Yao. Learning style compatibility for furniture. In *German Conference on Pattern Recognition*, pages 552–566. Springer, 2018.