Attentive Hierarchical Label Sharing for Enhanced Garment and Attribute Classification of Fashion Imagery

Stefanos-Iordanis Papadopoulos, Christos Koutlis, Manjunath Sudheer, Martina Pugliese, Delphine Rabiller, Symeon Papadopoulos and Ioannis Kompatsiaris

Abstract Fine-grained information extraction from fashion imagery is a challenging task due to the inherent diversity and complexity of fashion categories and attributes. Additionally, fashion imagery often depict multiple items while fashion items tend to follow hierarchical relations among various object types, categories and attributes. In this study, we address both issues with a 2-step hierarchical deep learning pipeline consisting of (1) a low granularity object type detection module (upper-body, lower-body, full-body, footwear) and (2) two classification modules for garment categories and attributes based on the outcome of the first step. For the category and attribute-level classification stages we examine a hierarchical label sharing (HLS) technique in two settings: (1) single-task learning (STL w/ HLS) and (2) multi-task learning with RNN and visual attention (MTL w/ RNN+VA). Our approach enables progressively focusing on appropriately detailed features for automatically learning the hierarchical relations of fashion and enabling predictions on images with complete outfits. Empirically, STL w/ HLS reached 93.99% top-3 accuracy while MTL w/ RNN+VA reached 97.57% top-5 accuracy for category

Stefanos-Iordanis Papadopoulos
CERTH-ITI, e-mail: stefpapad@iti.gr

Christos Koutlis
CERTH-ITI, e-mail: ckoutlis@iti.gr

Martina Pugliese
Mallzee, e-mail: martpugliese@gmail.com

Manjunath Sudheer
Mallzee, e-mail: manjunath@mallzee.com

Delphine Rabiller
Mallzee, e-mail: delphine@mallzee.com

Symeon Papadopoulos
CERTH-ITI, e-mail: papadop@iti.gr

Ioannis Kompatsiaris
CERTH-ITI, e-mail: ikom@iti.gr

classification on the DeepFashion benchmark, surpassing the current state-of-the-art without requiring landmark or mask annotations nor specialised domain expertise.

1 Introduction

Fashion, being a primarily visually-driven domain, has recently attracted the interest of computer vision researchers for numerous tasks, including attribute recognition, landmark detection, outfit matching and item retrieval [1]. In this study, we address two central pattern recognition problems on fashion imagery, namely clothing category and attribute classification. Contrasted with other domains, fashion datasets tend to be more diverse and fine-grained - relating to categories, patterns, styles, textile materials, colors, length among many others - rendering the training of deep learning computer vision models a rather challenging endeavour. Additionally, categories and attributes follow hierarchical relationships, meaning that certain attributes apply only to specific categories of certain object types (e.g. tie-front > blouse > upper-body). Finally, fashion imagery often depicts multiple garment items per image increasing the complexity of the problem.

To address the aforementioned problems, we propose a hierarchical two-stage deep learning pipeline that employs "hierarchical label sharing" (HLS). Essentially, HLS shares the predicted label of the previous task to the next; meaning the sharing of the object type labels with the category-level classifier and the predicted category with the attribute classifier. Our two-stage pipeline first identifies low granularity object types (upper-body, lower-body, full-body, footwear) in fashion images and then classifies the corresponding bounding boxes with regards to category and fine-grained attributes. For the second stage, we examine the performance of HLS in two settings: 1) single-task learning (STL w/ HLS) and 2) multi-task learning (MTL) with a recurrent neural network (RNN) and visual attention (VA). The "MTL w/ RNN+VA" method incorporates HLS in the RNN decoder. We train and evaluate both methods on DeepFashion [2], a widely used fashion dataset and the *Mallzee datasets* - created by Mallzee[1] - that also includes footwear. Combining a Faster R-CNN with an InceptionV2 backbone for object type detection and an EfficientNet-B4 architecture for category and attribute classification, we were able to surpass the state-of-the-art on category classification on the DeepFashion dataset. More specifically, "STL w/ HLS" scored 93.99% top-3 accuracy while "MTL w/ RNN + VA" scored 97.57% top-5 accuracy; the highest achieved scores for the dataset. A significant advantage of HLS is the ability to learn hierarchical relations among fashion attributes/categories/object types without requiring manually crafted rules by domain experts. Additionally, "MTL w/ RNN+VA" has the advantage of producing attention plots which are useful for interpreting the model's predictions. Finally, our two-stage pipeline has the ability to work efficiently with real-world fashion imagery without requiring further types of annotation.

[1] Mallzee is a fashion product aggregator company that provides mobile e-commerce services (https://mallzee.com/)

The main contributions of our work can be summarised as follows:

- We propose two novel hierarchical methods for category and attribute classification on fashion imagery, "STL w/ HLS" and "MTL w/ RNN+VA", that compete with the domain's state of the art.
- We utilize a two-stage pipeline recognising patterns in full-scale fashion imagery depicting multiple garments.
- We expand our analysis to footwear that are missing from DeepFashion (DF1) [2] and DeepFashion2 (DF2) [3] benchmarks.

2 Related Work

Recently, research on fashion related image classification has received a lot of attention from multiple deep learning disciplines, including image processing [1], multi-modal [4] [5] and scene graph learning [6]. In this section, we will mainly focus on the first, image processing, since it is more relevant to our work. Generally, category classification on fashion imagery is formulated as a multi-class classification problem while attribute classification as a multi-label classification problem whose objective is to identify fine-grained attributes relating to styles, patterns, fabric and length among others.

DeepFashion is a publicly available and widely used fashion related dataset, created by Liu et al. (2016) [2]. In the original paper, Liu et al. (2016), proposed FashionNet in order to assess the usefulness of DeepFashion. FashioNet jointly learned to predict clothing landmarks and attributes. The model was trained end-to-end to first estimate the landmark locations, pool/gate the extracted features and then identify the relevant attributes. Since the publication of DeepFashion and the development of FashionNet, the former has been used as a benchmark dataset and the latter as a baseline model for category and attribute classification in fashion.

Corbiere et al. (2017) [7] utilized a weakly supervised approach that learns from large-scale noisy data. Their model was trained contrastively, with the use of negative sampling, from image-text pairs with noisy and mostly unprocessed texts. By fine-tuning a dense layer on the DeepFashion dataset this approach was able to outperform FashionNet in texture and shape-related attributes but not on the overall evaluation.

Wang et al. (2018) [8] proposed the incorporation of domain knowledge to improve attribute classification by developing a fashion grammar capturing kinetic and symmetrical relationships between clothing landmarks. The researchers introduced a bidirectional convolutional recurrent network that leveraged their fashion grammar during the landmark prediction process. After being trained for landmark prediction, two branched fully connected layers were added and trained for category and attribute classification.

Ye et al. (2019) [9] combined cost-sensitive learning and over-sampling in order to rectify the problem of class imbalance present in DeepFashion. The authors introduced a weighted loss function that only back-propagated the most informative

nodes, therefore focusing on minority classes as the training progresses, and combined it with a semi-supervised Generative Adversarial Network for over-sampling the minority classes by producing synthetic samples.

Finally, Li et al. (2019) [10], designed a multi-task model that is trained end-to-end for landmarks, category, and attributes detection. They incorporated two knowledge sharing techniques regarding boundary awareness and structure awareness for transferring relevant information across tasks, yielding 93.01% top-3 accuracy on category classification and 59.83% top-3 recall on attribute classification, reaching the currently highest score on DeepFashion. The results for all aforementioned studies can be seen in Table 7.

A limitation of DeepFashion is that its images are annotated with only one clothing item even if more garments are visible. This phenomenon can create challenges during both training and evaluation stages [3]. To mitigate this problem, all aforementioned research works applied cropping of the training and testing images using the ground truth bounding boxes. However, models solely trained on cropped images can only extract local features from the image and thus can not generalise to images with complete outfits. To overcome this issue, we propose a hierarchical pipeline that separates the tasks of object type detection, category classification and attribute classification. A second limitation of DeepFashion is its lack of footwear, a significant product category for the fashion industry. For this reason, by utilizing Mallzee's database we created a new dataset that includes rich annotations about categories and attributes related to footwear in addition to upper-body, lower-body and full-body garments.

3 Methodology

3.1 Problem formulation

Pattern recognition on fashion imagery is considered a rather challenging task due to the inherent diversity and complexity of fashion items and their relations [1]. Fashion items tend to follow hierarchical relations between different object types, categories and attributes. "object types" are considered high-level descriptors of garments denoting their relation to the human body. Garments can be classified into upper-body, lower-body, full-body garments and footwear. Furthermore, fashion items can be classified into various categories and attributes. In this context, category classification is defined as a multi-class task consisting of mid-level descriptors such as "dress", "shirt", "trousers", while attribute classification is defined as a multi-label task with fine-grained descriptors such as "floral", "pencil", "frill". In fashion, object types, categories and attributes tend to follow hierarchical relations. Following the previous examples, a "dress" belongs to the "full-body" object type and "shirt" in "upper-body". The "frill" attribute is applicable to "dresses" - as a specialised "style" of "dress" - but not for example to footwear. However, other attributes such as "floral" can theoretically be applied to all categories of garments since it describes the print

or pattern of the garment. Another common challenge in the fashion domain is that fashion imagery often depict complete outfits. Thus, in production settings, a pattern recognition model should be able to recognise multiple object types and their location in full-scale images. To address both challenges, we propose a hierarchical two-stage deep learning pipeline that utilizes hierarchical label sharing (HLS) for automatically learning relations among garment object types, categories and attributes in full-scale fashion imagery.

3.2 Proposed architectures

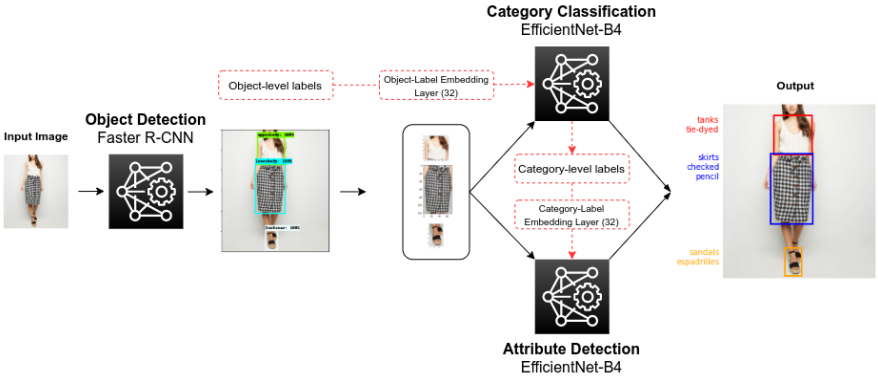
In this study we attempt to improve the classification of garment categories and fine-grained attributes by automatically learning the existing hierarchical relations among garment attributes, categories and object types while being able to perform predictions on fashion imagery with complete outfits. To this end, we propose a hierarchical two-stage deep learning pipeline that employs hierarchical label sharing (HLS). HLS shares the predicted labels from the previous stage to the next; meaning that the object type is shared with the category classifier and the category label is shared with the attribute classifier. Our proposed pipeline follows two-stages 1) object type detection and 2) category and attribute classification. A similar method has been applied for self-driving cars [11] but to the best of our knowledge, this is the first time it is attempted in the fashion domain.

3.2.1 First stage: object type detection

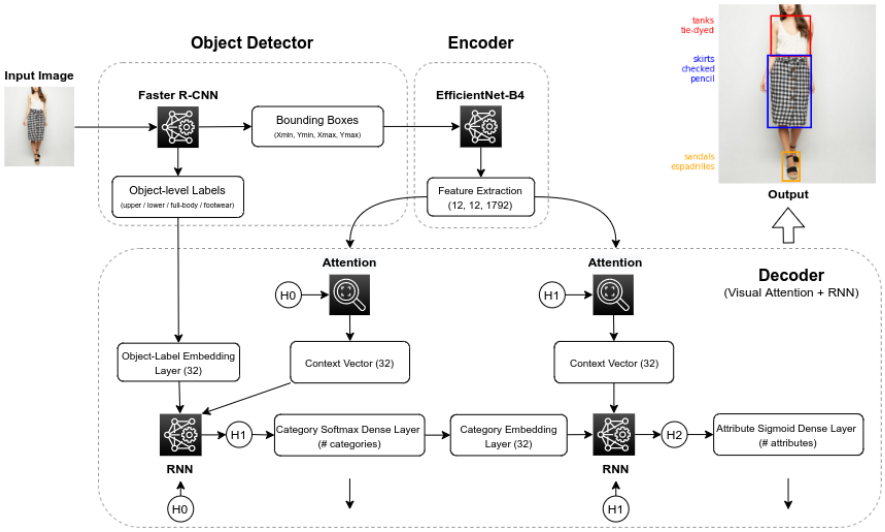
The first stage of our deep learning pipeline performs object type detection that identifies the object type and location (bounding boxes) of fashion-related objects on full-scale images. We utilized transfer learning where object type detection models, pre-trained on large-scale image dataset, are fine-tuned on a new dataset. We experimented with variants of Faster-RCNN [12], CenterNet [13] and EfficientNet-D1 and D2 [14]. These models were fine-tuned on the *Mallzee dataset* and the DF2 datasets; discussed in Section 4.1. Afterwards, we passed the images through the fine-tuned object type detection model and extracted the bounding boxes of each detected garment and its object type label.

3.2.2 Second stage: Category and attribute classification

The second stage of our proposed pipeline performs category and attribute classification employing HLS. We examine how HLS works in two settings: 1) single-task learning with hierarchical label sharing (STL w/ HLS) and 2) multi-task learning with RNN and visual attention (MTL w/ RNN+VA); that employs HLS in the RNN



(a) Workflow for "single-task learning" (STL) and "single-task learning with hierarchical label sharing" (STL w/ HLS) methods. The red intermittent line is only applied on the STL w/ HLS method where the label from the previous stage is shared with the next; meaning the object type label is shared with the category classifier and the category label is shared with the attribute classifier.



(b) Workflow for the "multi-task learning with RNN and visual attention" method relying on an encoder-decoder architecture. The vision encoder produces the image representations which are passed through an attention mechanism. In the initial stage, the attentive context vector is given to the RNN alongside the object type label embeddings and the hidden state 0 (H0), a series of zero values. The first resulting hidden state (H1) is passed through two fully connected layers of which the latter is activated by a softmax function with units equal to the number of categories. The context vector is re-calculated given the same image features and the H1 which are passed through the RNN alongside the predicted category embeddings and the H1. The predicted H2 is passed through two fully connected layers of which the later, with units equal to the number of attributes, is activated by a sigmoid function.

Fig. 1: Workflows for the proposed hierarchical methods.

decoder. However, in order to assess the effectiveness of HLS we also define a conventional "STL" method that does not utilize HLS (referred to as "Baseline STL").

Baseline Single task learning (STL). The training workflow for the Baseline STL follows a conventional image classification process. The images are first passed through the fine-tuned object detection model and the identified fashion-related object types are cropped around the predicted bounding boxes. The cropped images are passed through an image augmentation pre-processing layer that performs alterations to the images as a method of regularisation and by extension the mitigation of overfitting. More specifically, the images are horizontally flipped at random and are randomly rotated and zoomed by ±10%. The augmented images are then passed through the base convolutional encoder model. The features extracted from the last convolutional layer of the base model are pooled with the use of global average pooling thus transforming the output into a 2D tensor. A classification dense layer is added on top. In the case of category classification (a multi-class problem) the dense layer is activated by a softmax while for attribute classification (mutli-label task), the last dense layer is activated by a sigmoid function. Subsequently, the model is trained by an adaptive gradient descent optimizer (Adam or RMSProp) which performs gradient updates per individual parameter. The network's loss is calculated by the categorical cross-entropy and the binary cross-entropy for category and attribute classification respectively. The workflow is shown in Figure 1a (without the intermittent lines).

Hierarchical label sharing (HLS). Our first approach, "STL w/ HLS", follows the same workflow as "Baseline STL" with the addition that the predicted labels from the previous stage are passed to the next. This means that the object type labels for each image are shared with the category classifier, after being passed through an embedding layer and then concatenated with the image representation extracted from the convolutional backbone. Similarly, the garment-level category is shared with the attribute classifier. We hypothesize that HLS will be instrumental in automatically learning hierarchical relations among fashion attributes, categories and object types. The workflow for this method is shown in Figure 1a when applying the red intermittent lines.

Multi-task learning with RNN and visual attention. Our second approach, "MTL w/ RNN+VA" method relies on multi-task learning (MTL) where the same neural network is fine-tuned by two separate loss functions simultaneously. The network is optimised for both category and attribute classification simultaneously based on the categorical cross entropy and the binary cross entropy loss functions respectively. Our MTL architecture - partly inspired by [15] - integrates HLS in a recurrent neural network (RNN) decoder, more precisely a Gated Recurrent Unit (GRU). Moreover, we employ the attention mechanism proposed by [16] in order to enable the model to focus only on the relevant part of the image for the prediction of certain categories and attributes. A notable advantage of this approach is the ability to plot attention weights and thus interpret the model's predictions.

More specifically for our implementation, each image first passes through a vision encoder convolutional neural network (CNN) in order to extract its visual features $F = [f_1, f_2, \ldots, f_n]$, where $f_i \in \mathbb{R}^d$. The attention mechanism receives as input the

sequence of vectors F along with the previous GRU hidden state $h_{t-1} \in \mathbb{R}^s$ and calculates the "context" vector $c_t \in \mathbb{R}^d$ with $t = 1, 2$. The attention weights a_{ti} and the context vector c_t are calculated as below:

$$e_{ti} = V \cdot tanh((W_1 f_i + b_1) + (W_2 h_{t-1} + b_2)) \tag{1}$$

$$a_{ti} = \frac{exp(e_{ti}))}{\sum_{j=1}^{n} exp(e_{tj})} \tag{2}$$

$$c_t = \sum_{j=1}^{n} a_{tj} f_j \tag{3}$$

where $V \in \mathbb{R}^u$, $W_1 \in \mathbb{R}^{u \times d}$, $W_2 \in \mathbb{R}^{u \times s}$ and $b_1, b_2 \in \mathbb{R}^u$ are trainable parameters.

The calculated context vector is then concatenated with the object type embeddings and given as input to the decoder network in order to calculate the next hidden state:

$$h_1 = GRU(h_0, [c_1; E_o]) \tag{4}$$

where E_o are learnable embedding vectors for each object type o and h_0 is initialized with zeros. The output of the decoder h_1 is then passed through two fully connected layers:

$$U_2^c(U_1^c(h_1)) \tag{5}$$

of which the final U_2^c classifies the image into a garment category, through softmax activation. Consequently, the category and the decoder's previous state h_1 are given again to the decoder which performs the same process for the attribute classification:

$$h_2 = GRU(h_1, [c_2; E_c]) \tag{6}$$

$$U_2^a(U_1^a(h_2)) \tag{7}$$

where E_c are learnable embedding vectors for each garment category c. The MTL learning pipeline can be seen in Figure 1b. Finally, the loss function to be minimized is the sum of the the two losses (categorical and binary cross entropy) related to category and attribute classification respectively.

For the second stage we experiment with InceptionV3, Xception and EfficientNets from B0 to B4 with pre-trained weights on ImageNet or from self-training with noisy students [17]. These models consist of relatively limited number of parameters but have been shown to perform very well on the ImageNet benchmark[2]. For each model we also perform hyper-parameter tuning based on the validation accuracy, for identifying the optimal learning rate, dropout rate, batch size and the optimal number of pre-trained layers to fine-tune.

[2] https://paperswithcode.com/sota/image-classification-on-imagenet

4 Experimental Setup

In this section we discuss the datasets and evaluation metrics used in our study. For our experiments, we made use of two publicly available, large-scale, fashion image datasets, DeepFashion (DF1) [2] and DeepFashion2 (DF2) [3]. Additionally, we created a new dataset, which we term *Mallzee dataset*, that also contains annotations for footwear that were lacking from both DF1 and DF2.

4.1 Datasets

Public fashion datasets. DF1 consists of 289,222 images with rich annotations regarding 46 clothing categories and 1,000 fine-grained attributes related to textures, fabrics, shapes, parts and styles. A significant limitation of DF1 is that each image contains only one annotated garment even if more than one are visible. DF2 consists of 491,895 images in total annotated on 13 categories. Moreover, DF2 expands upon DF1 by including pixel-level mask annotations and by annotating multiple items per image.

We mainly utilize DF1 for the category and attribute classification and DF2 for the task of object type detection. The categories of DF2 are re-mapped as seen in Table 1 to fit our needs for low-granularity fashion-related object types.

Additionally, we calculate the Imbalance Ratio (IR), the MeanIR and SCUMBLE metrics for each class in order to examine the level of class co-occurrence and imbalance in multi-label data [18].

$$IR(y) = \frac{\arg\max_{y'=Y_1}^{Y_{|Y|}} (\sum_{i=1}^{|D|} h(y', Y_i))}{\sum_{i=1}^{|D|} h(y, Y_i)}, h(y, Y_i) = \begin{cases} 1 & y \in Y_i \\ 0 & y \notin Y_i \end{cases} \tag{8}$$

$$MeanIR = \frac{1}{|Y|} \sum_{y=Y_1}^{Y_{|Y|}} IR(y) \tag{9}$$

$$SCUMBLE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} [1 - \frac{1}{IR_i} (\prod_{l=1}^{|L|} IR_{il})^{1/|L|}] \tag{10}$$

where D is a multi-label dataset, Y its full set of labels, y the label being analyzed, and Y_i the labelset of the i-th instance in D. IR, shown in Eq. 8, is calculated individually for each label as the ratio between the majority class divided by all other classes individually. MeanIR, calculated by Eq. 9, simply reflects the mean value across all IRs. SCUMBLE, calculated by Eq. 10, takes into account both the quotient and product among the IR. The initial re-mapped DF2 had a meanIR of 1.6528 and a SCUMBLE metric of 0.1321 but after randomly down-sampling the dataset, with 42,000 objects per class in the training set and 8,000 objects per class in the validation set, the meanIR is reduced to 1 and SCUMBLE to 0, their optimal values.

Table 1: Garment categories per object type, for the DeepFashion2 (DF2) dataset and the *Mallzee dataset*. DF2 categories were re-mapped into the object type labels for the object type detection task.

Object type	**DF2 categories**	**Mallzee dataset categories**
Upper-body	Short sleeve top, Long sleeve top, Short sleeve outwear, Long sleeve outwear, Vest, Sling	Sweaters, Blouses, Coats and jackets, Formal jackets, Shirts, Hoodies, Camis, Tshirts, Cardigans, Tanks
Lower-body	Shorts, Trousers, Skirt	Shorts, Trousers, Skirts, Jeans
Full-body	Short sleeve dress, Long sleeve dress, Vest dress, Sling dress	Dresses, Jumpsuits, Playsuits
Footwear	-	Boots, Flats, Heels, Sandals, Trainers

Mallzee dataset. Considering the importance of footwear in the fashion industry, we deem their lack from both DF1 and DF2 to be a significant shortcoming. Therefore, we decided to create a new dataset that also includes annotated footwear imagery. The dataset is sourced from Mallzee, a popular fashion e-commerce website, and it comprises three parts, one per task: object type detection, category-level classification and attribute-level classification.

For the object type detection task, we created a dataset with annotated bounding boxes around the garments. The "object type detection dataset" consists of four classes: upper, lower, full-body and footwear. Our objective was to create a balanced dataset in terms of all four classes. To this end, we manually annotated images with bounding boxes with the use of LabelImg[3]. In order to get more images per class, we made use of Haar Cascades[4] for automatically detecting bounding boxes from flat-lay garments. Furthermore, in contrast to DF1 - that only contains one annotation per image - we annotated all garments present in an image. During the manual annotation process, our central criterion was that we annotated garments only if they were fully visible; or visible to a significant degree. The resulting "object type detection dataset" was still naturally imbalanced; since for example footwear often come in pairs. To overcome this issue, we created a more balanced dataset by up-sampling the minority classes with the use of image augmentation techniques[5]. More specifically, we applied vertical flips and rotations for flat-lay images and the same plus horizontal flips for manual annotations. The final "object type detection dataset" comprises 5,343 images depicting 15,359 objects: 5,160 upper-body, 3,749 lower-body, 4,985 full-body and 5,554 footwear images. A separate sample of 634

[3] https://github.com/tzutalin/labelImg

[4] https://github.com/opencv/opencv/tree/master/data/haarcascades

[5] We did not include instances that contained upper-body garments; being the majority class.

Table 2: Attribute key-value pairs for the *Mallzee dataset* following a taxonomy and the categories they can be applied to.

Att. key	Attribute value	Applies to
Prints	Ethnic, Graphic, Floral, Tropical, Striped, Checked, Animal Print, Polka dots, Paisley, Spots, Tartan, Geometric, Colourblock, Fair isle, Camouflage, Grid print, Dip-dyed, Tie-dyed, Zigzag, Washed	All objects & categories
Styles	Duster, Parka, Trench, Peacoat, Coatigan, Duffle, Bomber, Biker, Puffer, Anorak, Windcheater, Borg Windbreaker, Coach, Quilted, Trucker	Coats & Jackets
	Wedding, Bridesmaid, Bodycon, Tunic, Jumper, Shirt, Shift, Slip, Tea, Cocktail Pinafore, Wrap, Sundress, Smock	Dresses
	Sweatpants, Leggings, Culottes, Peg, Harem, Capri, Formal, Chino	Trousers
	Sweatshorts, Cutoff, Bermuda, Skort, Running, Cycling	Shorts
	Wellies, Winter, Chukka, Chelsea Biker, Cowboy, Sock	Boots
	Boxy, Varsity, Baseball, Boyfriend, Fitted, Sport, Polo, Muscle	T-shirts
	Loafers, Brogues, Ballerinas, Plimsolls, Boat, Moccasins	Flats
	Gladiator, T-bar, Toe-thong Flip-flops, Sliders	Sandals
	Pencil, Skater, A-line, Frill, Flowy	Skirts & Dresses
	Mules, D'orsay, Espadrilles	Flats & Heels
	Jeggings, Mom, Boyfriend	Jeans
	Tie-front, Bib, Popover	Blouses
	Formal, Collarless	Shirts
	Waterfall	Cardigans
	Blazer	Formal jackets
	Cargo	Trousers & Shorts
	Court	Heels

Table 3: Details for DeepFashion, DeepFashion2 and the *Mallzee dataset*.

Dataset	Images	Categories	Attributes
DeepFashion	289,222	46	1000
DeepFashion2	491,895	13	-
Mallzee (object-level)	16,550	4	-
Mallzee (category-level)	229,633	22	-
Mallzee (attribute-level)	310,753	16	110

images depicting 1,191 objects (303 upper-body, 205 lower-body, 175 full-body and 508 footwear) is used for evaluation.

For the category-level and attribute-level classification datasets we did not rely on manual annotation. Instead we retrieved the images from Mallzee's database with queries related to 22 garment categories and the 110 attributes classes related to patterns/prints and styles. We queried the target categories and attributes names and all their synonyms we could identify. We applied a series of rules and regular expressions for ensuring minimal mismatch rate.The final datasets comprise 229,633 and 310,753 images for category and attribute classification, respectively. In order to mitigate minor class imbalances we applied image augmentation for the minority classes; similarly to the "object-detection dataset". The 22 category labels can be seen in Table 1 grouped by their object type. Similarly, the attribute key-value pairs can be seen in Table 2, following Mallzee's fashion taxonomy. Table 3, presents and compares the statistics of DeepFashion, DeepFashion2 and Mallzee datasets. The attribute-level Mallzee dataset does not contain all 22 categories. Instead it has 16 slightly broader clustered labels which include: 'activewear', 'boots', 'dresses', 'flats', 'heels', 'jackets', 'jeans', 'shirts', 'shoes', 'shorts', 'skirts', 'sweaters', 'tops', 'trainers', 'trousers', and 't-shirts'. For our experiments, the two datasets were randomly shuffled and split into training, validation and testing sets with 0.8, 0.1, 0.1 ratios respectively.

4.2 Evaluation

For the evaluation of the object type detection task we rely on the COCO-challenge metrics[6] for the object type detection metrics, with a focus on the mean average precision (mAP) metric averaged over 10 intersections over union thresholds (IoU) (from 0.5 to 0.95 with steps of 0.05 size) - which is the central metric of the competition - and the average Recall@K (AR@K) that signify the average recall given K detections per image. For category classification we rely on top-K accuracy for K=3,5 and the recall rate at top-K for K=3,5 for attribute classification; since these metrics are also used to benchmark DF1 [2]. For experiments on the Mallzee dataset we also calculate and report the metrics for K=1. All eight methods presented in the results section with which we compare our proposed architectures, have been previously mentioned and elaborated in Section 2.

[6] https://cocodataset.org/#detection-eval

5 Results

5.1 Object type detection

On the DF2 dataset, CenterNet consistently shows the highest performance; scoring a mean average precision (mAP) of 75.6% and an average recall at 100 (AR@100) of 86.9%. On the other hand, using the *Mallzee* object type detection dataset, a Faster R-CNN model yields the highest scores with a mAP of 80.6% and an AR@100 of 85.8% for 4 classes. The performance of all object type detection models can be seen in Table 4. We use the default hyper-parameters for all models, as given by the TensorFlow object detection API[7]. The only exceptions are the batch size that is reduced to 2 (so as to fit in the GPU memory) while the learning rate is reduced to 1e-4 for EfficientDet-D1 and D2 since the default value is very high and causes overfitting during fine-tuning.

Table 4: Object type detection models trained on the re-mapped DeepFashion2 (DF2) and the *Mallzee dataset*. Evaluations performed on test sets of each dataset in terms of mean Average Precision (mAP) and Average Recall at 100 (AR@100). The re-mapped DF2 includes three classes (upper/lower/full-body) while the *Mallzee dataset* also includes footwear. (Bold denotes the best performing model by metric)

Evaluation Metric	Training	Evaluation	Faster R-CNN	EfficientDet-D1	EfficientDet-D2	CenterNet
mAP	DF2	DF2	73.0	72.1	64.8	**75.6**
AR@100			83.7	81.9	76.7	**86.9**
mAP	DF2	*Mallzee*	73.9	73.6	69.8	**76.9**
AR@100			84.2	82.8	80.4	**86.8**
mAP	*Mallzee*	*Mallzee*	**80.6**	62.6	60.5	73.2
AR@100			**85.8**	75.2	70.7	80.7

5.2 Category and attribute classification

5.2.1 Results on the *Mallzee dataset*

For the category and attribute classification tasks we experiment with hierarchical label sharing (HLS) in two settings, 1) single-task learning (STL w/ HLS) and 2) multi-task learning (MTL) with RNN and visual attention (VA). As baseline comparison we define a conventional STL method that does not utilize HLS; named

[7] https://tensorflow-object-detection-api-tutorial.readthedocs.io

Table 5: Single-task learning models trained and evaluated on the *Mallzee dataset* for category classification. (Bold denotes the best performing model by dataset)

Training Dataset	# of categories	Network	Accuracy@1
Full-body only	3	EfficientNet-B1	91.77
		EfficientNet-B2	92.34
		EfficientNet-B3	93.63
		EfficientNet-B4	**94.90**
		Xception	93.46
		InceptionV3	91.55
Lower-body only	4	EfficientNet-B1	91.79
		EfficientNet-B2	91.46
		EfficientNet-B3	93.16
		EfficientNet-B4	**94.44**
		Xception	93.75
		InceptionV3	92.70
Footwear only	5	EfficientNet-B1	93.39
		EfficientNet-B2	93.89
		EfficientNet-B3	93.94
		EfficientNet-B4	**93.86**
		Xception	93.17
		InceptionV3	91.87
Upper-body only	10	EfficientNet-B1	85.31
		EfficientNet-B2	86.04
		EfficientNet-B3	86.33
		EfficientNet-B4	**88.15**
		Xception	86.29
		InceptionV3	83.50
Full dataset	22	EfficientNet-B3	91.09
		EfficientNet-B4	**92.24**

Baseline STL. Additionally, we experiment with training four separate STL models, one for each object type. This approach could theoretically improve the specialization of each model and thus improve their predictive accuracy. As can be seen in Table 5, a fine-tuned EfficientNet-B4 for all 22 categories on the *Mallzee dataset*, yields a 92.24% accuracy@1 while the mean accuracy of the four separate models has a slight +0.6% advantage. However, having a single model for all classes will not suffer when encountering misclassified items from the object type detection phase. Meaning that for example, an object wrongly classified as an "upper-body" type while actually being "full-body" will be passed in the wrong model that has not been trained to recognise upper-body type items. We consider that the very slight advantage in accuracy is outweighed by the aforementioned disadvantage. Therefore, having a hierarchical STL architecture - where the output of the object type detection stage is passed directly to separate specialised classifiers - is not deemed optimal.

Table 6: Comparing the two hierarchical label sharing methods with the baseline STL, for category and attribute classification when trained and evaluated on the *Mallzee dataset*. STL stands for single task learning and MTL for multi-task learning. (Bold denotes the best performance)

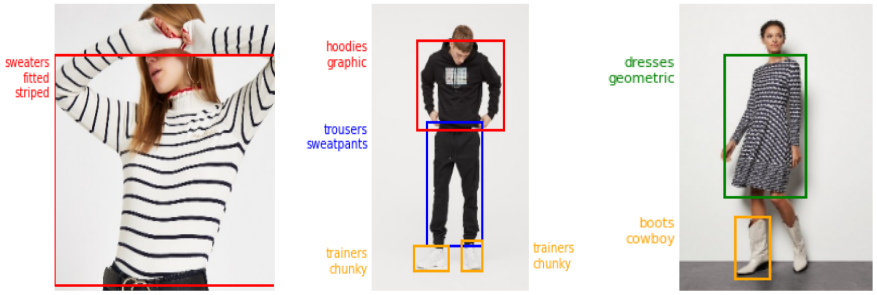
Method	**Category**			**Attributes**		
	Top-1	Top-3	Top-5	Top-1	Top-3	Top-5
Baseline STL	90.10	98.55	99.48	**78.75**	**94.32**	96.78
STL w/ HLS	**90.67**	**98.72**	**99.53**	77.62	93.70	**96.80**
MTL w/ RNN+VA	87.63	98.01	99.31	62.74	83.81	90.32

The results show that the EfficientNet-B4 architecture consistently outperforms all other models when its 100 to last layers (from 474 in total) are fine-tuned with the exception of the batch normalization layers. The aforementioned number of layers concerns the Keras[8] implementation of EfficientNet and includes 'reshaping', 'reduce', 'multiplication' and 'activation' layers as well as the number of convolutional, dense and batch normalisation layers. Fine-tuning additional layers does not further improve the model's overall performance while also increasing the required computational resources and time. As expected, we found that low learning rates, either 1e-4 or 5e-5, are optimal for all cases of fine-tuning the pre-trained networks. Moreover, higher dropout rates offer stronger regularisation which translates into increased training stability, less overfitting and overall improved performance. Finally, making use of pre-trained weights from self-trained EfficientNets with noisy students [17] improves the accuracy of the architecture when compared to using weights pre-trained exclusively on ImageNet. We apply the aforementioned insights from Baseline STL to both tasks and both methods employing HLS. For the following analysis we use all images of the Mallzee dataset but we did not utilize all 22 categories. Instead we mapped the categories onto the 16 categories found in the attribute-level dataset; so as to ensure the comparability of results.

As illustrated in Table 6, for category classification on the *Mallzee dataset*, STL w/ HLS outperforms the other methods in terms of top-1, top-3 and top-5 accuracy on categories. For attribute classification, HLS does not seem to further improve STL's performance with the exception of a negligible +0.02% in terms of top-5 recall for attributes. MTL w/RNN+VA has the lowest performance of the three settings across all metrics.

Finally, we created a merged pipeline that included the best performing models per task; object type detection, category and attribute classification. The pipeline receives full-scale fashion images which are processed by the object type detector that predicts the garments' bounding boxes. The images are then cropped around their predicted bounding boxes and they are individually passed to the category and attribute classifiers. Three inference examples are presented in Figure 2.

[8] https://keras.io



(a) (b) (c)

Fig. 2: Inference examples from the *Mallzee dataset* with full-scale images, depicting multiple garments. Red bounding boxes are for upper-body, blue for lower-body, green for full-body garments and yellow for footwear. The images are not part of the training set.

5.2.2 Results on the DeepFashion dataset

We trained the three methods (baseline STL, STL w/ HLS & MTL w/ RNN+VA) on the DF1 dataset as a benchmark for category and attribute classification. We did not employ the ground truth bounding boxes offered by DF1 but rather passed its images through our trained object type detection model and only kept the predicted items that matched the images' category level. This decision ensures the generalisability of our method to other datasets that do not include bounding boxes or mask annotations, arguably, two costly and time-consuming types of manual annotation. The object type detector had an 89.2% retention rate meaning that 10.8% of the DF1 was not retained. After randomly sampling 1,000 samples from DF1 we assessed that 3.5% of the mismatch images were due to mistakenly annotated instances in the DF1 while 7.3% were due to mistaken predictions by our model. This translates into approximately 92.7% accuracy for the object type detection model which was not trained on DF1 data.

After cropping the images around the predicted bounding boxes, an EfficientNet-B4 architecture is fine-tuned for category and attribute classification on DF1. The results of each method can be seen in Table 7 compared with all relevant studies. All three models surpass the state-of-the-art on category classification. Specifically, "STL w/ HLS" has the highest top-3 accuracy with 93.99% (+0.98%) while "MTL w/ RNN+VA" has the highest top-5 accuracy with 97.57% (+0.56%).

On attribute classification, our models do not perform as well, being lower than most previously reported results. This could be an issue resulting directly from our proposed architecture. However, this seems unlikely given the fact that the same network outperforms the state-of-the-art on the category classification task and performs very well on the *Mallzee dataset*. A more plausible explanation is that we are using a slightly different evaluation metric. In the original DeepFashion paper,

Liu et al. (2016) used the "recall rate at top-k" metric and reported a 54.61% top-5 accuracy for the attribute classification task [2]. More recently Liu et al. (2020) in MMFashion, reported 14.79% top-5 recall with "VGG + Landmark Pooling" and 30.84% top-5 recall with a "RNN + Landmark Pooling" on attribute classification using the DeepFashion dataset [19]. The employed model and the dataset were the same as the original publication but the end results were vastly different. We could not verify what is the cause of this mismatch. However, it is possible that slightly different evaluation metrics were used. The official DeepFashion GitHub page[9] directs to the MMFashion page[10] whose evaluation protocol[11] uses an altered recall formulation, one that takes only the top-N (N=50) predictions into account while calculating the recall at k. Similarly, we could not verify which evaluation metric was used by all other research teams since their code was not publicly available. In Table 7 we report the conventional recall@k scores. However, we also calculate the aforementioned "altered recall rate" (shown in the parenthesis of Table 7) which results in 65.79% top-3 and 73.57% top-5 by *Baseline STL*, 66.19% top-3 and 73.73% top-5 by *STL w/ HLS*, and 53.01% top-3 and 66.4% top-5 by *MTL w/ RNN+VA*. Again we can observe STL w/ HLS improving upon the performance of Baseline STL for attribute classification. Additionally, if the "altered recall" is indeed the correct evaluation metric for attribute classification on DF1, *STL w/ HLS* has surpassed the current SoTA by 6.36% in terms top-3 recall rate.

Table 7: Benchmarking on DeepFashion for category and attribute classification. In parenthesis we report an "altered recall-rate@k" found in the DeepFashion/MMFashion GitHub page[11] used for the attribute classification task. (Bold denotes the best performance per metric)

Method	**Category**		**Attributes**	
	Top-3	Top-5	Top-3	Top-5
Chen et al., 2012 [20]	43.73	66.26	27.46	35.37
Huang et al., 2015 [21]	59.48	79.58	42.35	51.95
Liu et al., 2016 [2]	82.58	90.17	45.52	54.61
Corbiere et al., 2017 [7]	86.30	92.80	23.10	30.40
Wang et al., 2018 [8]	90.99	95.78	51.53	60.95
Ye et al., 2019 [9]	90.06	95.04	52.82	62.49
Li et al., 2019 [10]	93.01	97.01	59.83	**77.91**
Liu et al., 2020 [19]	-	-	-	30.84
Baseline STL	93.71	97.40	34.71 (65.79)	43.90 (73.57)
STL w/ HLS	**93.99**	97.49	36.20 (**66.19**)	45.62 (73.73)
MTL w/ RNN + VA	93.72	**97.57**	26.85 (53.01)	35.22 (66.4)

[9] https://liuziwei7.github.io/projects/DeepFashion.html

[10] https://github.com/open-mmlab/mmfashion/

[11] https://github.com/open-mmlab/mmfashion/blob/150f35454d94a0de7ae40dfdca7193207bd3fc57/mmfashion/core/evaluation/attr_predict_eval.py/#L100

Performing an internal comparison, we could see that *STL w/ HLS* improves upon *Baseline STL* on DF1 for both tasks. Especially for attribute classification there are noticeable improvements of 1.49% in top-3 and 1.72% in top-5 recall; using the conventional recall formulation. This is in accordance with our initial hypothesis that HLS can improve image pattern recognition in fashion by capturing existing hierarchical relationships between object types, categories and attributes; without requiring further domain expertise. On the other hand, *MTL w/ RNN+VA* performs very well only on the category-level task but modestly on the attribute-level task. We hypothesize that this disparity can be partly attributed to the fact that the two tasks - while being related to the same fashion images and domain - are relatively dissimilar. By their nature, the task of identifying categories may require the extraction of geometrical features and shapes while attributes may require more fine-grained features related to textures, fabrics and styles of the garments. Moreover, in MTL, the same convolutional backbone is being optimised for both multi-class classification with 50 categories and for multi-label classification with 1000 (fairly noisy) attributes. On top of that, the category task reached its peak performance at 10 training epochs and remained stationary while the attribute task required 40 epochs. To this point, a further investigation on alternative ways of calculating and combining the two loss functions (such as weighting) could be insightful for explaining or even mitigating the aforementioned issues.

Despite the limited performance of *MTL w/ RNN+VA*, a noteworthy advantage is the ability to plot the attention weights on top of an image and interpret the model's predictions. Four examples are shown in Figure 3. Figure 3a depicts a "checked shirt". During the category classification task the model mostly focuses around the shirt's buttons and the neckline area to conclude that the garment is a "shirt". On the other hand, while predicting the garment's attributes, the model broadens its focus and attends multiple points in order to determine that the garment is "checked". In Figure 3b, depicting a "frill dress", the model gives additional attention to the lower parts of the dress in order to identify its "frill" attribute. On the other hand, Figure 3c indicates a case where the model performs a correct prediction ("quilted puffer jacket") but the attention plot is not particularly meaningful or interpretable. During the category classification, the model attends more intensely on the shirt under the jacket and not the jacket itself; while correctly classifying it as a jacket. This is a case where the model's prediction is correct but the attention plot is not informative nor interpretable. Finally, Figure 3d depicts a 'checked shirt', and the model correctly predicts its category ('shirt'). However, the model ignores the hierarchical label sharing information at the attribute stage (that the garment category is a 'shirt') and instead predicts a 'polo'; an incompatible combination since the 'polo' attribute only applies to t-shirts. Had the model exhibited higher attention on the long sleeves of the garments it should have taken that into consideration. Furthermore, the model mistakes the 'checked' attribute pattern with 'paisley'; a vastly different type of visual pattern.

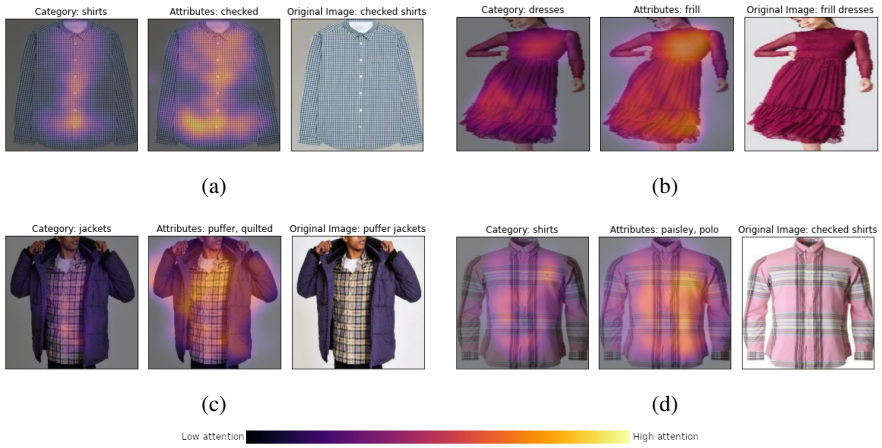


Fig. 3: Examples of attention plots and their predicted labels from the "MTL w/ RNN+VA" model on images that are not part of the training set. The ground truth labels are reported over the 'original images'.

6 Conclusions

In this study, we propose a deep learning pipeline that employs a hierarchical label sharing (HLS) technique. We examine the performance of HLS in two settings 1) single-task learning with hierarchical label sharing (STL w/ HLS) and 2) multi-task learning with RNN and visual attention (MTL w/ RNN+VA). Our hierarchical pipeline follows two-stages. It first performs object type detection, crops the images around the predicted bounding boxes and then applies category and fine-grained attribute classification. Evaluation on the DeepFashion benchmark shows that our method surpasses the current state-of-the-art (SotA) on category classification. Most notable, "STL w/ HLS" scored 93.99% top-3 accuracy while "MTL w/ RNN+VA" scored 97.57% top-5 accuracy.

When compared with previous studies, our approach offers the ability to work with full-scale fashion imagery depicting complete outfits. Also, it does not require landmark and mask annotations which are costly and time-consuming types of manual annotation. Furthermore, HLS can learn hierarchical fashion relationships between attributes/categories/types without requiring manually crafted rules by domain experts. Finally, by utilizing the *Mallzee dataset* the models was shown to cope with footwear, a very significant aspect of the fashion industry, that was completely missing from popular fashion dataset such as DeepFashion and DeepFashion2.

Regarding the further improvement of our models' performance, we consider the introduction of pre-trained human parts detection models in the object type detection phase, in order to improve the precise localisation of garments in relation to the human body [22]. Moreover, a hierarchical multi-label loss function could be

considered for explicitly discovering meaningful hierarchical relationships among categories and penalising non-possible multi-label combinations [23]. For future work, we plan on studying how the features extracted from fashion imagery can facilitate the improvement of trend forecasting and garment recommendations in the fashion domain.

Acknowledgements This work is partially funded by the project "eTryOn - virtual try-ons of garments enabling novel human fashion interactions" under grant agreement no. 951908. We would also like to thank Jamie Sutherland from Mallzee for his thoughts and input in this work.

References

1. Cheng, W., Song, S., Chen, C., Hidayati, S. & Liu, J. Fashion Meets Computer Vision: A Survey. *ACM Computing Surveys (CSUR)*. **54**, 1-41 (2021)
2. Liu, Z., Luo, P., Qiu, S., Wang, X. & Tang, X. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 1096-1104 (2016)
3. Ge, Y., Zhang, R., Wang, X., Tang, X. & Luo, P. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 5337-5345 (2019)
4. Ma, Y., Yang, X., Liao, L., Cao, Y. & Chua, T. Who, Where, and What to Wear? Extracting Fashion Knowledge from Social Media. *Proceedings Of The 27th ACM International Conference On Multimedia*. pp. 257-265 (2019)
5. Arslan, H., Sirts, K., Fishel, M. & Anbarjafari, G. Multimodal sequential fashion attribute prediction. *Information*. **10**, 308 (2019)
6. Sadegharmaki, S., Kastner, M. & Satoh, S. FashionGraph: Understanding fashion data using scene graph generation. *2020 25th International Conference On Pattern Recognition (ICPR)*. pp. 7923-7929 (2021)
7. Corbiere, C., Ben-Younes, H., Ramé, A. & Ollion, C. Leveraging weakly annotated data for fashion image retrieval and label prediction. *Proceedings Of The IEEE International Conference On Computer Vision Workshops*. pp. 2268-2274 (2017)
8. Wang, W., Xu, Y., Shen, J. & Zhu, S. Attentive fashion grammar network for fashion landmark detection and clothing category classification. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 4271-4280 (2018)
9. Ye, Y., Li, Y., Wu, B., Zhang, W., Duan, L. & Mei, T. Hard-Aware Fashion Attribute Classification. *ArXiv Preprint ArXiv:1907.10839*. (2019)
10. Li, P., Li, Y., Jiang, X. & Zhen, X. Two-stream multi-task network for fashion recognition. *2019 IEEE International Conference On Image Processing (ICIP)*. pp. 3038-3042 (2019)
11. Manikandan, N. & Ganesan, K. Deep Learning Based Automatic Video Annotation Tool for Self-Driving Car. *ArXiv Preprint ArXiv:1904.12618*. (2019)
12. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances In Neural Information Processing Systems*. **28** pp. 91-99 (2015)
13. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. & Tian, Q. Centernet: Keypoint triplets for object detection. *Proceedings Of The IEEE/CVF International Conference On Computer Vision*. pp. 6569-6578 (2019)
14. Tan, M., Pang, R. & Le, Q. Efficientdet: Scalable and efficient object detection. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 10781-10790 (2020)

15. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. & Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. *International Conference On Machine Learning*. pp. 2048-2057 (2015)
16. Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate. *ArXiv Preprint ArXiv:1409.0473*. (2014)
17. Xie, Q., Luong, M., Hovy, E. & Le, Q. Self-training with noisy student improves imagenet classification. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 10687-10698 (2020)
18. Charte, F., Rivera, A., Jesus, M. & Herrera, F. Concurrence among imbalanced labels and its influence on multilabel resampling algorithms. *International Conference On Hybrid Artificial Intelligence Systems*. pp. 110-121 (2014)
19. Liu, X., Li, J., Wang, J. & Liu, Z. MMFashion: An Open-Source Toolbox for Visual Fashion Analysis. *ArXiv Preprint ArXiv:2005.08847*. (2020)
20. Chen, H., Gallagher, A. & Girod, B. Describing clothing by semantic attributes. *European Conference On Computer Vision*. pp. 609-623 (2012)
21. Huang, J., Feris, R., Chen, Q. & Yan, S. Cross-domain image retrieval with a dual attribute-aware ranking network. *Proceedings Of The IEEE International Conference On Computer Vision*. pp. 1062-1070 (2015)
22. Gong, K., Liang, X., Li, Y., Chen, Y., Yang, M. & Lin, L. Instance-level human parsing via part grouping network. *Proceedings Of The European Conference On Computer Vision (ECCV)*. pp. 770-785 (2018)
23. Wehrmann, J., Cerri, R. & Barros, R. Hierarchical multi-label classification networks. *International Conference On Machine Learning*. pp. 5075-5084 (2018)