Robust Semantic Segmentation for Street Fashion Photos

Anh H. Dang*, Wataru Kameyama†
*GITI, Waseda University, Tokyo, Japan
†Faculty of Science and Engineering, Waseda University, Tokyo, Japan
anh@aoni.waseda.jp, wataru@waseda.jp

***Abstract*—In this paper, we aim to produce the state-of-the-art semantic segmentation for street fashion photos with three contributions. Firstly, we propose a high-performance semantic segmentation network that follows the encoder-decoder structure. Secondly, we propose a guided training process using multiple auxiliary losses. And thirdly, the 2D max-pooling-based scaling operation to produce segmentation feature maps for the aforementioned guided training process. We also propose mIoU+ metric taking noise into account for better evaluation. Evaluations with the ModaNet data set show that the proposed network achieves high benchmark results with less computational cost compared to ever-proposed methods.**

***Index Terms*—semantic segmentation, street fashion photos, label pooling, mIoU+**

I. Introduction

DIFFERENT from the classic object detection and classification problem, semantic segmentation requires each pixel in the input image to be assigned to a class of objects. Fig. 1 shows examples of inputs and corresponding ground-truth labels in semantic segmentation problem for street fashion photos.

Fully convolutional neural network (FCN) for semantic segmentation [1] has laid the foundation for applying CNN into dense segmentation. Recently proposed models such as SegNet [2], DeepLabv3+ [3], and PSPNet [4] have achieved high benchmark results on data sets such as MSCOCO [5], CityScapes [6], and ADE20K [7].

ModaNet [8] is the first large-scale street fashion data set with pixel-level annotation published by S. Zheng et al.. This data set consists of 55,176 fully annotated images, where 52,377 images are for training and the remaining 2,799 images are used for validation.

In this paper, we aim to produce the state-of-the-art semantic segmentation for street fashion photos with three contributions. First, we propose a lightweight asymmetric network that follows the encoder-decoder structure. Secondly, we propose a guided training process with auxiliary training objectives. And thirdly, the 2D max-pooling-based scaling operation is proposed to produce labels to be used in one of the auxiliary training objectives. For a better evaluating segmentation result, we also propose the mIoU+ metric. Different from the conventional mIoU metric, which only counts the classification accuracy of individual pixels, segmentation noise is also taken into account in mIoU+. By both mIoU and mIoU+, the proposed network achieves the highest benchmark result in ModaNet while keeping less computational cost compared with ever-proposed methods.

Fig. 1. Samples from our custom street fashion data set. In the top row, original images are shown, and in the bottom row, corresponding segmentation ground truths are shown. The class names for each color are shown in Table I. Photos are public domain works downloaded from Pexels.com, and labels are manually annotated by the authors.

The rest of the paper is organized as follows. In Section II, we introduce previous works on semantic segmentation and the use of auxiliary loss functions. In Section III, we describe our network design, auxiliary loss functions that we use to train the network, including image pyramid loss, segmentation pyramid loss, and label pooling loss. In Section IV, we describe the experimental setting, the mIoU+ metric, and the evaluation result. The paper is concluded in Section V.

II. Related Works

A. Pre-Deep Learning Era

Semantic segmentation has been a challenge in the field of computer vision. Before the deep learning era, the state-of-the-art works have been based on Texton Forest [9] and conditional random field (CRF) [10]. CRF is still being used

Manuscript received October 19, 2019. This research is supported by funding from leaftnet Co., Ltd. in Japan.

Anh H. Dang is with GITI, Waseda University, Tokyo, Japan. (corresponding author, phone: +81-80-1367-9637, email: anh@aoni.waseda.jp)

Wataru Kameyama is with Faculty of Science and Engineering, Waseda University, Tokyo, Japan (email: wataru@waseda.jp)

as a post-process method to refine the segmentation output [1], [4], [11]–[14].

B. Fully Convolutional Neural Network

Early works on this topic mostly adopt the straight network design. The first proposed model is FCN [1]. The most important contribution of FCN is converting the fully connected classification layers of image classification networks into a 1×1 (i.e., pointwise) convolutional layers to produce pixel-level segmentation prediction. Hence, it can be implemented on top of the ever-proposed classification models such as GoogLeNet [15], VGG [16], and ResNet [17]. The authors of FCN have found that their proposal works best by using VGG-16 as the network base.

PSPNet [4] introduces the spatial pyramid pooling scheme, which results in better context-awareness in the final result. In this pyramid pooling scheme, features maps from different layers of the base network are resized and concatenated. The concatenated feature map is then used as input for a pointwise CNN to produce segmentation results.

C. Encoder-Decoder Based

Later works on the topic mostly utilize the encoder-decoder structure. Models following this approach usually yield better performance. Popular models in this category include SegNet [2] and U-Net [18]. SegNet is a CNN based autoencoder. It utilizes the indices from 2D max-pooling layers in the encoder to upscale the feature map using unpooling layers in the decoder. U-Net implements skip connections between the corresponding encoder and decoder blocks.

D. Dilated Convolutional Neural Network

In [11], F. Yu et al. propose both dilated convolutional neural network (DCNN) for semantic segmentation and a reference network design. DCNN allows the deeper layers of the network to capture the context without losing resolution. The main drawback of this design is the great demand for computational resources because the feature map is rarely down-sampled.

DeepLabv3+ [3] combines all of the above approaches and achieves state-of-the-art performance in many benchmarks.

E. Auxiliary Losses

As networks become deeper, new challenges arise. One of the most challenging problems is the vanishing gradient [19]. In this problem, the gradient becomes too small in the layers being far away from the training loss function.

At first, auxiliary losses are commonly used to overcome the problem. For example, in GoogLeNet [15], besides the main softmax classification loss at the end of the network, another two similar classification losses are added into the middle of the network. Thus, the weights of early blocks are learned mostly by gradient propagated from auxiliary losses. In the research on GANs [20], besides the usual real or fake discrimination, Chen et al. propose an auxiliary loss to discriminate the orientation of the input and output pairs to produce a more robust model. Undoubtedly, selecting the type of auxiliary objectives and their position greatly influences the performance of the network. The auxiliary training objectives also depict the type of features learned by the network. Thus, it does not guarantee that the best feature would be learned.

Another solution to the gradient vanishing problem is using skip connections [21], [22]. In [23], skip connections are used to patch feature maps from early blocks to deeper blocks of the encoder. In [24], ResBlocks [17] are used to replace the conventional CNN blocks in both encoder and decoder, resulting in a very deep encoder-decoder based network.

Even though skip connection has become more popular due to its simplicity, it is not the replacement for auxiliary loss. Perhaps, they can be complements to each other. In [4], Zhao et al. conduct an ablation study for auxiliary loss on ResNet [17] based FCN [1]. By adding an auxiliary loss after the res4b22 residue block and weighted it appropriately, the network performance is gained by 0.94% on pixel accuracy. In [25], multiple spatially scaled versions of training labels are used as auxiliary training objectives.

TABLE I
DATA SET STATISTIC

Id.	Color	Class	Inst. Count		Avg Inst. Size	
			Train	Val	Train	Val
0		Background	–	–	–	–
1		Bag	19,603	948	2.46%	2.53%
2		Belt	13,081	636	0.46%	0.44%
3		Boots	6,719	365	2.40%	2.36%
4		Footwear	37,468	1,753	0.94%	0.93%
5		Outer	22,597	1,093	7.43%	7.42%
6		Dress	13,764	662	10.46%	10.52%
7		Sunglasses	8,340	411	0.30%	0.30%
8		Pants	21,950	1,064	5.65%	5.47%
9		Top	33,131	1,544	4.79%	5.04%
10		Shorts	6,709	322	2.75%	2.83%
11		Skirts	12,953	622	6.37%	6.23%
12		Headwear	5,164	281	1.22%	1.21%
13		Scarf & Tie	4,711	284	2.85%	3.20%

III. PROPOSAL

A. Motivation

1) Problems: Two common problems of semantic segmentation are category confusion and inconspicuous segmentation [4]. Despite efforts to tackle the problems in previous researches such as [3] and [4], the problems still occur on street fashion photos, as shown in Fig. 5 in Appendix A.

In category confusion, the models fail to identify the correct class of the whole segment. For example, PSPNet fails to identify the outerwear in the image (n). And with the image (a) and (b), all the models recognize boots as an ordinary footwear. Another example of this problem is the segmentation of the image (k) by DeepLabv3+. We observe that this problem usually happens with networks that have high context-awareness.

When the above-mentioned problem is limited to local areas, it creates inconspicuous segmentation. For example, with

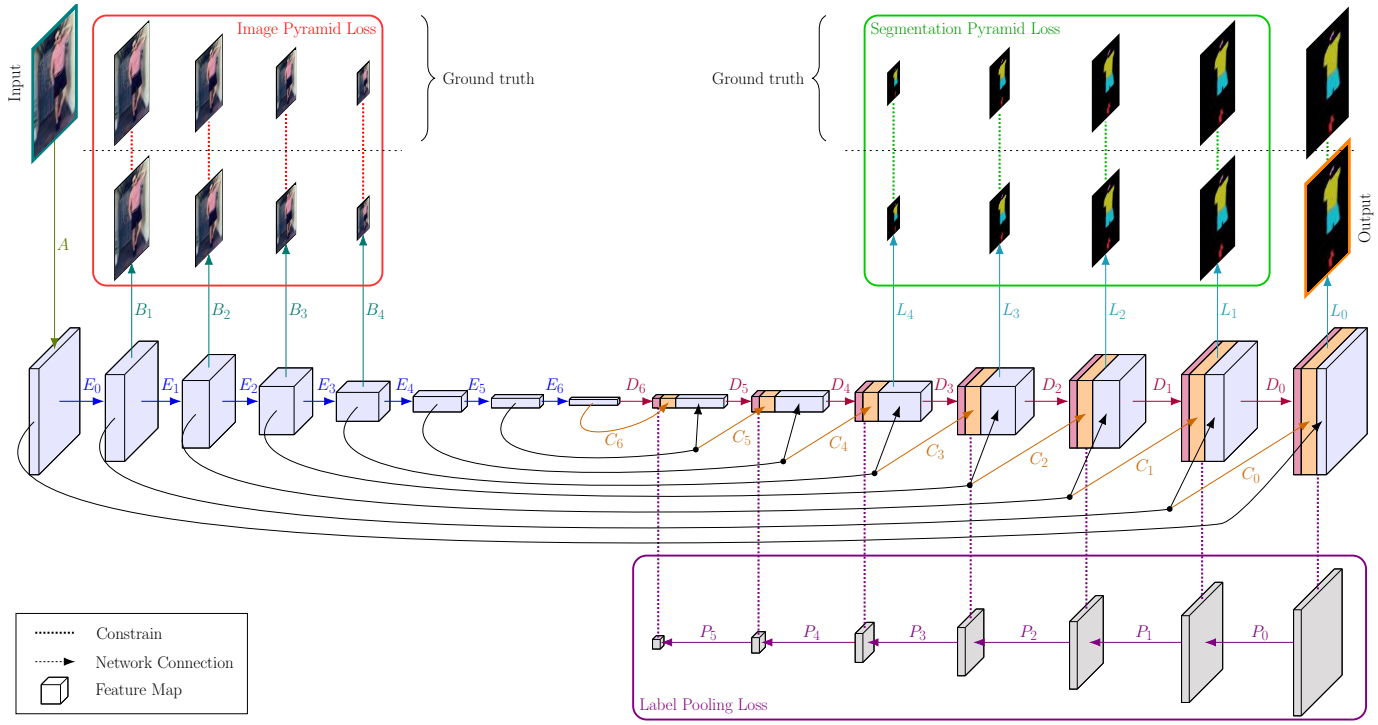

Fig. 2. Overview of the Proposed Network Structure with all three auxiliary losses. The three auxiliary losses are explained in detail in Section III-D. Ground truth for Image Pyramid Loss and Segmentation Pyramid Loss are scaled versions of the image and the ground truth segmentation. However, in Label Pooling Loss, the initial ground truth on the bottom right of the figure is one-hot vector version of the ground truth segmentation. This one-hot version of segmentation is then progressively scaled-down by $P_{[0..5]}$. This label pool feature is explained in detail in Section III-C. Moreover, in Label Pooling Lost, constrains (·····) are made only between label pool feature maps ( ) and output of decoders ( ). Network connections that are not necessary to generate output from input are ignored during inference. The detailed configuration of the whole network is described in Table II.

input image (i), SegNet detects parts of the dress as top-wear and outer-wear. With image (f), PSPNet frequently confuses between pants and skirts. Thus, it results in segmentation with a considerable amount of noise.

PSPNet [4] deliberately addresses this problem by proposing the spatial pyramid pooling module (PPM). This module is expected to increase the size of the receptive field of the network. PPM is also adopted in [3]. However, it appears that the receptive field is still limited for the street fashion problem. A possible reason is that the PPM operates on the feature map produced by a CNN head. Thus, information is already lost during the process, and the important information may not be produced simply by pooling the feature map.

2) Direction: We observe that, for street fashion photo, the above-mentioned problems can be eliminated by knowing whatever a type of apparels is presented in the whole image. For example, in the image (d) of Fig. 5, there are only two types of apparels presented in the image that are dress and pants (a small dark gray area under the model's left arm as in the ground truth image). Thus, if a network only considers dress and pants for the segmentation result, the problem of class confusion and inconspicuous would greatly be reduced.

On the one hand, it is uncomplicated to create a separated model to detect whatever the type of clothes is presented in an image. On the other hand, it is not efficient to create and train separated networks to solve a single problem. Therefore, we merge two types of networks into one and further extend the concept of apparel detector to all of the scales.

3) Implementation: Based on the encoder-decoder structure, we first set the length of the network so that the feature map at the end of the decoder is 1×1. It is to ensure the high context awareness of the network. Secondly, at every scale of the decoder, we expect the network to produce a prediction to indicate whatever the type of clothes is presented in the receptive field of the corresponding pixel of the feature map, i.e., the network first detects the presence of apparel type over the whole image, and then refines it until reaching the required resolution. Ground truth for such prediction can be produced by applying 2D max-pooling on a one-hot vector form of the original ground truth. This process is explained in detail in Section III-C.

B. Network Structure

Fig. 2 shows the structure of our proposed network. It comprises two main parts: encoder and decoder. Both the encoder and decoder parts consist of 7 CNN blocks (E_i and D_i blocks, where $i \in [0..6]$). Feature map is downscaled every time it is processed by an encoder block, and correspondingly upscaled every time it is processed by a decoder block. We organize this network into 7 different levels based on 7 different scales of the feature maps.

Similar to U-Net, skip connections with identity function are implemented between encoder and decoder blocks of the same level (black arrows ⟶ as in Fig. 2). However, in our proposal, the feature map produced by an encoder also leaks

TABLE II
NETWORK PARAMETERS

Block	Output	Filters	Kernel	St.[a]	Pd.[b]	Ac./Op.[c]
A	224×224	32	5×5	1	2	ReLu
		64	3×3	1	1	ReLu
		32	1×1	1	0	ReLu
B_1	112×112	64	3×3	1	1	ReLu
		3	3×3	1	1	Sigmoid
B_2	56×56	– ” –	– ” –	– ” –	– ” –	– ” –
B_3	28×28	– ” –	– ” –	– ” –	– ” –	– ” –
B_4	14×14	– ” –	– ” –	– ” –	– ” –	– ” –
E_0	112×112	64	4×4	2	1	ReLu
		128	3×3	1	1	ReLu
		64	1×1	1	0	ReLu
E_1	56×56	128	– ” –	– ” –	– ” –	– ” –
E_2	28×28	256	– ” –	– ” –	– ” –	– ” –
E_3	14×14	512	– ” –	– ” –	– ” –	– ” –
E_4	7×7	1024	– ” –	– ” –	– ” –	– ” –
E_5	3×3	1024	3×3	3	1	ReLu
		2048	3×3	1	1	ReLu
		1024	1×1	1	0	ReLu
E_6	1×1	1024	3×3	1	0	ReLu
		2048	1×1	1	0	ReLu
		1024	1×1	1	0	ReLu
C_0	112×112	128	3×3	1	1	ReLu
	224×224	1	2×2	2	0	UnPool
C_1	112×112	– ” –	– ” –	– ” –	– ” –	– ” –
C_2	56×56	– ” –	– ” –	– ” –	– ” –	– ” –
C_3	28×28	– ” –	– ” –	– ” –	– ” –	– ” –
C_4	14×14	– ” –	– ” –	– ” –	– ” –	– ” –
C_5	3×3	128	3×3	1	1	ReLu
	7×7	1	3×3	3	1	UnPool
C_6	1×1	128	1×1	1	0	ReLu
	3×3	1	3×3	3	0	UnPool
D_0	112×112	128	3×3	1	1	ReLu
		14	3×3	1	1	Sigmoid
	224×224	1	2×2	2	0	UnPool
D_1	112×112	– ” –	– ” –	– ” –	– ” –	– ” –
D_2	56×56	– ” –	– ” –	– ” –	– ” –	– ” –
D_3	28×28	– ” –	– ” –	– ” –	– ” –	– ” –
D_4	14×14	– ” –	– ” –	– ” –	– ” –	– ” –
D_5	3×3	128	3×3	1	1	ReLu
		14	3×3	1	1	Sigmoid
	7×7	1	3×3	3	1	UnPool
D_6	1×1	128	1×1	1	0	ReLu
		14	1×1	1	0	Sigmoid
	3×3	1	3×3	1	0	UnPool
P_0	56×56	1	2×2	2	0	MaxPool
P_1	28×28	– ” –	– ” –	– ” –	– ” –	– ” –
P_2	14×14	– ” –	– ” –	– ” –	– ” –	– ” –
P_3	7×7	– ” –	– ” –	– ” –	– ” –	– ” –
P_4	3×3	– ” –	3×3	3	1	– ” –
P_5	1×1	– ” –	3×3	1	1	– ” –
L_0	224×224	128	3×3	1	1	ReLu
		14	3×3	1	1	Softmax
L_1	112×112	– ” –	– ” –	– ” –	– ” –	– ” –
L_2	56×56	– ” –	– ” –	– ” –	– ” –	– ” –
L_3	28×28	– ” –	– ” –	– ” –	– ” –	– ” –
L_4	14×14	– ” –	– ” –	– ” –	– ” –	– ” –

[a]Stride, [b]Padding, [c]Activation/Operation

into the next level of the decoder. To adapt the feature map into the larger scale, we employ CNN - 2D unpooling blocks C_i. In our network, encoder blocks scale down the feature map by utilizing CNN with stride 2 instead of using 2D max-pooling operation. Thus, different from SegNet [2], the 2D unpooling layer in our network doesn't utilize the pooling indices.

As mentioned, in this network, we expect decoder blocks to produce the prediction on the presence of a class within the whole image and then gradually refine the prediction result. Therefore, all the decoder blocks have the same design that output only 14 channels feature map, which is the number of segmentation classes of ModaNet (13 classes plus background, as shown in Table II).

Element-wise sigmoid function is used as the activation function for D_i blocks as follows.

$$d_i = \frac{1}{1 + \exp(-d'_i)} \tag{1}$$

Where d_i is the output of D_i block, and d'_i is the pre-activation value of D_i.

Segmentation prediction is produced by L_0 block. In this network, besides L_0, there are another 4 L_i blocks where $i \in [1..4]$. These blocks produce smaller-scale versions of the segmentation prediction. In general, the scale of the prediction produced by L_i block is 2^{-i}. The input of L_i block is the concatenation of feature maps output from C_i, D_i and E_i blocks. Pixel-wise softmax is used to produce the output of L_i blocks as follows.

$$l_{ij} = \frac{\exp\left(l'_{ij}\right)}{\sum_{k=1}^{K} \exp\left(l'_{ik}\right)} \tag{2}$$

Where l_{ij} denotes the value of channel j of the feature map produced by L_i, l'_{ij} denotes the pre-activation value of l_{ij}, and K denotes the total number of channels which also is the number of segmentation classes.

From level 1 to level 4, different scales of the input image are reconstructed by B_i blocks where i is the level number. The input of B_i block is the feature map e_{i-1} produced by E_{i-1} block. All the B_i blocks reconstruct the input at the scale of 2^{-i}. Thus, all the outputs from B_i blocks create a spatial scale pyramid of the input image. Element-wise sigmoid, as in (1) is used as the activation function for B_i layers. Thus, different from works such as [25] and [26], we are not using the image pyramid as input but as auxiliary training objective.

C. Label Pooling

Previous works involving multiple-scale inputs or outputs only consider spatial scaling. In [25], they are used as an auxiliary training objective. In [26], they are used to create multi-scale fusion features. In [4] and [3], the network is trained with different spatial scaled versions of input and output to produce more robust features.

However, with spatial scaling, details from the original input eventually are lost at smaller scales. To avoid such problems, instead of spatially scaling the label, we use max-pooling operation on the one-hot label vector to produce multiple scales of labels. As such, the existence of a segment is

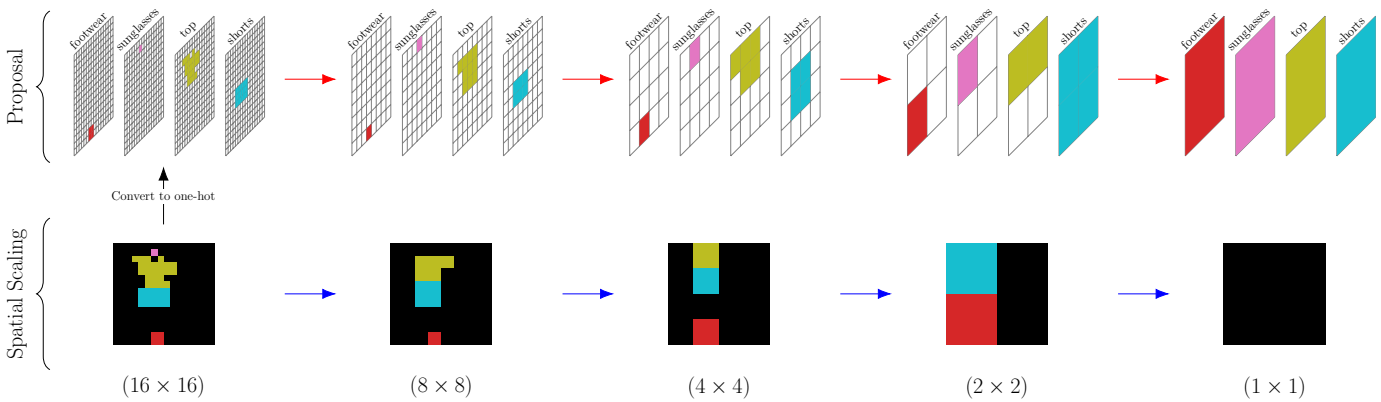

Fig. 3. Comparison between proposed 2D max-pooling-based label scaling and conventional label scaling. With conventional label scaling, the label is progressively scaled-down using nearest-neighbor interpolation (blue arrows →). In our proposal, the original label (bottom left) is first converted to one-hot vectors (top left) and then progressively scaled-down by 2D max-pooling operation (red arrows →). The segmentation color codes in the label are described in Table I. On the top row, classes rather than footwear, sunglasses, top, and shorts are ignored.

preserved even in the smallest scale. This process is illustrated in Fig. 3.

In Fig. 3, the spatial scaling operation makes the existence of segmentation vanished. After the first scale down operation, the segmentation of sunglass class has vanished. From the scale of 4×4 to 2×2, the top segment has vanished. By the time of scaling down to 1×1 pixel, all the segmentations vanished. On the other hand, the proposed label pool features retain all of the segmentation even at the smallest scale.

Shown in Fig. 2, D_i blocks are guide-trained by the result of P_i blocks where i is the level number, and P_i blocks are 2D max-pooling operation on the input label. Their configuration is shown in Table II. This is to avoid the detail loss when scaling down the label. Also shown in Table II, the strides of P_i are matched with the strides of D_i and E_i blocks. Furthermore, the kernel size of P_i is also matched with the kernel size of D_i.

D. Training Objectives

With the segmentation prediction l_0 coming from L_0, we utilize pixel-wise cross-entropy as a training objective.

$$H\left(t, l_0\right) = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{K-1} t_{ij} \log\left(l_{0ij}\right) \tag{3}$$

Where t is the ground-truth, t_{ij} is the j-th channel of the i-th pixel of t, l_{0ij} is the j-th channel of the i-th pixel of l_0, K is the number of segmentation class, and N is the total number of pixel in the output. Besides this conventional training criteria, we introduce the three auxiliary training objectives as follows.

1) Image Pyramid Loss (IPL): Different from popular works, we do not utilize multi-scaled input to reinforce the training process. Instead, we expect the network to reconstruct scaled versions of the input using feature maps produced by encoder blocks. Thus, additional scaled inputs and processing are not required during the inference process. We penalize the difference between output b_i from B_i block and the input image x_i by binary cross-entropy loss as follows.

$$H\left(x_i, b_i\right) = -\frac{1}{N} \sum_{j=0}^{N-1} \Big(x_{ij} \cdot \log\left(b_{ij}\right) + \left(1 - x_{ij}\right) \cdot \log\left(1 - b_{ij}\right) \Big) \tag{4}$$

Where x_i and b_i are input image and reconstructed image at scale 2^{-i} with $i \in [1..4]$, x_{ij} and b_{ij} are the j-th element of x_i and b_i, N is the total number of elements in x_i and b_i (i.e. number of pixel $\times$ number of color channels). We use binary cross entropy (i.e. log loss) as the error function. Then, the image pyramid loss is calculated by:

$$IPL = \frac{1}{4} \sum_{i=1}^{4} H(x_i, b_i) \tag{5}$$

2) Segmentation Pyramid Loss (SPL): It is the average of the cross-entropy between segmentation and ground truth across different scales.

$$SPL = \frac{1}{4} \sum_{i=1}^{4} H(t_i, l_i) \tag{6}$$

Where $H(\cdot)$ is binary cross-entropy loss similar to (4), t_i and l_i are ground truth and predicted segmentation at scale 2^{-i} with $i \in [1..4]$.

3) Label Pooling Loss (LPL): It is the average of binary cross-entropy loss between label pool features and output of decoders across different scales as follows.

$$LPL = \frac{1}{6} \sum_{i=1}^{6} H(p_i, d_i) \tag{7}$$

Where $H(\cdot)$ is binary cross-entropy loss similar to (4), p_i and d_i are ground truth and prediction of label pool feature at scale 2^{-i}.

The final loss is calculated by averaging all the above mentioned losses as follows:

$$loss = \frac{1}{4} \left(H\left(t, l_0\right) + IPL + SPL + LPL\right) \tag{8}$$

Fig. 4. Illustration of segmentation results. Photos are public domain works downloaded from Pexels.com. Label are manually annotated by the authors.

IV. Evaluation

Using the ModaNet data set, we compare our model with U-Net [18], PSPNet [4], SegNet [2], and DeepLabv3+ [3].

A. Data Set

We split the original training set into new training and evaluation sets. The new evaluation set consists of 2,400 images, and the new training set consists of the remaining 49,977 images.

The randomized splitting process is constrained so that there are at least 280 instances of each class available in the evaluation set to ensure the quality of evaluation. The statistic of training and validation data sets are shown in Table I.

B. Data Augmentation

We train and evaluate all the networks with input and output sizes of 224×224. To make the network more robust, the following pipeline is used for data augmentation:

1) Random horizontal flipping
2) Random expanding with a max expansion ratio of 1.5. In this step, black bars of random size t, b, l and r are padded into the original image so that $l + r \leq 0.5 \times w$ and $t + b \leq 0.5 \times h$, where w and h are width and height of the input of this step, t and b are the sizes of black bars padded on the top and bottom of the image, and l and r are the sizes of black bars padded on the left and right side of the image.
3) Randomly cropping the image with the scale ratio range $(0.5, 1]$ and aspect ratio range $[\frac{3}{4}, \frac{4}{3}]$. Thus, width w and height h of the cropping window are randomized so that:
   - $w \leq w_0$ and $h \leq h_0$
   - $\frac{3}{4} \leq \frac{w}{h} \leq \frac{4}{3}$
   - $0.5 \times (w_0 \times h_0) < w \times h \leq w_0 \times h_0$

   Where w_0 and h_0 are the width and height of the input of this step. Hence, the top left corner of the cropping window (x, y) must satisfy the following conditions:
   - $0 \leq x \leq w_0 - w$
   - $0 \leq y \leq h_0 - h$
4) Adding Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 25.5$. Thus, the output image is $x = \min(255, \max(0, x + \mathrm{G}(\mu, \sigma)))$ where $\mathrm{G}(\cdot)$ is the Gaussian function.
5) Resize to $224 \times 224 \times 3$

We then normalize the input image by scaling pixel value into $[0, 1]$ range. Since the label is an image containing pixel-level segmentation of the input image, it also needs to be augmented correspondingly, except for the step 4. Furthermore, nearest-neighbor sampling must be used in all the steps that involve interpolation to preserve class information.

C. Metrics

1) mIoU: We utilize intersection over union (IoU, i.e., Jaccard distance) as the performance metric. We first compute the IoU of individual class as follows.

$$IoU_i = \frac{1}{N} \sum_{j=0}^{N-1} \frac{|T_{ij} \cap L_{ij}|}{|T_{ij} \cup L_{ij}|} \tag{9}$$

Where IoU_i is the IoU score of class i, N is the total number of photos in the data set, T_{ij} is the set of all the

TABLE III
IoU OF INDIVIDUAL CLASSES

	A	B	C	D	E	F	G	H	Unet	DLv3+[a]	PSPNet	SegNet
Background	0.979	0.979	0.978	0.979	*0.979*	**0.980**	0.978	0.979	0.979	0.975	0.977	0.955
Bag	0.690	0.692	0.691	0.694	0.693	0.689	0.707	0.694	*0.701*	0.674	**0.708**	0.429
Belt	*0.465*	0.462	0.467	0.442	0.456	0.454	0.454	0.440	**0.479**	0.394	0.415	0.205
Boots	0.556	**0.577**	*0.570*	0.543	0.575	0.557	0.569	0.559	0.561	0.535	0.567	0.369
Footwear	0.630	0.624	0.629	0.628	**0.638**	0.638	0.631	0.622	*0.637*	0.586	0.555	0.452
Outer	0.642	*0.644*	0.623	0.629	0.639	0.651	0.627	0.638	0.623	0.625	**0.665**	0.438
Dress	0.669	*0.668*	0.633	0.663	0.651	0.657	0.651	0.649	0.594	0.660	**0.689**	0.462
Sunglasses	0.634	0.611	0.652	0.606	*0.650*	0.625	0.646	0.621	**0.675**	0.531	0.534	0.321
Pants	*0.805*	**0.810**	0.793	0.790	0.802	0.808	0.801	0.792	0.787	0.761	0.800	0.683
Top	0.647	0.650	0.625	0.641	*0.653*	0.660	0.641	0.652	0.624	0.610	**0.679**	0.478
Shorts	0.686	**0.718**	0.686	0.697	0.697	0.692	0.688	0.708	0.662	0.715	*0.711*	0.487
Skirts	0.661	*0.674*	0.659	0.671	0.646	0.673	0.666	0.652	0.618	0.683	**0.709**	0.503
Headwear	0.608	0.586	*0.606*	0.584	0.582	0.590	0.606	0.590	**0.618**	0.545	0.594	0.258
Scarf & Tie	0.393	*0.429*	0.396	0.409	0.439	0.426	0.397	0.395	0.420	0.370	**0.473**	0.129
mIoU	0.648	**0.652**	0.643	0.641	*0.650*	0.650	0.647	0.642	0.641	0.619	0.648	0.441
inference time (ms)	100.96	78.99	68.81	68.80	75.11	75.23	*66.16*	66.05	116.24	137.83	115.98	**29.61**
training time (h)	19.568	19.328	17.825	*17.625*	19.359	19.297	17.831	17.483	30.84	33.64	23.68	**9.04**

[a]DeepLabv3+

pixels belongs to the i-th class in j-th ground truth, L_{ij} is the set of all pixels predicted as i-th class in the j-th prediction, and $|\cdot|$ is the cardinality of a set. Thus, the mIoU metric is calculated as follows.

$$mIoU = \frac{1}{M}\sum_{i=0}^{M-1} IoU_i \tag{10}$$

Where M is the total number of segmentation classes.

2) mIoU+: Because the mIoU metric favors the total number of accurately classified pixels, a prediction with noise frequently results in higher mIoU compared to a prediction with no noise. Depending on the application, prediction with low noise may be favored over absolutely high mIoU prediction.

Therefore, we propose mIoU+ (mIoU-plus) metric in which noise is taken into account. This metric is not based on individual pixel but connected components (i.e. individual segments). Given a prediction and a ground truth segmentation, the connected component based segmentation score of a class is calculated as follows.

$$CCSS_i(U,V) = \frac{1}{|U_i|}\sum_{u\in U_i} \max_{v\in V_i} IoU\,(u,v) \tag{11}$$

Where $CCSS_i$ is the segmentation score of the i-th class between predicted segmentation U and ground truth V, U_i is the set of all connected components of i-th class in the prediction, V_i is the set of all connected components of i-th class in the ground truth. However, because this score is not symmetric (i.e., $CCSS_i(U,V) \neq CCSS_i(V,U)$), the segmentation score is calculated as follows.

$$SS_i(U,V) = CCSS_i(U,V) \wedge CCSS_i(V,U) \tag{12}$$

Where $SS_i(U,V)$ is the segmentation score of the i-th class between two segmentations U and V. Similar to the conventional mIoU, IoU+ of each class is computed as follows.

$$IoU+_i = \frac{1}{N}\sum_{j=0}^{N-1} SS_i(U_j,V_j) \tag{13}$$

Where $IoU+_i$ is IoU+ score of the i-th class, N is the total number of samples, U_j is the set of connected components from the j-th prediction, and V_j is the set of connected components from the j-th ground truth. The score for a whole segmentation with multiple connected components is calculated as follows.

$$mIoU+ = \frac{1}{K}\sum_{i=0}^{K-1} SS_i(U,V) \tag{14}$$

Where K is the total number of segmentation classes.

D. Ablation Study on Effect of Auxiliary Training Objectives

We investigate more into the effect of auxiliary training objective on the model performance. We retrain our network with different training objective configurations. The loss function used in this experiment is as follows.

$$loss = \frac{H\,(t_0,l_0) + \alpha IPL + \beta SPL + \gamma LPL}{1+\alpha+\beta+\gamma} \tag{15}$$

Where $\alpha,\beta,\gamma \in \{0,1\}$. In practice, when $\alpha = 0$, b_i computations are ignored. Similarly, when $\beta = 0$, $l_{[1..4]}$ computations are ignored. However, when $\gamma = 0$, d_i still need to be computed because it is an integrated part of the model.

There are 8 different configurations of the loss function. We annotate them as configuration A to configuration H, as shown in Table V.

TABLE IV
IOU+ OF INDIVIDUAL CLASSES

	A	B	C	D	E	F	G	H	Unet	DLv3+[a]	PSPNet	SegNet
Background	*0.373*	0.344	0.369	0.336	0.359	0.347	0.339	0.338	0.345	0.379	**0.404**	0.233
Bag	**0.441**	0.398	0.434	0.361	*0.429*	0.392	0.424	0.385	0.373	0.364	0.427	0.124
Belt	0.290	0.257	0.312	0.245	**0.316**	0.257	*0.297*	0.240	0.287	0.204	0.211	0.082
Boots	*0.347*	0.303	0.358	0.269	0.335	0.297	**0.360**	0.280	0.297	0.299	0.301	0.100
Footwear	0.502	0.474	0.503	0.483	*0.503*	0.489	**0.504**	0.475	0.500	0.434	0.418	0.247
Outer	0.412	0.379	0.383	0.356	*0.409*	0.375	0.408	0.329	0.286	0.381	**0.425**	0.104
Dress	0.425	0.393	0.370	0.336	*0.398*	0.382	0.393	0.360	0.208	0.366	**0.434**	0.082
Sunglasses	0.501	0.471	**0.541**	0.433	0.531	0.463	*0.514*	0.402	0.502	0.355	0.392	0.157
Pants	0.629	0.613	**0.642**	0.568	*0.635*	0.621	0.623	0.567	0.557	0.591	0.641	0.316
Top	0.432	*0.439*	0.438	0.394	0.452	0.428	0.401	0.391	0.347	0.406	**0.465**	0.173
Shorts	0.488	0.426	*0.490*	0.393	**0.527**	0.454	0.449	0.374	0.364	0.453	0.497	0.203
Skirts	*0.459*	0.413	0.438	0.370	0.465	0.426	0.454	0.390	0.297	0.436	**0.497**	0.163
Headwear	**0.479**	0.399	0.468	0.362	0.452	0.379	*0.467*	0.310	0.441	0.336	0.400	0.083
Scarf & Tie	**0.278**	0.230	0.239	0.185	0.270	0.186	0.238	0.172	0.184	0.211	*0.263*	0.049
mIoU+	0.433	0.396	*0.428*	0.363	**0.434**	0.392	0.419	0.358	0.356	0.373	0.412	0.151
inference time (ms)	100.96	78.99	68.81	68.80	75.11	75.23	*66.16*	66.05	116.24	137.83	115.98	**29.61**
training time (h)	19.568	19.328	17.825	*17.625*	19.359	19.297	17.831	17.483	30.84	33.64	23.68	**9.04**

[a]DeepLabv3+

E. Settings

We implement all the networks using Chainer deep learning framework [27]. LeCun normal weight initializer [28] is used. The models are trained by Adam optimizer [29] with the learning rate of 1×10^{-3} and the decay rate of 0.99. The machine used to carry out the experiment is a Linux box equipped with three Nvidia Pascal GPUs.

We train each configuration for three times with 100 epochs each and take averages of the best mIoU and mIoU+.

F. Result

We report the experimental results using mIoU metric in Table III. Our proposed network configured with three different auxiliary losses outperforms all the ever-proposed models in terms of performance. Among all the auxiliary loss configurations, configuration B achieves the highest mIoU score. This configuration consists of only image pyramid loss and segmentation pyramid loss. Among the ever-proposed networks, PSPNet achieves the highest mIoU score. Moreover, our top proposed network takes only $2/3$ of the time for training as well as inference compared to PSPNet.

We realize that the mIoU performance of the model is worsened when combining label pooling loss with the other two auxiliary training losses. In fact, configuration B, D, and F achieve higher mIoU compared to configuration A, C, and E.

TABLE V
DIFFERENT AUXILIARY CONFIGURATIONS

	A	B	C	D	E	F	G	H
α	1	1	1	1	0	0	0	0
β	1	1	0	0	1	1	0	0
γ	1	0	1	0	1	0	1	0

We report experimental results using mIoU+ metric in Table IV. The training and inference time in Table IV are carried over from Table III. We observe that all the models achieve their best mIoU and mIoU+ in the same epoch. Furthermore, mIoU and mIoU+ are loosely proportional to each other during the training process.

As shown in Table IV, the proposed model with configuration E achieves the highest mIoU+ score. Configuration E consists of segmentation pyramid loss and label pooling loss. Configuration A with all the auxiliary loss functions is the runner-up. Among the ever-proposed model, PSPNet also achieves the highest mIoU+ score.

The proposed model used to generate samples in Fig. 4 and Fig. 5 is trained with configuration A.

V. CONCLUSION

In this paper, we propose a high-performance semantic segmentation model for street fashion photos. This network infers the existence of the classes over the whole image, and progressively refines the result up to the desired resolution. We also propose a new label pool feature that can be used to improve the performance of the proposed network. And finally, we provide benchmarking results of the network with and without 3 different types of auxiliary loss. For better evaluation, we propose mIoU+ metric in which noises are taken into account.

We compare the performance of our network to the other state-of-the-art networks, including U-Net [18], PSPNet [4], SegNet [2], and DeepLabv3+ [3]. We report the evaluation result using both conventional mIoU and newly proposed mIoU+ metrics. The experiment shows that our network requires less time to train and infer while achieves the highest segmentation performance in both mIoU and mIoU+ metrics.

For future work, we will extend the evaluation on the scene parsing problem using data sets such as MSCOCO [5], CityScapes [6], and ADE20K [7].

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

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

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[8] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "Modanet: A large-scale street fashion dataset with polygon annotations," *arXiv preprint arXiv:1807.01394*, 2018.

[9] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[10] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images." in *Cvpr*, vol. 2, 2011, p. 3.

[11] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[12] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[14] ——, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[19] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[20] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised gans via auxiliary rotation loss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 154–12 163.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[23] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187.

[24] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 11–19.

[25] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.

[26] J. Li, W. Speier, K. C. Ho, K. V. Sarma, A. Gertych, B. S. Knudsen, and C. W. Arnold, "An em-based semi-supervised deep learning approach for semantic segmentation of histopathological images from radical prostatectomies," *Computerized Medical Imaging and Graphics*, vol. 69, pp. 125–133, 2018.

[27] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, vol. 5, 2015, pp. 1–6.

[28] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Anh H. Dang (S'09) received his bachelor degree in business administration, information & communication technology from Ritsumeikan Asia Pacific University (Beppu, Oita, Japan) in 2010. He then received the master degree in computer science from Waseda University (Shinjuku, Tokyo, Japan) in 2012. Since 2012, he is a Ph.D. candidate at Waseda University. He is a member of IEEE, ACM, and IEICE. His research interests are machine learning, artificial intelligence, and computer vision.

Prof. Wataru Kameyama (M'86) received the bachelor's, master's, and D.Eng. degrees from the School of Science and Engineering, Waseda University, in 1985, 1987, and 1990, respectively. He joined ASCII Corporation in 1992, and was transferred to France Telecom CCETT from 1994 to 1996 for his secondment. After joining Waseda University as an Associate Professor in 1999, he has been a Professor with the Department of Communications and Computer Engineering, School of Fundamental Science and Engineering, Waseda University, since 2014. He has been involved in MPEG, MHEG, DAVIC, and the TV-Anytime Forum activities. He was a Chairman of ISO/IEC JTC1/SC29/WG12, and a Secretariat and Vice Chairman of the TV-Anytime Forum. He is a member of IEEE, IEICE, IPSJ, ITE, IIEEJ, and ACM. He received the Best Paper Award of Niwa-Takayanagi in 2006, the Best Author Award of Niwa-Takayanagi in 2009 from the Institute of Image Information and Television Engineers, and the International Cooperation Award from the ITU Association of Japan in 2012.

APPENDIX A
SEGMENTATION RESULTS

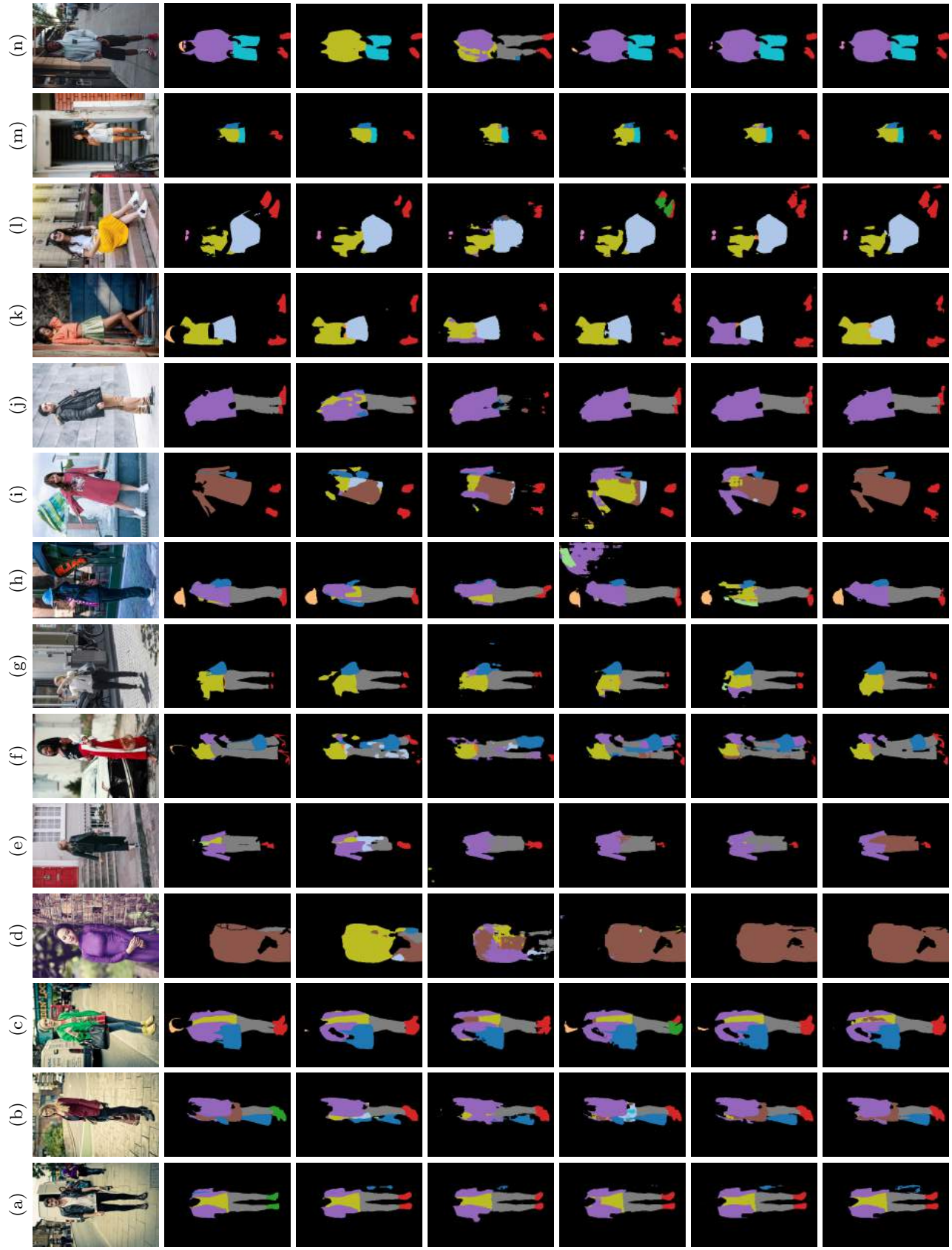

Fig. 5. Illustration of segmentation results coming from the models. Photos are public domain works downloaded from Pexels.com. Label are manually annotated by the authors.