Clothing Parsing using Extended U-Net

Gabriela Vozáriková[a], Richard Staňa[b] and Gabriel Semanišin[c]
Institute of Computer Science, Pavol Jozef Šafárik University in Košice, Jesenná 5, Košice, Slovakia

Keywords: U-Net, Clothing Parsing, Segmentation, Computer Vision, Multitask Learning, Deep Learning, Fully-convolutional Network.

Abstract: This paper focuses on the task of clothing parsing, which is a special case of the more general object segmentation task well known in the field of computer vision. Each pixel is to be assigned to one of the clothing categories or background. Due to complexity of the problem and lack of data (until recently) performance of the modern state-of-the-art clothing parsing models expressed in terms of mean Intersection over Union metric (IoU) does not exceed 55%. In this paper, we propose a novel multitask network by extending fully-convolutional neural network U-Net with two side branches – one solves a multilabel classification task and the other predicts bounding boxes of clothing instances. We trained this network using a large-scaled iMaterialist dataset (Visipedia, 2019), which we refined. Compared to well performing segmentation architectures FPN, DeepLabV3, DeepLabV3+ and plain U-Net, our model achieves the best experimental results.

1 INTRODUCTION

Recently, the fashion industry has been undergoing a digital transformation by expanding into online platforms. Automated processing and analysis of fashion images have been gaining much attention. Clothing parsing in itself is an important tool in fashion image analysis (e.g. for automatic colour tagging of clothing items), but also serves as an important intermediate step in many other tasks. For example in (Aoki et al., 2019) a fashion segmentation model is used to boost the performance of a fashion style estimation model. Segmentation models are also often utilized in Deep Generative models, as is the case for clothing parsing in the image synthesis model proposed in (Xian et al., 2018). Another example outside the fashion domain is described in (Simon et al., 2020), in which a clothing parsing model is used as one of the four steps of filtering candidates for person identification in surveillance videos based on their outfit.

Although being part of the more general segmentation task, clothing parsing poses its specific challenges. They consist mainly in ambiguous and numerous clothing categories (sometimes difficult to differentiate between even for human annotators), occlusion, deformities, high intra-class variation. These obstacles, in combination with very limited amount of data (until recently), result in the state-of-the-art segmentation models' performance not exceeding 55% mean Intersection over Union metric. (Martinsson and Mogren, 2019) reports state-of-the-art performance on two benchmark datasets for clothing parsing - Refined Fashionista with mean IoU of 51.78% and CFPD with mean IoU of 54.65%. (Zheng et al., 2018) provides a description of a recently released large-scale fashion dataset and reports IoU between 28% and 68% separately for 13 clothing categories for DeepLabV3+ architecture.

In this paper, we approach the clothing parsing problem using an extended and modified version of U-Net architecture trained on the large-scaled Kaggle iMaterialist (Fashion) dataset (Visipedia, 2019). U-Net (Ronneberger et al., 2015) is a fully-convolutional network developed for biomedical image segmentation, but it proved to be successful in many other domains. Deep layers of the U-Net network extract semantically rich features, while spatial information is preserved by cross connections. We used a modified version of U-Net by replacing backbone with Resnet34 pre-trained on ImageNet and extended it by two side branches solving object detection and multilabel classification task. The evaluation shows that these additional branches contribute to the increased performance in the clothing parsing segmentation task.

[a] https://orcid.org/0000-0002-0111-972X
[b] https://orcid.org/0000-0001-7938-2117
[c] https://orcid.org/0000-0002-5837-2566

Vozáriková, G., Staňa, R. and Semanišin, G.
Clothing Parsing using Extended U-Net.
DOI: 10.5220/0010177700150024
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 15-24
ISBN: 978-989-758-488-6

Our contribution lies namely in:

- We propose a simplified version of the clothing parsing task that we argue is still complex enough to suffice many real-life applications.
- We propose an extension to the standard U-Net architecture by providing additional two side branches in order to increase the capacity of the model to capture global dependencies.
- We perform a refinement of the large-scaled iMaterialist dataset (Visipedia, 2019), which results in a significant performance boost of the proposed segmentation model.
- We provide results of a selection of modern segmentation models, namely Feature Pyramid Networks (FPN), DeepLabV3, DeepLabV3+ and U-Net trained on the refined iMaterialist dataset.

This paper is organized into five sections. Section 2 reviews prior work in semantic segmentation of clothing items. Section 3 describes the proposed network architecture and training specifics. Section 4 focuses on the refined version of iMaterialist dataset, discusses the model performance and identifies its bottleneck. Section 5 closes with a summary and conclusion.

2 RELATED WORK

Clothing parsing, also known as semantic segmentation of clothing items, is an important tool in fashion image analysis. One of the first approaches to clothing parsing and creation of Fashionista dataset (685 images) for benchmarking purposes is described in (Yamaguchi et al., 2012). This approach was highly reliant on the performance of the pose estimation model. (Liu et al., 2013) used additional color-category meta tags provided by users and introduced CFPD dataset with 2 682 annotated images.

More recent work has used powerful deep convolutional neural networks achieving state-of-the-art results without the need for additional meta tags. Tangseng et al. (Tangseng et al., 2017) augment fully-convolutional networks (FCNs) by a branch that predicts combinatorial preference of garments. In (Khurana et al., 2018) texture cues extracted by Gabor filters are used for clothing type classification boosting. In (Martinsson and Mogren, 2019) architecture based on feature pyramid networks with a ResNeXt backbone was used for the clothing parsing task.

All of these models were trained and evaluated on rather small-size datasets (aforementioned Fashionista and CFPD). Recently, in order to facilitate the training of better performing models, three big-scale datasets were released - namely ModaNet (Zheng et al., 2018), DeepFashion2 (Ge et al., 2019) and a dataset (Visipedia, 2019) released as part of the Kaggle iMaterialist (Fashion) Challenge 2019 at FGVC6 (iMaterialist). ModaNet and DeepFashion2 papers also provide an overview of the performance of a selection of modern segmentation models trained on the corresponding datasets. To our knowledge, no research was conducted using the iMaterialist dataset.

3 ARCHITECTURE OF THE PROPOSED NETWORK

We base our neural network on the modified U-Net architecture with the input image dimension of 288x192x3 (Ronneberger et al., 2015), which we enrich by two side branches - bounding box and multilabel classification branch. Figure 1 illustrates the architecture of our network. Our compound loss function consists of 3 parts - object detection, multilabel and segmentation loss term.

3.1 U-Net

U-Net architecture is a special type of fully-convolutional neural networks. It has a typical encoder-decoder structure with the aim of dense pixel-wise predictions generation. We have made a couple of modifications to the original U-Net architecture, namely:

1. Taking our hardware constraints into account, we decided to use ResNet34 pre-trained on ImageNet as the encoder.
2. Simple nearest neighbor technique was used as an upsampling layer. We also experimented with trainable upsampling layers, but we did not acquire better results.

3.2 Additional Branches

Our U-Net branch extensions were deeply inspired by the clothing parsing architecture described in (Tangseng et al., 2017). They used VGG-16 based fully convolutional network FCN 8s extended by a side path that encoded and predicted combinatorial preference of garment items. In simpler terms, the added side path solves multilabel classification task - predicts what clothing categories are present in the input image. This side branch proved useful as was demonstrated by increased Intersection over Union metric. Inspired by this paper, we extended U-Net

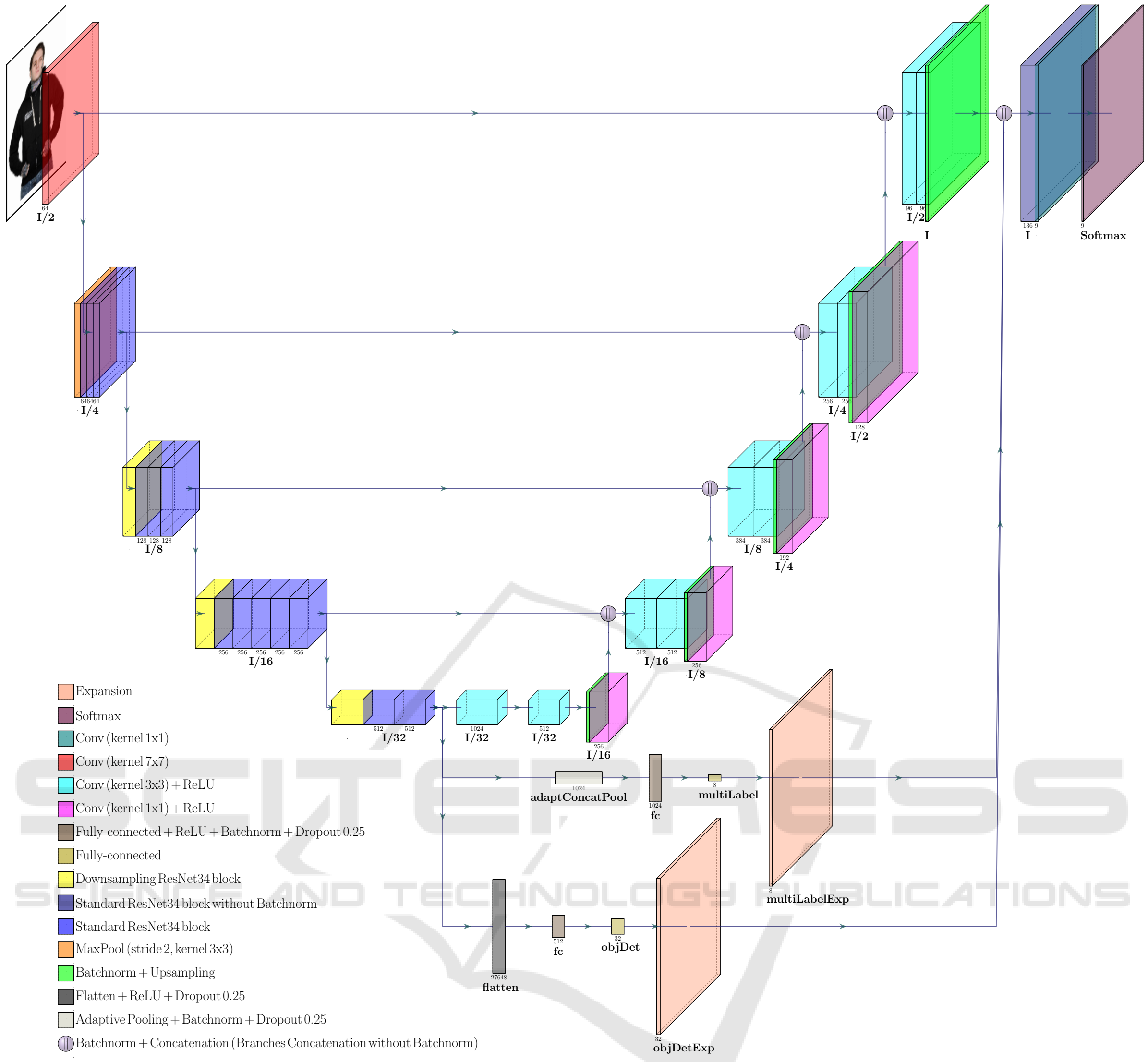

Figure 1: Visualization of Our model. Standard and downsampling ResNet34 blocks are described in ResNet paper (He et al., 2016). ‘Adaptive Pooling’ stands for the concatenation of Adaptive Max Pooling and Adaptive Average Pooling layer. Basic nearest neighbor upsamling layer was used. ‘multiLabel’ stands for the ‘ClassBranch’ logit output, additional sigmoid function is used before the multilabel classification binary cross-entropy loss calculation. ‘objDet’ stands for the ‘BBbranch’ logit output, additional sigmoid function is used before the object detection smooth L1 loss calculation. ‘Expansion’ is described in subsection 3.2. Please, zoom in if content not visible.

not only by the aforementioned multilabel classification branch (’ClassBranch’), but also by object detection/localization bounding box branch (‘BBbranch’) - the goal of this branch is to predict bounding boxes of clothing instances (maximum of 1 instance per clothing category). In other words, ‘BBbranch’ focuses on where the clothing items are on the more global image (not pixel) level. On the contrary, ’ClassBranch’ is location agnostic. Insertion of the branches is illustrated in Figure 1.

’ClassBranch’ is implemented as a subnetwork consisting of an adaptive average and max pooling layer followed by two fully connected layers with dimensions 512 and number of clothing categories. The first fully connected layer is followed by ReLU, Batchnorm and dropout layer. ‘BBbranch’ is implemented as two fully connected layers without any preceding pooling layer. We want to emphasize that ‘BBbranch’ is not solving the object detection task in its general form, which has no prior knowledge about the number of objects in the input image. On the contrary, we use the fact that there is a maximum of one instance of each clothing category in the input image. Each output channel of ‘BBbranch’ is dedicated to bounding box prediction of a specific clothing category, and no complex region proposal subnetwork is needed. It is important to note that no additional annotation work was done, the ground truth bounding box position calculation was inferred automatically from the ground truth segmentation mask.

Output layers of these additional branches are injected back into the segmentation network by concatenation. More specifically, logit outputs of the branches are expanded so that they match the resolution of the input image (e.g. for the ’ClassBranch’ eight logit output neurons are transformed into eight layers of resolution 288x192. Each layer contains 288x192 clones of the corresponding output neuron). This way of branches injection enables gradient flow from the main segmentation loss to the branches. The concatenation is followed by a residual convolutional layer and final convolutional layer outputting segmentation mask logit prediction. We also experimented with the setting, in which no inclusion of the additional branches into the main segmentation part was performed.

3.3 Loss Function

Loss function used in our network is a weighted sum of three terms:

First is a weighted cross-entropy loss (WCE) pertaining to the main pixel-wise segmentation task. Because of the data imbalance (dominance of the background class), we have set the weight of WCE to 1 for the background class and 2 for every of the foreground classes. This loss term was used with a weight of 1.

Second loss term is binary cross-entropy pertaining to the multilabel classification task of the ‘ClassBranch’. This loss term was used with a weight of 0.5.

Third component is the smooth L1 loss described in (Girshick, 2015). According to the paper, this loss is less sensitive to outliers. The ground truth bounding box position was expressed relative to the input image size (resulting in values in [0,1] range). Therefore, additional sigmoid function was introduced before the smooth L1 loss calculation. Not all clothing categories were present in the input image. Therefore, only loss terms from the channels corresponding to the clothing categories present in the input image contributed to the final loss term. This loss term was used with a weight of 75.

Weights of the particular loss terms were determined empirically, so that each loss term would contribute approximately evenly to the final compound loss (unweighted loss terms have different value ranges).

3.4 Network Training Specifics

For reproducibility, this subsection provides our network training details.

Preprocessing - based on the most prevalent height to width ratio in our dataset (3:2) and the fact that Resnet34 backbone downsizes images by 32, we resize every image to 288x192.

Data augmentation - rather conservative data augmentation was performed, namely horizontal flip, mild changes in lighting and rotation. It is important to note, that segmentation mask was transformed in the same way as the original input image, and the ground truth bounding box was inferred from the transformed mask (calculation performed more efficiently on the GPU).

We used a batch size of 16. Because of GPU memory constraints while still aiming for a batch of sufficient size, regarding the weights and activations the whole training was performed using half-precision (floating point 16), but loss terms and gradients were 32 floating-point precision.

We used GPU NVidia GTX 1080. The training was two-phased, which is characteristic of transfer learning. In the first phase, we froze the Resnet34 backbone pre-trained on Imagenet and trained only the rest of the network (all the branches included). This phase was ten epochs long, using Adam optimizer and cyclical learning rate scheduler (Smith, 2017) with learning rate maximum of 10^{-4}. Then we unfroze the whole network, continued for another ten epochs, Adam optimizer and cyclical learning rate scheduler with learning rate maximum of 10^{-4}. Weight decay of 10^{-3} was used in both phases (weight decay decoupled from the Adam optimizer was used as described in (Loshchilov and Hutter, 2017)). This training procedure was determined empirically based on the training behavior of the Plain U-Net. Training of all of the models was closely monitored in terms of losses and metrics. There was no indication of performance suffering from poorly chosen training hyperparameters, hence no specific training hyperparameters modifications for particular models were performed.

The implementation of the network design and training was done in FastAI (Howard et al., 2018) and PyTorch (Paszke et al., 2017) environment. (Yakubovskiy, 2020) implementation of DeepLabV3, DeepLabV3+ and FPN architectures was used.

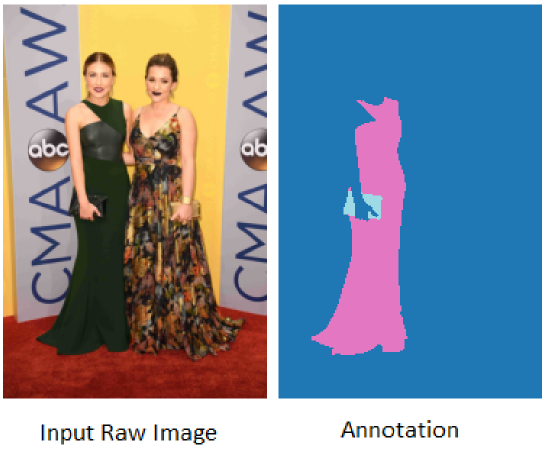

Figure 2: Example of an annotation issue in iMaterialist dataset (Visipedia, 2019). In images with more than one person present, only one of the people was segmented.

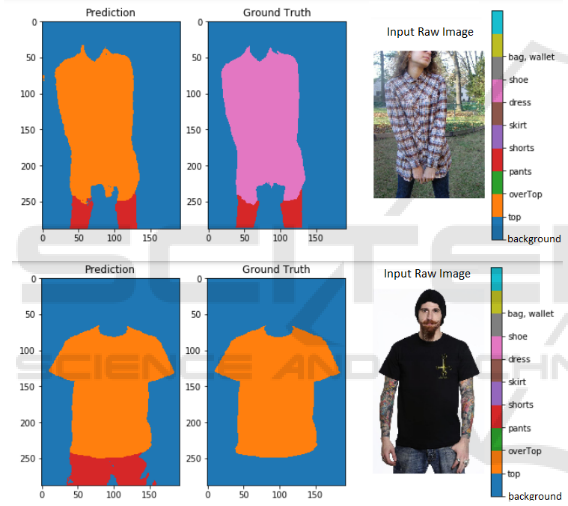

Figure 3: Example of annotation inconsistency in the iMaterialist dataset. In the case of images with a partial view, sometimes the cut off lower body garment was annotated as ‘pants’ and sometimes it was annotated as ‘background’. ‘Ground Truth’ stands for the annotation, ‘Prediction’ is the output of our segmentation model.

4 DATASET AND RESULTS

This section provides a description of the dataset used and model results.

4.1 Dataset

As discussed in the introduction section, the majority of the previous clothing parsing research was performed on datasets of small size.

We use a large-scale dataset (Visipedia, 2019) released as part of the Kaggle iMaterialist (Fashion) Challenge 2019 at FGVC6. It contains the total of 45 622 clothing images (from which approx. 3 500 depict single clothing item, while the rest depict complex clothing outfits as worn by people) from daily-life, celebrity events, and online shopping with diverse poses (not limited to the full-frontal view), occlusion types, scales, viewpoints, clothes layering. The dataset recognizes 27 main clothing categories.

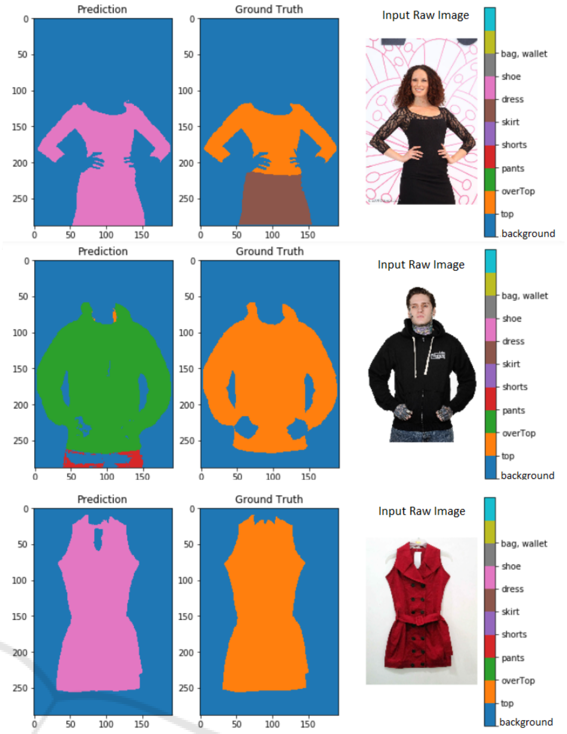

Figure 4: Demonstration of challenging nature of the clothing parsing task. Some very common clothing combinations are easily confused even for human annotators, e.g. dress vs top+skirt combination as depicted in the first row. Furthermore, assignment to a clothing category might be fuzzy for some items, especially when depicted solo without any referential human models. The third row provides an example of such an item - without additional info about the length of the item, its assignment to clothing category is rather ambiguous. ‘Ground Truth’ stands for the annotation, ‘Prediction’ is the output of our segmentation model.

We made a couple of modifications to the dataset. Upon inspection rather ambiguous (even for a human) and numerous clothing categories we reduced the number of clothing categories by merging or excluding some of them. Final clothing categories and their relation to the original dataset:
‘top’: merging of ‘shirt, blouse’ and ‘top, t-shirt, sweatshirt’
‘overtop’: merging of ‘sweater’, ‘cardigan’, ‘jacket’, ‘vest’ and ‘coat’
‘pants’: ‘pants’

Table 1: Performance of the plain U-Net model trained on the base vs refined dataset. ‘IoU excl. bg’ stands for IoU excluding the background category, ‘IoU overall’ means IoU including background, ‘IoU Mask’ is IoU of special segmentation with only two possible classes - foreground and background.

Dataset	IoU excl. bg	IoU overall	Iou Mask
Base	70.01	89.44	91.38
Refined	**72.10**	90.23	92.03

Dataset	Bg	Top	Overtop	Pants	Shorts	Skirt	Dress	Shoe	Bag
Base	97.02	69.22	69.43	76.61	59.31	47.54	75.20	62.37	40.10
Refined	97.24	71.02	71.72	78.25	61.60	51.12	77.21	65.05	46.04

Table 2: Performance expressed in terms of IoU. ‘IoU excl. bg’ stands for IoU excluding background, ‘IoU overall’ means IoU including background, ‘IoU Mask’ is IoU of special segmentation with only two possible classes - foreground and background. The best performance of each category is bold.

Method	IoU excl. bg	IoU overall	Iou Mask
FPN	67.77	88.18	89.44
DeepLabV3	68.91	88.33	88.80
DeepLabV3+	70.85	89.31	90.15
Plain U-Net	72.10	90.23	92.03
Our model	**73.38**	**90.66**	**92.18**
ClassBranch only	73.01	90.57	92.16
Branches	72.72	90.44	92.13

Method	Bg	Top	Overtop	Pants	Shorts	Skirt	Dress	Shoe	Bag
FPN	96.29	66.74	66.56	74.35	55.99	47.91	73.68	56.31	33.69
DeepLabV3	96.03	68.05	68.81	74.28	59.41	49.26	74.18	53.80	39.64
DeepLabV3+	96.54	70.14	70.05	76.33	60.93	50.88	76.45	58.18	40.36
Plain U-Net	97.24	71.02	71.72	78.25	61.60	51.12	77.21	65.05	46.04
Our model	97.29	**72.49**	72.68	**79.81**	**65.90**	**53.83**	**77.97**	**65.97**	**48.31**
ClassBranch only	**97.30**	72.23	**72.70**	78.95	63.83	51.84	77.61	65.70	46.47
Branches	97.27	71.46	72.24	79.08	63.56	53.39	77.42	65.60	48.09

‘shorts’: ‘shorts’
‘skirt’: ‘skirt’
‘dress’: ‘dress’
‘shoe’: ‘shoe’
‘bag, wallet’: ‘bag, wallet’.

Additionally, we excluded the ‘jumpsuit’ category with approximately 900 images. We argue that this reduced version still suffices many real-life applications (automatic colour extraction in the fashion domain, suspicious person detection in surveillance videos, etc.) while significantly lowering the complexity of the clothing parsing task.

There was one major dataset annotation issue we had to address. In images with more than one person present, only one of the people was segmented. Figure 2 shows an example of such an annotation. We could not even identify the selection criterion for choosing the person to be segmented. It proved to be malicious for the model training, as when training on these images, we were by means of the loss function penalizing the model for correctly identifying clothing items of all the people in the input image.

To address this issue, firstly we applied Mask R-CNN (Massa and Girshick, 2018) object detection model to select out images with more than one person (in a total of 12 188 images more than one person was detected). Then we performed an iterative image cropping with respect to the segmentation ground truth mask (GT mask) - we cropped the rectangle containing the GT mask keeping the original height of the image and adding extra x% of the GT mask width to the left and right of the rectangle (iteratively $x = 30\%, 10\%, 0\%$). After each iteration, cropped images with only one detected person were put aside, the rest proceeded to the next iteration. Finally, cropped images after the most restrictive type of cropping (with $x = 0\%$) with more than one detected person were manually inspected (approx. 1 030 images). The final cleaned dataset contained 43 470 images. The effect of this dataset cleaning is shown in Table 1. It contributed to more than 2% increase in IoU excluding background metric.

During visual inspection of the results of our model, we identified another annotation inconsistency. In the case of images with a partial view, sometimes the cut off lower body garment was annotated

Table 3: Confusion matrix (in %) for predictions of our model on the test dataset. The first matrix expresses row percentages, i.e. looking at the particular clothing category ground truth pixels what is the distribution of the corresponding predicted labels. The second expresses column percentages, i.e. looking at the particular clothing category predicted pixels what is the distribution of the corresponding ground truth labels.

		Predicted								
		bg	top	overTop	pants	shorts	skirt	dress	shoe	bag
Ground Truth	bg	98.08	0.30	0.38	0.36	0.03	0.08	0.52	0.18	0.09
	top	1.88	83.56	8.59	0.33	0.09	0.19	5.26	0.00	0.11
	overTop	1.96	7.03	86.53	0.48	0.04	0.26	3.34	0.00	0.37
	pants	2.62	0.79	0.92	92.14	0.37	1.23	1.20	0.56	0.16
	shorts	1.61	2.13	0.57	4.38	79.78	7.21	4.18	0.01	0.14
	skirt	1.80	0.83	0.97	3.21	2.99	65.08	24.72	0.05	0.34
	dress	1.98	3.00	2.35	0.19	0.08	2.15	90.10	0.04	0.11
	shoe	11.08	0.01	0.15	3.30	0.02	0.10	0.73	84.41	0.21
	bag	13.64	2.69	10.71	1.20	0.55	1.90	5.81	0.13	63.37

		Predicted								
		bg	top	overTop	pants	shorts	skirt	dress	shoe	bag
Ground Truth	bg	99.21	3.82	4.87	9.96	5.57	4.90	4.48	21.20	20.05
	top	0.15	84.45	8.68	0.71	1.48	0.95	3.54	0.00	1.92
	overTop	0.15	6.65	81.83	0.97	0.60	1.20	2.10	0.03	6.31
	pants	0.09	0.34	0.40	84.83	2.48	2.63	0.34	2.21	1.22
	shorts	0.01	0.13	0.04	0.58	77.20	2.24	0.17	0.01	0.15
	skirt	0.03	0.19	0.22	1.53	10.34	72.20	3.66	0.10	1.36
	dress	0.22	4.27	3.35	0.58	1.80	15.28	85.43	0.46	2.77
	shoe	0.09	0.00	0.01	0.69	0.03	0.05	0.05	75.92	0.37
	bag	0.06	0.16	0.62	0.15	0.50	0.55	0.22	0.07	65.86

as ‘pants’ and sometimes it was annotated as ‘background’ (see Figure 3). It resulted in almost 10% of the predicted ‘pants’ pixels being annotated as ‘background’, see confusion matrix Table 3. The issue is beyond the scope of this paper.

The dataset was divided into training, validation and test subsets of sizes 30 429, 3 913 and 9 128 respectively. Stratified sampling was used, so that the clothing categories distribution would be approximately the same in the training, validation and test subset.

4.2 Evaluation Metrics

We used several evaluation metrics to measure the performance of the main clothing parsing task, and also the performance of the tasks solved by the side branches.

Clothing parsing task - we calculated Intersection over Union metrics (IoU), also known as Jaccard index, in several subversions - IoU excluding the background category ‘IoU excl. bg’, IoU including background ‘IoU overall’, IoU of a special segmentation with only two possible classes - foreground and background (how well the model predicts which pixels are clothing pixels regardless of the specific clothing category assignment) ‘IoU Mask’ and IoU of each clothing category separately.

Multilabel classification task - F1, F2 score and accuracy for the clothing classes separately.

Bounding boxes localization task - only visual inspection in combination with the final loss term assessment.

Performance of our final model in terms of these metrics is presented in Table 2 and 4. To our knowledge, there is no clothing parsing research performed on the dataset (Visipedia, 2019), so we resort to comparing our model to a plain U-Net, DeepLabV3, DeepLabV3+ and FPN architectures, which have proved to perform well in the segmentation task.

Table 4: Performance of the ClassBranch of our model.

F1	F2	% of images	Top	Overtop	Pants	Shorts	Skirt	Dress	Shoe	Bag
87.14	87.17	66.26	87.22	91.33	94.82	97.50	93.22	91.17	96.02	92.58

Table 5: Comparison of our model performance without and with the ground truth (GT) clothing category information injection. 'IoU ex. bg' stands for overall IoU excluding the background category and the rest of the columns represent IoU per corresponding clothing category.

Method	IoU ex. bg	Bg	Top	Overtop	Pants	Shorts	Skirt	Dress	Shoe	Bag
Without GT	73.38	97.29	72.49	72.68	79.81	65.90	53.83	77.97	65.97	48.31
With GT	87.37	97.32	87.62	86.25	89.02	83.51	86.27	91.01	67.63	52.94

4.3 Model Performance Analysis

Table 2 summarizes the performance of our model (and two additional) compared to various baselines. Compared to FPN, DeepLabV3, DeepLabV3+ and a plain U-Net, our model achieves the best experimental results. We reason that the contribution of the additional branches lies in increasing the capacity of the model to capture global dependencies.

Additional two models are provided for the ablation study purposes. 'ClassBranch only' represents the model with only the 'ClassBranch' with its output being incorporated back into the segmentation trunk by means of multiplication. 'Branches' stands for the model with both 'ClassBranch' and 'BBbranch', but no incorporation into the main segmentation trunk is performed. Advantage of the 'Branches' model stems from the fact that during inference, both branches can be detached. Hence in inference time no extra computation is required compared to the plain U-Net, while still achieving better performance.

Confusion matrix calculated on the segmentation masks shown in Table 3 clearly shows that our model is not performing well in distinguishing between dress vs skirt + top, between top vs overtop and shorts vs skirt combinations. As Figure 4 demonstrates, due to diversity in the fashion domain, different viewpoints, poses, clothes deformities in the input image and sometimes absence of referential human models, image clothing category classification task poses a challenge even for a human. Performance metrics of the ClassBranch shown in Table 4 (particularly surprisingly low percentage of matched images) indicate the same issue.

To put it quantitatively, we injected the ground truth clothing category information to our model during inference and performed model evaluation in this new setting. The results of this evaluation can be seen in Table 5. With the ground truth injection IoU excluding background metric rises from 73.38 to 87.37. The evaluation supports the hypothesis that clothing category identification is the bottleneck of our model. Figure 5 visually demonstrates the effect of partial ground truth information injection to the model.

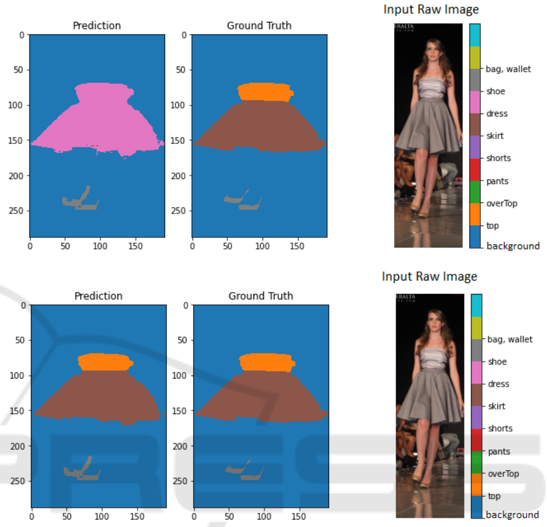

Figure 5: Demonstration of the effect of the partial ground truth clothing category information injection to the model. First row depicts output of our model with no prior information. Second row shows output of our model after injection of the 'no-dress' information (probabilities for the 'dress' category were set to 0).

On the other hand, the high value of IoU of segmentation with only foreground and background classes 'Iou Mask' implies good ability of our model to eliminate the background.

4.4 Qualitative Evaluation

To visually demonstrate the performance of our model, we randomly sampled 1 000 images from the test subset and sorted outputs according to their loss. Figure 6 provides example outputs at each of the 3 loss levels. Majority of the failure cases were because of the wrong clothing category assignment, as discussed in the previous subsection. This issue is more frequent in images of women fashion since it is more diverse. Our model shows good performance in the background elimination.

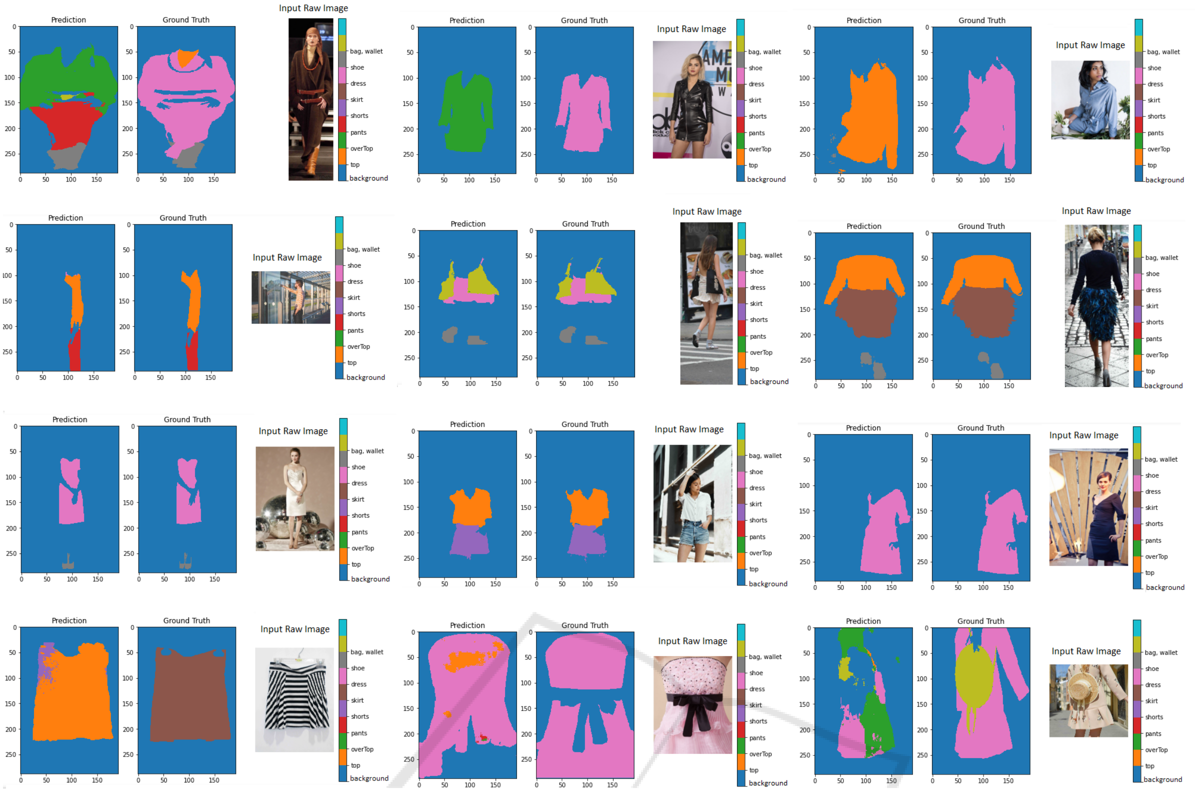

Figure 6: Visual demonstration of the performance of our model. 1 000 images were randomly sampled from the test dataset, and corresponding outputs were sorted according to their loss value. The first row represents outputs with the highest loss value, the second outputs from the middle loss category, and the third row presents outputs with the lowest loss term. The last row provides examples of failure cases.

5 CONCLUSION AND FUTURE WORK

This paper addresses the clothing parsing task with motivation of its applicability in real-life scenarios. Inspired by (Tangseng et al., 2017), an extended version of U-Net architecture was proposed by attaching two side branches. The first branch solves the multilabel clothing category classification task, the other localizes bounding boxes of the clothing items. This paper introduces a simplified version of the clothing parsing task. Refinement of the iMaterialist dataset (Visipedia, 2019) was performed. The empirical results presented in this paper support the hypothesis that additional branches of our model contribute to the performance improvement.

The model performance analysis subsection indicates that the bottleneck of our model is clothing category classification. Therefore, our recommendation is to inject to the model any (even partial) known prior knowledge about which clothing categories are present in the input image. For example, in automatic color extraction for fashion e-shops application, one should for images in ‘skirts’ subsection inject this ‘skirt, but no dress’ information.

In our future work, we would like to address the issue of our model’s confusion with the clothing categories. We would like to explore whether incorporating the annotators’ confusion regarding clothing category assignment could be useful.

REFERENCES

Aoki, R., Nakajima, T., Oki, T., and Miyamoto, R. (2019). Accuracy improvement of fashion style estimation with attention control of a classifier. In *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pages 289–294. IEEE.

Ge, Y., Zhang, R., Wang, X., Tang, X., and Luo, P. (2019). Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5337–5345.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Howard, J. et al. (2018). Fastai. https://github.com/fastai/fastai.

Khurana, T., Mahajan, K., Arora, C., and Rai, A. (2018). Exploiting texture cues for clothing parsing in fashion images. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2102–2106. IEEE.

Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., and Yan, S. (2013). Fashion parsing with weak color-category labels. *IEEE Transactions on Multimedia*, 16(1):253–265.

Loshchilov, I. and Hutter, F. (2017). Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Martinsson, J. and Mogren, O. (2019). Semantic segmentation of fashion images using feature pyramid networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*.

Massa, F. and Girshick, R. (2018). Maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Simon, J., Bilodeau, G.-A., Steele, D., and Mahadik, H. (2020). Color inference from semantic labeling for person search in videos. In *International Conference on Image Analysis and Recognition*, pages 139–151. Springer.

Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE.

Tangseng, P., Wu, Z., and Yamaguchi, K. (2017). Looking at outfit to parse clothing. *CoRR*, abs/1703.01386.

Visipedia (2019). iMaterialist Competition - Fashion. https://github.com/visipedia/imat_comp.

Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., and Hays, J. (2018). Texturegan: Controlling deep image synthesis with texture patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8456–8465.

Yakubovskiy, P. (2020). Segmentation models pytorch. https://github.com/qubvel/segmentation_models.

Yamaguchi, K., Kiapour, M. H., Ortiz, L. E., and Berg, T. L. (2012). Parsing clothing in fashion photographs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577. IEEE.

Zheng, S., Yang, F., Kiapour, M. H., and Piramuthu, R. (2018). Modanet: A large-scale street fashion dataset with polygon annotations. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1670–1678.