Multi-Scale Feature Based Fashion Attribute Extraction Using Multi-Task Learning for e-Commerce Applications

Viral Parekh and Karimulla Shaik
Flipkart Internet Private Limited, India

Keywords: Multi-Scale Features, Feature Pyramid Network, Multi-Task Learning, Visual Attribute Extraction.

Abstract: Visual attribute extraction of products from their images is an essential component for E-commerce applications like easy cataloging, catalog enrichment, visual search, etc. In general, the product attributes are the mixture of coarse-grained and fine-grained classes, also a mixture of small (for example neck type, sleeve length of top-wear), or large (for example pattern of print on apparel) regions of coverage on products which makes attribute extraction even more challenging. In spite of the challenges, it is important to extract the attributes with high accuracy and low latency. So we have modeled attribute extraction as a classification problem with multi-task learning where each attribute is a task. This paper proposes solutions to address above mentioned challenges through multi-scale feature extraction using Feature Pyramid Network (FPN) along with attention and feature fusion for multi-task setup. We have experimented incrementally with various ways of extracting multi-scale features. We use our in-house fashion category dataset and iMaterialist 2021 for visual attribute extraction to show the efficacy of our approaches. We observed, on average, $\sim 4\%$ improvement in F1 scores of different product attributes in both datasets compared to the baseline.

1 INTRODUCTION

E-commerce product catalog represents the products with set of images and its attributes. The quality of images, correctness and completeness of attributes plays a very important role in enriching buyer's experience through discovery. In general, seller provides the product information while listing the products. Images contain very significant information about the products, especially in fashion category, which can be leveraged to enrich product information without explicitly asking for it from sellers. The extraction of attributes also lifts other use cases such as image search, visual question answering etc. It is essential to extract product attributes at low latency to meet partner requirements and business performance. In addition, the number of attributes per product sub-category (example: t-shirt, top, dress, kurta, shirt, etc.) can range from 10 to 15 and number of classes per attribute can range from 10 to 80. Given the low latency, high number of product attributes and corresponding values, it is impractical to design or build attribute specific models. Instead, the attribute extraction as a multi task learning (MTL) with each attribute as a task, is optimal choice. But the image attribute extraction (IAE) is a challenging task due to intra-task and intra-class variance, task and class imbalances and presence of coarse-grained and fine-grained classes, mixture of small (for example neck type, sleeve length of top-wear), or large (for example pattern of print on apparel) regions of coverage on products. For prediction of the attributes that covers small regions on images, model is required to look at specific details, in other words the features at high spatial resolution, whereas for attributes that cover large regions, model is required to leverage abstract features.

Deep CNN models extract abstract features from deep layers whereas preliminary features from early layers. In other words, there is a trade-off between feature localisation and class discrimination power as we move from early to deep layers of the model. In this paper, we attempt to leverage the features at different scales in multi-task learning set up. Product attribute can have more than one value, for example a product can be suitable for wedding and party. So each task of IAE in MTL set up is modelled as a multi-label classifier.

In this work, we propose two modifications for the IAE network: 1) We add Feature Pyramid Network(Lin et al., 2017a) to the model architecture proposed in (Parekh et al., 2021). This results in having rich multi scale features representation at backbone.

Parekh, V. and Shaik, K.
Multi-Scale Feature Based Fashion Attribute Extraction Using Multi-Task Learning for e-Commerce Applications.
DOI: 10.5220/0011686700003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 644-651
ISBN: 978-989-758-634-7; ISSN: 2184-4321

2) Scale specific attention
Our main contributions are the following:

- We amended IAE MTL with the Feature Pyramid Network.
- Scale specific attention.
- We do extensive experimentation for these modification on the in-house dataset and present our analysis. For our experiments we see consistent improvement over the baseline for all the tasks.

The rest of this paper is organized as follows. In the next section, we give a brief review of the existing clothing style recognition algorithms. The proposed method is described in Section 4. In Section 5, we report experimental results on two different datasets: 1) In-house fashion category 2) iMaterialist-2021. Finally, we conclude this paper in Section 6.

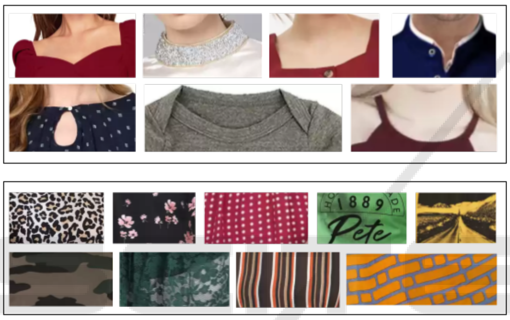


Figure 1: Few examples of two fine-grained attributes - *neck type* (top 2 rows) and *print type* (bottom 2 rows).

2 CHALLENGES AND MOTIVATION

In this section we have discussed various challenges related to our in house fashion category dataset.
1. Distribution: The dataset has heavy imbalance at every level - category, attribute and attribute values. At category level, the smallest one *cargo* has around 2700 data-points while the largest one *t-shirt* has around 2.3 lakhs data-points. At attribute level the *ideal for* is present for almost 100% of the data-points, while many attributes are present for only $1-2\%$ of the data-points. Similar imbalance is observed at attribute value level for all the attributes.
2. Missing Attributes: For a given product, few attributes are mandatory for the sellers to list, while many others are optional, e.g., color, pattern, product category are mandatory attributes, while pattern coverage, closure type are *good to have* attributes. Thus, the dataset has several missing values, especially the *good to have* attributes.
3. Label Distribution: One attribute might be applicable for multiple products, but the allowed attribute values might have different distribution for different products, e.g., *closure type* attribute has values like *button* and *zip* for jeans, while it has values like *drawstring* and *elastic* for pyjamas, while trousers have all the above values.
4. Fine-grained Attributes: Few of the attributes are extremely fine-grained with large number of attribute values. For example pattern, print type, neck type, top type have 89, 91, 50 and 49 attribute values respectively. Few examples are give in Figure 1.
5. Cross-product Interference: In most of the images, the main clothing item is worn by human model. So along with the primary product, other clothing items are also visible, e.g., in a trouser image some portion of the t-shirt or shirt is also visible. In some cases, accessories (e.g., dupatta, bags) occlude the main product. These cases are challenging when we want to make prediction for generic attributes like pattern, print type, occasion, etc.
6. Uniqueness: The dataset used in our work has some unique features and challenges compared to the existing data sets.

Indian Fashion Categories: As the data set is collated from an Indian e-commerce database, it contains several categories which are specific to India, e.g., we have categories like sari, kurta, kurti, etc.

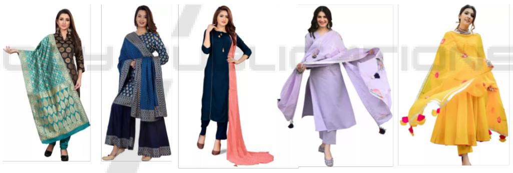


Figure 2: Different ways to wear dupatta.

Style of Wearing: Most of the western wears have standard way of wearing them. However there are many ways to drape a sari. In Figure 2, we observe different ways in which a dupatta (with ethnic-set) can be worn.

With the data challenges that are mentioned above, a feature representation at one scale is suboptimal for the extraction of all product attributes from its image. In other words, attributes that represent product sub-parts (e.g. neck shape, sleeve length etc.) require features at higher spatial scale than that of attributes representing large region of product. To elaborate, attributes like pattern, length etc. require low level features like edges, corners etc., while attribute like closure type requires features at higher spatial resolution along with semantic understanding. We leverage deep CNN feature representation at multiple scales.

In deep CNNs, input image goes through many convolution layers as shown in figure 3. In those convolution layers, the network learns new and increasingly complex features in its layers. The first convolution layer(s) learn features such as edges and simple textures. Later convolution layers learn features such as more complex textures and patterns. The last convolution layers learn features such as objects or parts of objects.

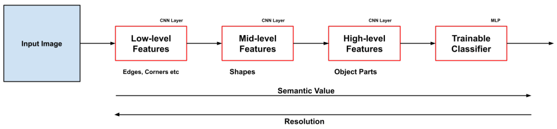


Figure 3: Different features recognised at different layers.

The semantic representations generated by a given CNN corresponding to an input image is the union of all feature maps generated by each convolution layer of the CNN, and not only the final feature map. Relying on several feature maps provide the networks with different spatial scales. The features maps at different scales can be used as common representation for all tasks, however Feature Pyramid Network (FPN) which is adopted for mutli-scale feature representation for object detection task has been proven for its accuracy. The main contribution of FPN is to enhance the semantic representation capability of shallower layer feature maps, using the semantic information encoded in deeper layer feature maps. The main weakness of feature maps generated by shallow layers is that they are not semantically as rich as the feature maps generated by deeper layers. It is because the process of semantic encoding of input images into feature maps is a hierarchical process where the basic semantics appear in the early layer feature maps, while the more complex semantics appear in the feature maps of deeper layers.

The attention mechanism in Neural Networks tends to mimic the cognitive attention possessed by human beings. The main aim of this function is to emphasize the important parts of the information, and try to de-emphasize the non-relevant parts. Since working memory in both humans and machines is limited, this process is key to not overwhelm a system's memory. In deep learning, attention can be interpreted as a vector of importance weights. When we predict an element, which could be a pixel in an image or a word in a sentence, we use the attention vector to infer how much is it related to the other elements.

Our hypothesis is that the combination of multi-scale features and scale specific attention is optimal for extraction of heterogeneous product attributes in MTL set up. We perform ablation study by altering the methods of computing multi-scale features, attention and prove the efficacy of the proposed method. The details of variants of the method are given in subsequent sections.

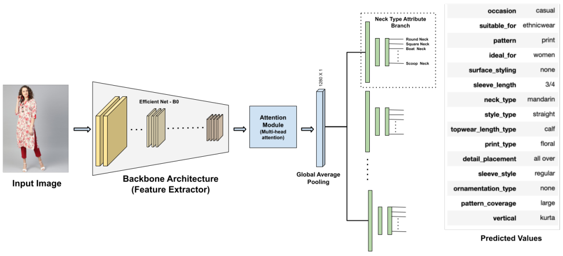


Figure 4: Baseline architecture proposed in (Parekh et al., 2021).

3 RELATED WORK

In the image classification task or image based multi-task learning setup, it is possible that different tasks or concepts require different spatial and semantic information in order to improve the performance. However traditional classification or multi task network uses features only from the last layer. In this paper we propose a architecture which leverage features from multiple scale to improve performance of multiple tasks.

Using features maps from multiple scales has been an important idea for object detection tasks. This helps in detecting object of different scales, aspect ratio and region of interest on various benchmarks(Lin et al., 2014; Everingham et al., 2015). The Single-Shot Detector(SSD)(Liu et al., 2016) has been one of the first networks which used features from different layers of the network to detect object of different scales. SSD used output from early convolution layers to detect smaller objects and output from later layer to detect larger objects. But SSD has some problems detecting small-scale objects because early convolution layers contain low-level information but less semantic information for tasks such as classification. FPN(Lin et al., 2017a) solve this problem by having both top-down and bottom-up pathways. Using this, reconstructed higher resolution feature map also has rich semantic information. FPN also have lateral connections between bottom-up and top-down feature maps to help the detector to predict the location better. There have been many extensions of original FPN e.g. BiFPN(Tan et al., 2020), NASFPN(Ghiasi et al., 2019), PANet(Liu et al., 2018), etc. but not much work has been done for using FPN for classification.

In (Baloian et al., 2021), authors have presented the evidence on how features from different scales can be useful to extract certain attributes like texture or color but it does not include the method to combine the learning from multi-scale features. In this work, we add FPN to classification network to classify different attributes with varying region of interests for visual attribute extraction task.

We use Image Attribute Extraction(IAE) problem to show efficacy of our approach. We use this problem because it is a challenging classification as different attributes require have different region of interests (ROI). We use In-house dataset for IAE that contains both coarse-grained (neck type, sleeve length) and fine-grained (neck type, sleeve style, pattern) attributes. There are many approaches that have been proposed for IAE(Ferreira et al., 2018). But we build on the work of (Parekh et al., 2021) which uses multi-task classification to solve this problem.

4 METHOD

In this section, we have discussed the baseline approach, our implementation details of adopting the baseline along with key components of the proposed approach.

4.1 Building Blocks

4.1.1 Model Details

In this section we have discussed the building blocks for the baseline network as well as proposed approach. The baseline network is borrowed from (Parekh et al., 2021), few changes are done in order to adapt to our experiment setup.

4.1.2 Backbone Network

The overall framework is shown in Figure 4. The input image is passed through the backbone network, which serves as the feature extraction unit. The output of backbone network is the base feature vector, which is the common input for all the branch networks, each dedicated to one attribute. EfficientNet (Tan and Le, 2019) is used as as the base feature extractor. In our experiments, we have chosen multi scale features from the EfficientNetV2-M, which gives $12 \times 12 \times 1280$-d feature vector as output. This is passed through an attention module followed by global average pooling to get the final 1280×1 feature vector, which is passed to the branch networks. We use a CBAM (Woo et al., 2018) as attention module.

4.1.3 Feature Pyramid Network

A Feature Pyramid Network (FPN), is a feature extractor that takes a single-scale image of an arbitrary size as input, and outputs proportionally sized feature maps at multiple levels, in a fully convolutional fashion. This process is independent of the backbone convolutional architectures. It therefore acts as a generic solution for building feature pyramids inside deep convolutional networks to be used in subsequent tasks.

We have used the architecture similar to (Lin et al., 2017a). In our experiments, the feature extractor module is EfficientNetV2-M(Tan and Le, 2019) . To build the FPN on top of our feature extractor, we have chosen final layer before the activation from each block of EfficientNetV2-M network. The layers chosen are as follows: *block3e_add, block5n_add, block7e_add*. The forward and backward pathways are created in the same manner as in the original paper but we have kept the number of channels at each feature layer consistent. The output of the FPN contains feature vectors with dimension $48 \times 48 \times 80$, $24 \times 24 \times 176$, and $12 \times 12 \times 512$. Finally we concatenate these feature vectors after applying global average pooling to get 768x1 dimensional feature vector.

4.1.4 Feature Fusion Module

Feature fusion, the combination of features from different layers or branches, is an important and frequently used module in deep learning architectures. It is often implemented via simple operations, such as summation or concatenation. In all over experiments we have used concatenation to fuse the features from different multi-scale features.

4.1.5 Branch Network

The base feature vector obtained from the Feature Fusion Module is passed through several branch networks. Each branch network caters to each attribute of interest and contains individual trainable classification network. Each branch consists of one fully connected layer followed by an output layer of size equal to the number of attribute values for that attribute.

Number of hidden layers and units in each hidden layers can be different for different branches as required by the complexity of the attributes.

4.2 Baseline

Baseline model is similar to (Parekh et al., 2021) as shown in Figure 4, and consists of Backbone network, Attention module and Branch network. This model

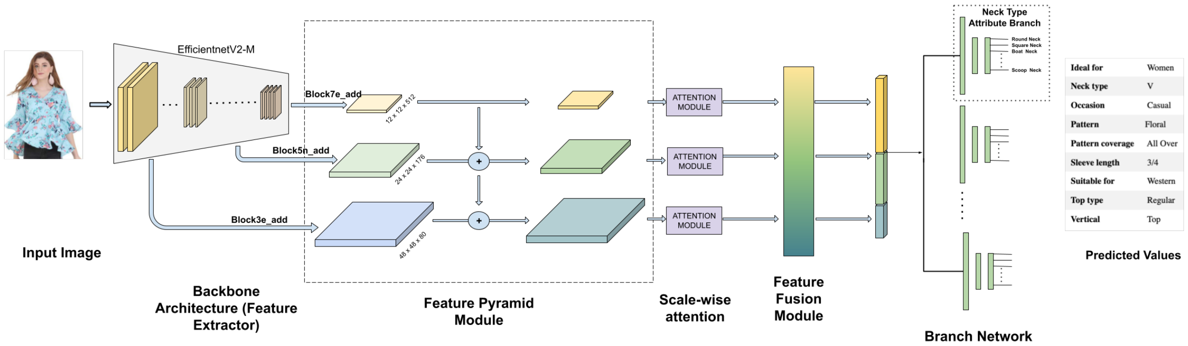


Figure 5: Proposed architecture with Multi-scale features, Feature Pyramid Network, scale-wise attention module and Feature Fusion Module.

Table 1: Performance comparison between baseline and FPN architecture with in house dataset (Micro F1 Scores at attribute level).

Attribute Granualarity	**coarse**							**fine**		
Attribute ROI	**high**			**low**				**high**	**low**	
attribute	**category**	**bottomwear length**	**topwear length**	**closure**	**sleeve length**	**distressed**	**rise**	**pattern type**	**sleeve style**	**neck type**
Baseline	0.927	0.807	0.524	0.488	0.798	0.749	0.746	0.592	0.690	0.548
Approach-1	0.920	0.803	0.512	0.492	0.788	0.728	0.743	0.592	0.684	0.526
Approach-2	0.928	0.827	0.540	0.515	0.805	0.755	0.752	0.609	0.705	0.556
Approach-3	**0.942**	0.825	0.557	0.509	**0.829**	0.773	0.758	0.608	**0.721**	**0.596**
Approach-4	0.939	**0.830**	**0.564**	**0.530**	0.828	**0.776**	**0.762**	**0.616**	0.718	0.584

does not use multi-scale features hence only single attention module is applied to final feature layer of the backbone network as discussed in 4.1.2. More implementation details are provided in section 5.2.

4.3 Proposed Approach

As depicted in Figure 5, our proposed approach consists of backbone network, feature pyramid module, scale-wise attention, feature fusion module and branch network. We have discussed incremental modification over the baseline model with details in 5.2.1 to 5.2.5.

5 EXPERIMENTS AND RESULTS

5.1 Datasets

Public Dataset: The iMaterialist-2021 (FGVC8, 2021) is a multi-label attribute recognition database with 228 attribute labels for each example. We choose the 8 global attributes ('length', 'neckline type', 'opening type', 'silhouette', 'textile pattern', 'waistline') to show the efficacy of our approach. The total 45589 images were split into train and validation set of images. the Test set contains 1158 images.

Internal Dataset: The dataset used in this work is obtained from in-house catalog, and consists of 8 product categories uploaded by the sellers in the last 2 years. After manual inspection of all the possible attributes of these categories, we have identified 10 visual attributes which are applicable to one or more products.

Table 2: Different types of fashion product attributes.

	Coarse grained	Fine grained
Large ROI	Vertical, Topwear length, Bottomwear length	Pattern
Small ROI	Sleeve length, Distressed, Rise, Closure	Neck type, Sleeve style

The visual attributes are a mix of fine-grained (e.g., print type, print coverage, neck type) and coarse-grained attributes (e.g., pattern, top wear length, product). Overall, our train and test split consists of 234257 and 70765 images respectively. the train set was further divided to train and validation set during experiments.

Table 3: Performance comparison between baseline and FPN architecture on iMaterialist 2021 dataset (Micro F1 Scores at attribute level).

Attributes	**textile pattern**	**opening type**	**silhouette**	**length**	**waistline**	**neckline type**
Baseline	0.684	0.680	0.604	0.554	0.529	0.169
approach-1	0.679	0.681	0.639	0.580	0.565	0.198
approach-2	0.659	**0.709**	0.599	0.548	0.522	0.124
approach-3	0.699	0.700	0.629	0.583	**0.558**	0.197
approach-4	**0.708**	0.661	**0.642**	**0.635**	0.549	**0.226**

5.2 Experiments

To examine the effectiveness of various components, we performed the following set of experiments and compared the performance on internal dataset as well as iMaterialist 2021 dataset.

Training details
Here, we have used EfficientNetV2-M pre-trained with Imagenet dataset as the backbone network. In the branch network, there are total 10 classification branches, out of which 9 are for the attributes, and one for product category classification for internal dataset. Each branch contains one hidden layers followed by the output layer. The number of nodes in the output layer is the same as the number of allowed attribute values for that attribute, e.g., *closure type* attribute has 4 attributes values (‘Button’, ‘Zipper’, ‘Drawstring’, ‘Elastic’), thus there will be 4 nodes in its output layer. The input images are resized to 380×380 by maintaining aspect ratio.

We use image augmentations like zoom in and out within range of +/- 20%, translation within +/- 20% both in horizontal and vertical directions, rotation within +/- 10 degrees, shear in range +/-10% and flipping of images from left to right with 50% probability. The model is trained with Adam optimizer, learning rate of $1e-5$ with decay rate of 0.75 per 8 epochs with batch size of 128.

The loss function for this multi-label classification setup with support to handle the missing label is same as (Parekh et al., 2021), where we ignore the loss of the missing attributes.

5.2.1 Baseline

In this experiment, we trained the model similar to (Parekh et al., 2021). Here we used pretrained Efficientnet-V2M model as a backbone, followed by CBAM attention and the branch network as mentioned in 4.1.1.

5.2.2 Approach-1: Multi-Scale Features Without Using FPN, Without Attention

To compare if the features from multiple scale help for better prediction in multi task setting, we trained a model by using features from 3 different layers along with separate attention module for each layer followed by branch network.

5.2.3 Approach-2: Multi-Scale Features Using FPN, Without Attention

In this experiment we use multi-scale features along with the FPN module. In this experiment we do not apply any attention so that we can measure the effectiveness of the Multi-scale features and FPN compared to baseline.

5.2.4 Approach-3: Multi-Scale Features Using FPN, with Attention

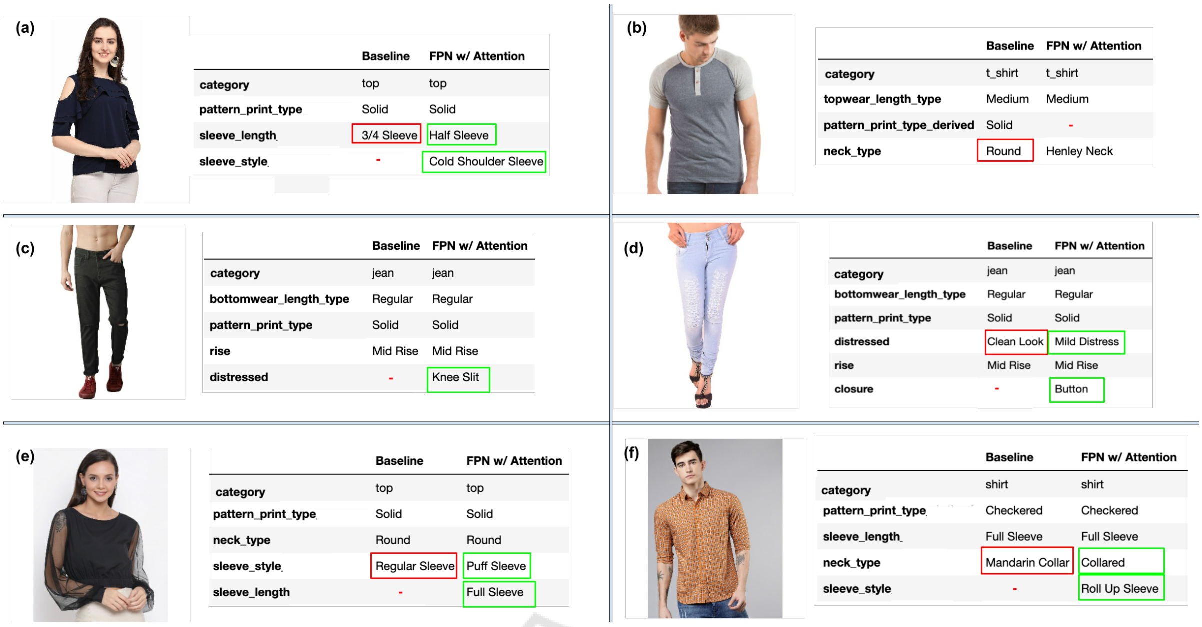
Now In this experiment, We use multi-scale features along with the FPN module followed by scale-wise separate attention module to check if attention module is useful in multi-scale features or not.

5.2.5 Approach-4: Multi-Scale Features Using RetinaNet FPN, with Attention

We performed this experiment to directly reuse the FPN module proposed in (Lin et al., 2017b). Here the difference compared to Experiment-4 is that, in Exp-4, the number of channels for each scale remains unchanged, while in this experiment output for all the scale have the same 256 number of channels. So after the feature fusion, instead of 768×1 dimensional vector as mentioned in 4.1.3, we get 1280×1 dimensional vector.

5.3 Results

We have compared the performance of all the approaches mentioned in the experiments section on in house dataset and iMaterialist 2021 dataset. In Table 1, the attribute wise micro F1 scores for in house



(a)

	Baseline	FPN w/ Attention
category	top	top
pattern_print_type	Solid	Solid
sleeve_length	3/4 Sleeve	Half Sleeve
sleeve_style	-	Cold Shoulder Sleeve

(b)

	Baseline	FPN w/ Attention
category	t_shirt	t_shirt
topwear_length_type	Medium	Medium
pattern_print_type_derived	Solid	-
neck_type	Round	Henley Neck

(c)

	Baseline	FPN w/ Attention
category	jean	jean
bottomwear_length_type	Regular	Regular
pattern_print_type	Solid	Solid
rise	Mid Rise	Mid Rise
distressed	-	Knee Slit

(d)

	Baseline	FPN w/ Attention
category	jean	jean
bottomwear_length_type	Regular	Regular
pattern_print_type	Solid	Solid
distressed	Clean Look	Mild Distress
rise	Mid Rise	Mid Rise
closure	-	Button

(e)

	Baseline	FPN w/ Attention
category	top	top
pattern_print_type	Solid	Solid
neck_type	Round	Round
sleeve_style	Regular Sleeve	Puff Sleeve
sleeve_length	-	Full Sleeve

(f)

	Baseline	FPN w/ Attention
category	shirt	shirt
pattern_print_type	Checkered	Checkered
sleeve_length	Full Sleeve	Full Sleeve
neck_type	Mandarin Collar	Collared
sleeve_style	-	Roll Up Sleeve

Figure 6: Qualitative Results: Red and Green boxes indicates incorrect and correct predictions respectively. No boxes are marked if the predicted attributes values are correct for both the models. –indicates, no values crossed prediction threshold.

dataset are presented. The proposed Approach-3 and 4 both performs better compared to baseline by 4-5%. Here The low ROI attributes have better gain compared to high ROI attributes. Similarly fine grained attribute have better gain compared to coarse grained attribute in FPN model. We see the similar results on iMaterialist 2021 dataset as mentioned in Table 3.

In Figure 6 we have demonstrated a few qualitative examples for subjective analysis. The attribute predictions (threshold 0.3) for baseline and proposed approach-3 are provided. In Figure 6c and 6d, the low ROI attributes like distressed (Knee Slit) and Closure (Button) are not predicted by baseline model (with required confidence score) but the proposed model is able to predict these values with high confidence. Similarly, Figure 6a and 6b the half sleeve is wrongly predicted as 3/4 sleeve and henley neck is predicted as round neck by baseline model, but the proposed model is able to predict these attribute correctly which supports over hypothesis.

6 CONCLUSION AND FUTURE WORK

In this paper, we introduce multi-scale feature based fashion attribute extraction using multi-task learning. We perform ablation study with variants of multi-scale feature extraction and addition of attention on in-house fashion dataset and iMaterialist-2021. Product attributes in fashion are relatively more heterogeneous in nature compared to categories, so we chose the category for our experiments. We demonstrate that our proposed methods outperform baseline methods. We conclude that when tasks of MTL are heterogeneous in nature, multi scale feature extractor along with scale specific attention is preferred approach. To strengthen our hypothesis, we are repeating the experiments with other backbones and subjective quality analysis with feature visualisation.

REFERENCES

Baloian, A., Murrugarra-Llerena, N., and Saavedra, J. M. (2021). Scalable visual attribute extraction through hidden layers of a residual convnet. *CoRR*, abs/2104.00161.

Everingham, M., Eslami, S., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136.

Ferreira, B. Q., Baía, L., Faria, J., and Sousa, R. G. (2018). A unified model with structured output for fashion images classification. *arXiv preprint arXiv:1806.09445*.

FGVC8 (2021). imaterialist 2021 dataset.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Parekh, V., Shaik, K., Biswas, S., and Chelliah, M. (2021). Fine-grained visual attribute extraction from fashion wear. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3973–3977.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Tan, M., Pang, R., and Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790.

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, Cham. Springer International Publishing.