Computers & Graphics 80 (2019) 104–113

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CAD & Graphics 2019

PortraitNet: Real-time portrait segmentation network for mobile device

Song-Hai Zhang[a,*], Xin Dong[a], Hui Li[b], Ruilong Li[a], Yong-Liang Yang[c]

[a] Tsinghua University, Beijing, China
[b] Cisco Systems(China) Research & Development Co, Ltd. Hangzhou Branch, Hangzhou, China
[c] University of Bath, Claverton Down, Bath, UK

ARTICLE INFO

Article history:
Received 9 March 2019
Accepted 23 March 2019
Available online 4 April 2019

Keywords:
Portrait
Semantic segmentation
Boundary loss
Consistency constraint loss
Mobile device

ABSTRACT

Real-time portrait segmentation plays a significant role in many applications on mobile device, such as background replacement in video chat or teleconference. In this paper, we propose a real-time portrait segmentation model, called PortraitNet, that can run effectively and efficiently on mobile device. PortraitNet is based on a lightweight U-shape architecture with two auxiliary losses at the training stage, while no additional cost is required at the testing stage for portrait inference. The two auxiliary losses are boundary loss and consistency constraint loss. The former improves the accuracy of boundary pixels, and the latter enhances the robustness in complex lighting environment. We evaluate PortraitNet on portrait segmentation dataset EG1800 and Supervise-Portrait. Compared with the state-of-the-art methods, our approach achieves remarkable performance in terms of both accuracy and efficiency, especially for generating results with sharper boundaries and under severe illumination conditions. Meanwhile, PortraitNet is capable of processing 224×224 RGB images at 30 FPS on iPhone 7.

1. Introduction

Semantic image segmentation with high accuracy and efficiency using convolutional neural networks has been a popular research topic in computer vision. With the rapid development of mobile techniques, automatic portrait segmentation as a specialized segmentation problem attracts more and more attention, since it favors many mobile applications which require background editing (e.g., blurring, replacement, etc.) on portrait images, as shown in Fig. 1.

Semantic segmentation can be formulated as a dense prediction task. The goal is to predict for every pixel which object class it belongs to. In recent years, semantic segmentation methods based on deep convolutional neural networks [1–6] have made significant progresses. However, compared with general cases, portrait images exhibit unique characteristics: (1) it usually contains at least one person whose face area covers at least 10% of the whole portrait image [7,8]; (2) it often has ambiguous boundaries and complex illumination conditions. In mobile applications, predicting precise segmentation boundaries while adapting to varying illuminations are crucial for user experience. Achieving high accuracy and efficiency at the same time is also challenging. Previous portrait segmentation works [7,8] mainly focus on improving accuracy but not efficiency, thus are not suitable for real-time segmentation on mobile device due to the involvement of sophisticated network architecture. In this paper, we propose a novel semantic segmentation network called PortraitNet, which is specifically designed for real-time portrait segmentation on mobile device with limited computational power. In terms of network architecture, according to the characteristics of portrait images, we configure PortraitNet with $32\times$ down-sampling rate in the encoder module to achieve large receptive field and high inferring efficiency. We also employ U-shape [3] architecture to up-sample the feature maps for better segmentation result. The decoder module consists of refined residual blocks [9] and up-sampling blocks. We modify the residual blocks by replacing the normal convolution blocks by depthwise separable convolution. In terms of network training, as predicting precise segmentation boundaries is difficult for convolutional neural networks, we design an auxiliary boundary loss to help the network generate better portrait boundaries. Meanwhile, we take into account complex illumination conditions in portrait images and utilize the consistency constraint loss to improve the robustness. With the two auxiliary losses, we achieve the accuracy of 96.62% on EG1800 dataset and 93.43% on Supervise-Portrait dataset at 30 FPS on iPhone 7 with input image size of 224×224.

* Corresponding author.
E-mail address: shz@tsinghua.edu.cn (S.-H. Zhang).

https://doi.org/10.1016/j.cag.2019.03.007
0097-8493/

Fig. 1. Portrait segmentation applications on mobile device. (a) Original image. (b) The corresponding segmentation. (c) and (d) Two important applications based on portrait image segmentation.

2. Related work

PortraitNet is related to research in semantic segmentation and lightweight convolutional neural networks. This section reviews typical semantic segmentation methods with deep convolutional networks, and classical lightweight architectures.

Semantic image segmentation is a fundamental research topic in computer vision. Many applications require highly accurate and efficient segmentation results as a basis for analyzing and understanding images. With the recent advances of deep learning, semantic segmentation methods based on deep convolutional neural networks [9–12] have made great achievements, especially for improving segmentation precision. Fully convolution networks [1] is the first essential work that proposed an end-to-end network for pixel-wise segmentation. It also defined a skip architecture to produce accurate masks. SegNet [2] came up with a classical encoder-decoder architecture for segmentation, while a similar method was UNet [3]. The main difference is that SegNet [2] transferred pooling indices from encoder to decoder to produce sparse feature maps, while UNet [3] transferred high resolution features from encoder to up-sampled features in decoder. A series of research works named Deeplab [4–6] presented the most accurate methods of semantic segmentation at present. Deeplabv1 [4] used dilated convolution to maintain the size of feature maps and use CRFs to refine the segmentation result. Deeplabv2 [5] proposed a module called atrous spatial pyramid pooling (ASPP) for improvement. Deeplabv3 [6] removed the CRFs module and modified ASPP module to improve the accuracy. Although these semantic segmentation methods result in high precision, the efficiency is relatively low.

Compared with large models with high complexity, there are some segmentation works that pay more attention to the efficiency. ENet [13] proposed a new network architecture which is deep and narrow, the speed is much faster while the accuracy decline is obvious. ICNet [14] incorporated multi-resolution branches to improve the accuracy of the model, but the model is still too large to run on mobile device. BiSeNet [15] is the state-of-art real-time segmentation method on the CitySpace dataset [16]. However, this method is not suitable for small size input images because of the crude up-sampling modules.

Automatic portrait segmentation as a specialized semantic segmentation is important in mobile computing era. Shen et al. [7] collected the first human portrait dataset named EG1800 and designed a segmentation network to distinguish the portrait and background. Du et al. [8] designed a boundary-sensitive network to improve the accuracy using soft boundary label. Yu et al. [17] proposed a Border Network to improve the accuracy of segmentation. However, the existing works focused on accuracy but not the computational efficiency. With the growing demand of mobile applications, a number of researches aiming at efficient models for mobile device have been proposed [18–21]. Depthwise separable convolutional layers are widely used in lightweight networks. PortraitNet employs MobileNet-v2 [19] as backbone to extract features in the encoder module and uses depthwise separable convolution to substitute traditional convolution in the decoder module to build a lightweight network.

Portrait images usually have complex illumination conditions. Thus how to improve the robustness of the model under varying lighting conditions is very important. Zheng et al. [22] proposed a stability training method to improve the robustness of deep neural networks. Euclidean distance was used in this method to evaluate the results. The stability training process could also benefit segmentation networks. However, Euclidean distance is not a good measurement when most pixels in the prediction differs little from the ground truth. Inspired by model distillation [23], we employ soft label and KL divergence in consistency constraint loss to assist training and improve robustness.

3. Method

In this section, we elaborate our method in detail. We first introduce the architecture of PortraitNet, which is specifically designed for mobile device, and includes two modules, the encoder module and the decoder module. Then, we describe two auxiliary losses used in PortraitNet to improve segmentation accuracy without causing extra cost at the testing stage.

3.1. PortraitNet architecture

Fig. 2 shows the architecture of PortraitNet. The encoder module is used to extract features from the raw RGB image. In contrast to general object segmentation, a portrait often occupies a large area of the whole image. Achieving high accuracy requires a good understanding of rich global and spatial information. Furthermore, in order to achieve real-time performance on mobile device, we use small input size of 224×224 with $32\times$ down-sampling rate in the encoder module while utilizing image global information. Meanwhile, we adopt the U-shape architecture with $32\times$ up-sampling rate in the decoder module to reconstruct spatial information. We concatenate the feature maps as fusion maps in the decoder module to fully exploit the capabilities of the model. Inspired by lightweight research works [18–21], we use depthwise separable convolutions instead of traditional convolutions to improve inferring efficiency. Each convolutional layer is followed by a BatchNorm layer [24] and a ReLU layer. To reduce the complexity of the model, the decoder architecture is relatively simple compared to the encoder. It only contains two main operations, namely up-sampling and transition. Up-sampling layers employ de-convolution to up-sample the feature maps. Each layer

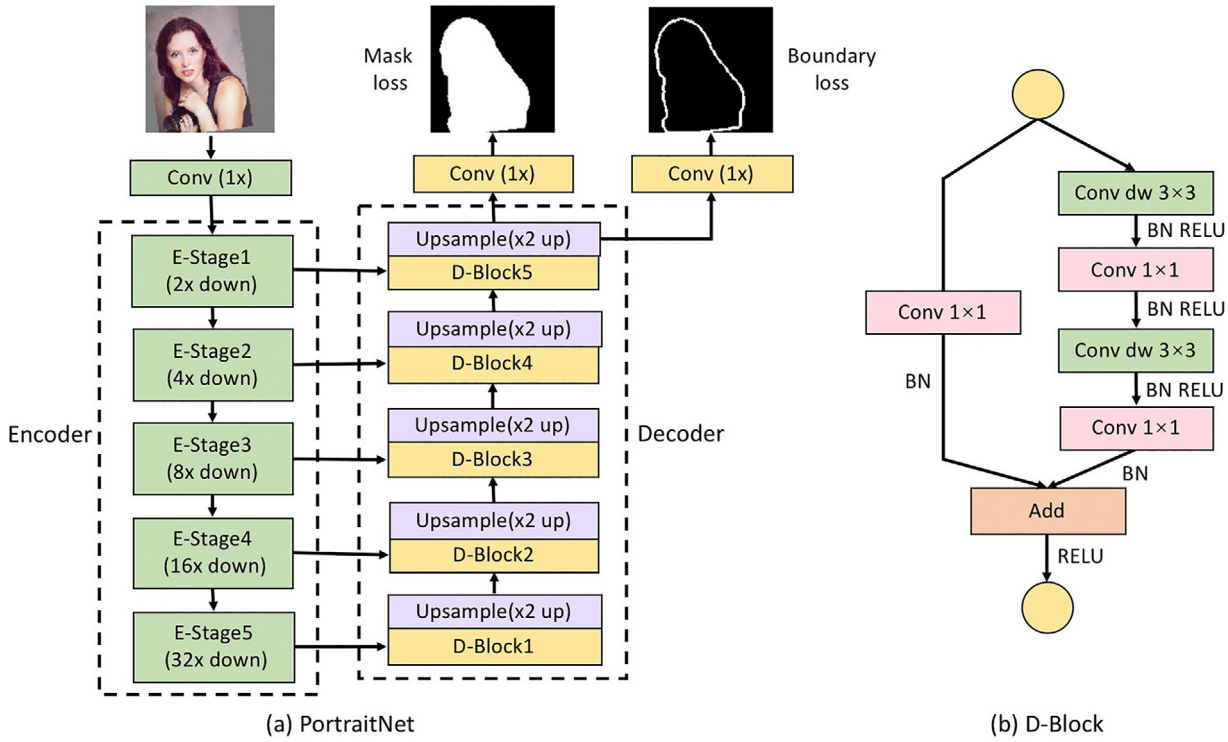

Fig. 2. Overview of PortraitNet. (a) The architecture of PortraitNet. The green blocks represent the encoder module, numbers in brackets represent the down-sampling rates. Each green block represents several convolutional layers. The yellow and purple blocks represent the decoder module. Each up-sampling operation will up-sample the feature maps by $2\times$. (b) The architecture of D-Block in the decoder module. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

up-samples the feature maps by $2\times$. We use modified residual blocks [9] as transition modules in the decoder module. Fig. 2(b) shows the architecture of transition blocks. There are two branches in the block. One branch contains two depthwise separable convolutions. The other contains a single 1×1 convolution to adjust the number of channels.

In PortraitNet, we utilize MobileNet-v2 [19] as backbone in the encoder module. And we use massive depthwise convolutions in PortraitNet to get a higher running speed, which makes the model suitable for mobile device.

3.2. Auxiliary losses

In order to improve the running speed of the model, PortraitNet uses depthwise separable convolution layers to extract features and up-sample the feature maps subsequently. As a lightweight segmentation model, the precision declines compared with sophisticated models. Therefore, we propose to add two effective auxiliary losses during the training process, which helps to improve the performance without causing extra cost for inferring results.

Boundary loss. Compared with general object segmentation, portrait segmentation is more sensitive to the segmentation boundaries. The network needs to generate sharper boundaries in favor of applications such as background replacement or blurring. To utilize the useful information contained in semantic boundaries, we propose to add a semantic boundary loss in addition to the original semantic segmentation loss. We slightly change the last layer in the decoder module by adding a new convolution layer in parallel to generate boundary detection maps, as illustrated in Fig. 2(b). On the other hand, the boundary convolutional layer will not be used for segmentation inference. Different from [17], we only employ one convolutional layer for boundary prediction instead of adding a boundary branch. We use the boundary auxiliary loss to further learn the pivotal features of the portrait images, such that the learned feature can be effectively used for inferring better segmentation.

We generate boundary ground truth from manual labeled mask ground truth using traditional boundary detection algorithm such as Canny [25]. In order to reduce the difficulty of learning boundaries, we set the width as 4 for 224×224 input images (see Fig. 3). Since more than 90% of pixels in the boundary ground truth images are negative, the representation of boundary is difficult to learn. We therefore use focal loss [26] to guide the learning of boundary masks. The overall loss L is:

$$L_m = -\sum_{i=1}^{n}(y_i\log(p_i) + (1-y_i)\log(1-p_i)) \tag{1}$$

$$L_e = -\sum_{i=1}^{n}((1-p_i)^{\gamma}y_i\log(p_i) + p_i^{\gamma}(1-y_i)\log(1-p_i)) \tag{2}$$

$$L = L_m + \lambda \times L_e \tag{3}$$

In the above, L_m is the cross-entropy loss and L_e is the focal loss. λ is the weight of boundary loss. y_i represents the ground truth label of pixel i. p_i represents the predicted probability of pixel i. The predicted probability p_i in Eqs. (1) and (2) is computed as:

$$p_i(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \tag{4}$$

where z is the original output of PortraitNet, and K is the number of groundtruth classes.

As only one convolutional layer is used to generate boundary mask, the mask features and boundary features could make invalid

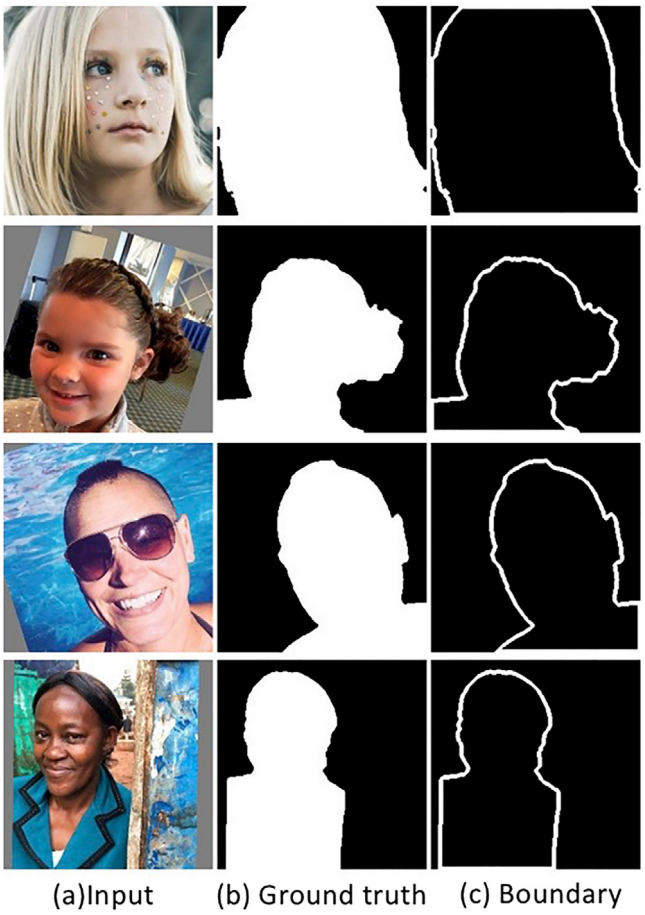

Fig. 3. The ground truth boundary generated by Canny operator. (a) Original image. (b) The corresponding segmentation. (c) The ground truth boundary.

competition in the feature representations. To avoid this, small λ should be set. The boundary loss can improve the sensitivity of the model to the portrait boundary, which in turn improves the segmentation accuracy.

Consistency constraint loss. It is straightforward to use the ground truth semantic segmentation as the supervision signal, where the portrait pixels in the image are manually labeled as 1, otherwise 0. Such labels are usually called hard labels, because they only have binary categories. However, it has been proved that soft labels with more information can further benefit the model training. There are some research works focusing on using soft labels to improve the accuracy of tiny models through model distillation [23,27]. For the input images, they use a well-trained huge teacher model to generate soft labels, and exploit the soft labels to supervise the training of tiny student model. Model distillation requires a tedious training process, and the amount of data may not be sufficient to train a huge teacher model. Compared with complicated model distillation, we propose a novel method to generate soft labels using the tiny network itself with data augmentation. We also use consistency constraint loss to assist the model training, as shown in Fig.4.

Commonly used data augmentation includes two main categories. One is deformation enhancement, such as random rotate, flip, scale, crop, etc. The other is texture enhancement, such as changing the brightness, contrast, sharpness of images, adding random noise or Gaussian filtering, etc. For an original image, we firstly use deformation enhancements to generate image A, and then apply texture enhancement on image A to generate A'. Texture enhancement will not change the shape of the images, so the

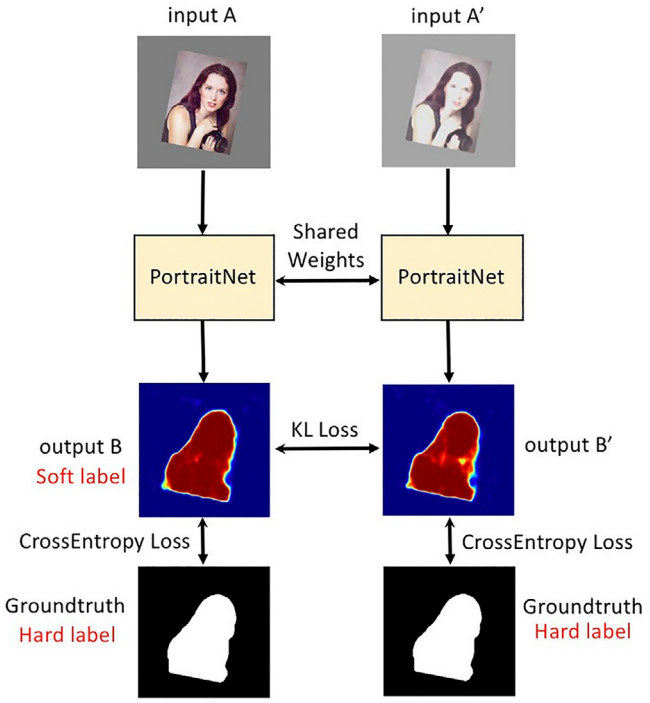

Fig. 4. Illustration of consistency constraint loss.

segmentation of image A and A' are the same. Suppose the network output of image A is heatmap B and the output of image A' is heatmap B', then heatmap B and B' should be the same theoretically. However, due to the texture augmentation methods, the quality of image A' is worse than A. As a result, the generated B' is worse than B. Hence we use the heatmap B with higher quality as the soft labels for heatmap B'. Specifically, we add a consistency constraint loss between heatmap B and B', which is formulated as a KL divergence:

$$L'_m = -\sum_{i=1}^{n}(y_i \log(p_i) + (1-y_i)\log(1-p_i)) - \sum_{i=1}^{n}(y_i \log(p'_i) + (1-y_i)\log(1-p'_i)) \tag{5}$$

$$L_c = \frac{1}{n}\sum_{i=1}^{n} q_i \times \log\frac{q_i}{q'_i} \times T^2 \tag{6}$$

$$L = L'_m + \alpha \times L_c \tag{7}$$

Here α is used to balance the two losses. T is used to smooth the outputs. p_i and p'_i in Eq. (5) are defined as follows:

$$p_i(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad p'_i(z'_j) = \frac{e^{z'_j}}{\sum_{k=1}^{K} e^{z'_k}} \tag{8}$$

And q_i and q'_i in Eq. (6) are defined similarly:

$$q_i(z_j) = \frac{e^{\frac{z_j}{T}}}{\sum_{k=1}^{K} e^{\frac{z_k}{T}}} \qquad q'_i(z'_j) = \frac{e^{\frac{z'_j}{T}}}{\sum_{k=1}^{K} e^{\frac{z'_k}{T}}} \tag{9}$$

Fig. 5. Sample portrait images in EG1800.

Fig. 6. Sample portrait images in Supervise-Portrait.

The consistency constraint loss could further improve the accuracy of the model, and enhance its robustness under different illumination conditions (Fig. 4).

4. Experiments

In this section, we first introduce the datasets and experimental setup, then evaluate the performance of PortraitNet and the effectiveness of the two auxiliary losses.

4.1. Dataset

We train and test our method on two well-known portrait segmentation datasets: EG1800 [7] and Supervise-Portrait.

EG1800: EG1800 contains 1800 portrait images collected from Flickr, and each image is manually labeled at pixel level. The images are mainly self-portrait captured by the front camera of a mobile phone. The final images in EG1800 are scaled and cropped automatically to 800×600 according to the bounding box generated by the face detector running on each image. The 1800 images are divided into two groups. One is the training dataset with 1500 images, while the other is the validating/testing dataset with 300 images. Since several image URL links are invalid in the original EG1800 dataset, we finally use 1447 images for training and 289 images for validation. Some sample portrait images are shown in Fig. 5.

Supervise-Portrait: Supervise-Portrait is a portrait segmentation dataset collected from the public human segmentation dataset Supervise.ly [28] using the same data process as EG1800. Supervise.ly dataset contains high-quality annotated person instances. The images are carefully labeled with person segmentation masks. We further run a face detector on the dataset and automatically crop the images according to the face bounding boxes. We discard the images on which face detector failed and finally collect 2258 portrait images with different sizes. We randomly select 1858 images as training dataset and 400 images as validating/testing dataset. We name the resultant dataset as Supervise-Portrait. Compare with EG1800, portrait images in Supervise-Portrait have more complicated background and severe occlusion. Some sample portrait images are shown in Fig. 6.

4.2. Data augmentation

To improve the generality of the trained model, we use several data augmentation methods to supplement the original training dataset, leading to better segmentation results. These data augmentation methods can be divided into two categories: one is deformation augmentation, the other is texture augmentation. Deformation augmentation augments the position or size of the target, but will not affect the texture. On the other hand, texture augmentation complements the texture information of the target while keeping the position and size.

The deformation augmentation methods used in our experiments include:

- random horizontal flip
- random rotation {−45°–45°}
- random resizing {0.5–1.5}
- random translation {−0.25–0.25}

The texture augmentation methods used in our experiments include:

- random noise {Gaussian noise, $\sigma = 10$ }
- image blur {kernel size is 3 and 5 randomly }
- random color change {0.4–1.7}
- random brightness change {0.4–1.7}
- random contrast change {0.6–1.5}
- random sharpness change {0.8–1.3}

Every operation in deformation augmentation and texture augmentation added up with the probability of 0.5 during training. After data augmentation, we normalize the input images before training using image mean ([103.94, 116.78, 123.68], BGR order) and image val (0.017). The normalization equation is $(image - mean) \times val$. Fig. 7 shows the data augmentation methods used in the experiments.

4.3. Experimental setup

We implement our model using the Pytorch framework [29]. All competitive models are trained using a single NVIDIA 1080Ti graphics card. We use Adam algorithm with batchsize 64 and weight decay 5e−4 during training. The initial learning rate is 0.001. We use $(lr \times 0.95^{\frac{epoch}{20}})$ to adjust the learning rate with 2000

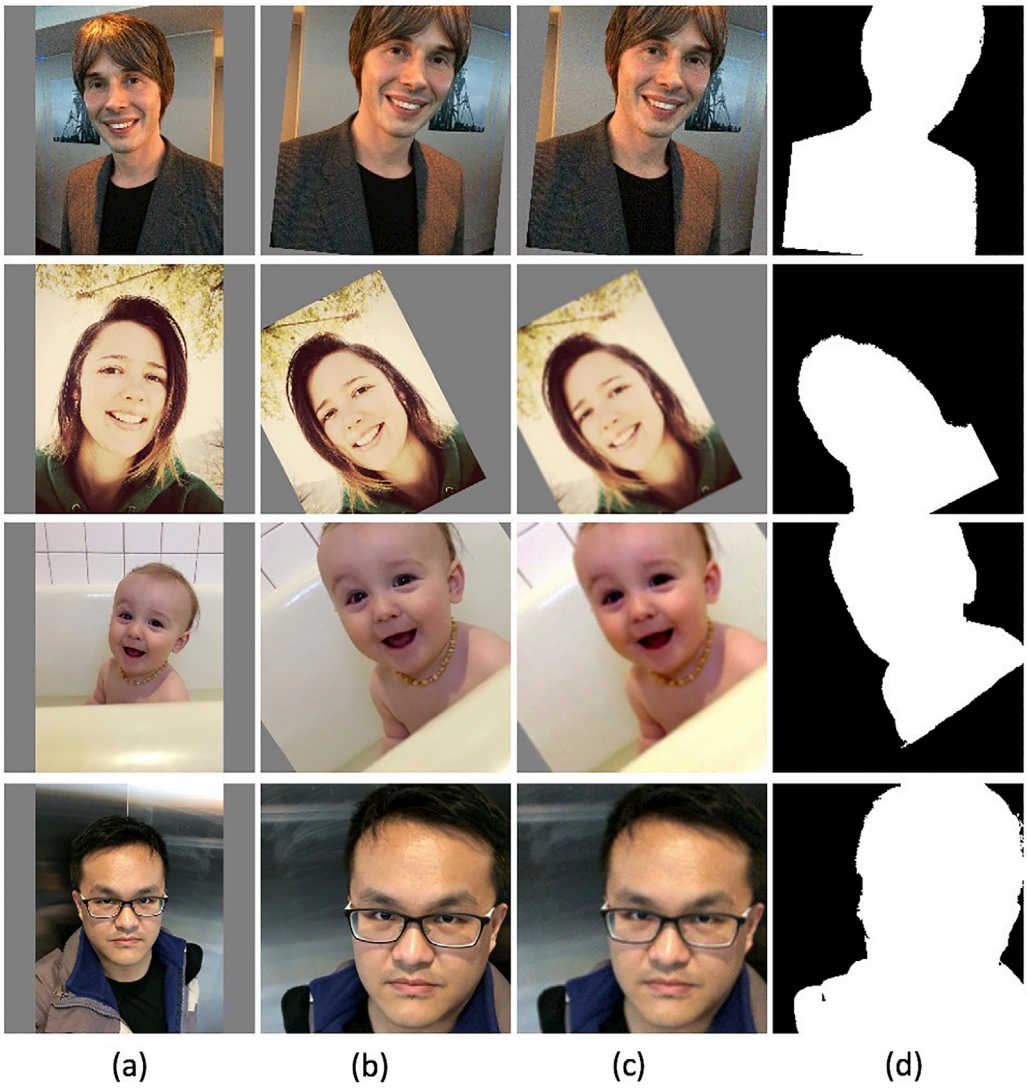

Fig. 7. Data augmentations used in PortraitNet. (a) Original images. (b) Result images after adding deformation augmentations on (a). (c) Result images after adding texture augmentations on (b). (d) The ground truth segmentation corresponding to (b) and (c).

epochs. In order to achieve higher running speed, we train and test our model on 224×224 RGB images with three channels.

4.4. Ablation study

In this sub-section, we carefully evaluate the performance of PortraitNet. First, we validate the effectiveness of two auxiliary losses respectively. Next, we demonstrate the performance of PortraitNet by comparing with the state-of-the-art methods.

4.4.1. Boundary loss

The boundary accuracy greatly influences the user experiences in the applications of portrait segmentation such as background blurring or replacement. We add a new convolutional layer paralleled with mask prediction layer to predict the portrait boundary. In order to verify the effectiveness of boundary loss, we conduct two experiments with or without boundary loss. More specifically, we train PortraitNet-M with only segmentation loss, and train PortraitNet-B with both segmentation loss and boundary loss. We empirically set $\lambda = 0.1$ to balance segmentation loss and boundary loss. We do not use α in focal loss and set $\gamma = 2$ in Eq. (2). We initialize PortraitNet-B with well-trained PortraitNet-M. All hyper-parameters of the experimental setup are same in different experiments.

Table 1
Accuracy comparison of PortraitNet with different losses.

Method	EG1800 (%)	Supervise.ly (%)
PortraitNet-M(ours, Exp.1)	96.32	92.63
PortraitNet-B(ours, Exp.2)	96.54	93.04
PortraitNet-C(ours, Exp.3)	96.57	93.17
PortraitNet(ours, Exp.4)	**96.62**	**93.43**

The quantitative metric used to evaluate segmentation precision is the mean Interaction-over-Union(IOU) as follows:

$$mean\ IOU = \frac{1}{N} \times \sum_{i=1}^{N} \frac{maskPD_i \cap maskGT_i}{maskPD_i \cup maskGT_i}, \tag{10}$$

where $maskPD_i$ and $maskGT_i$ represent segmentation result and ground truth label of ith image of test dataset, respectively. The quantitative comparison is shown in Table 1. It can be seen that the boundary loss improves the IOU accuracy by 0.22% (from 96.32% to 96.54%) on EG1800 dataset, and by 0.41% (from 92.63% to 93.04%) on Supervise-Portrait dataset.

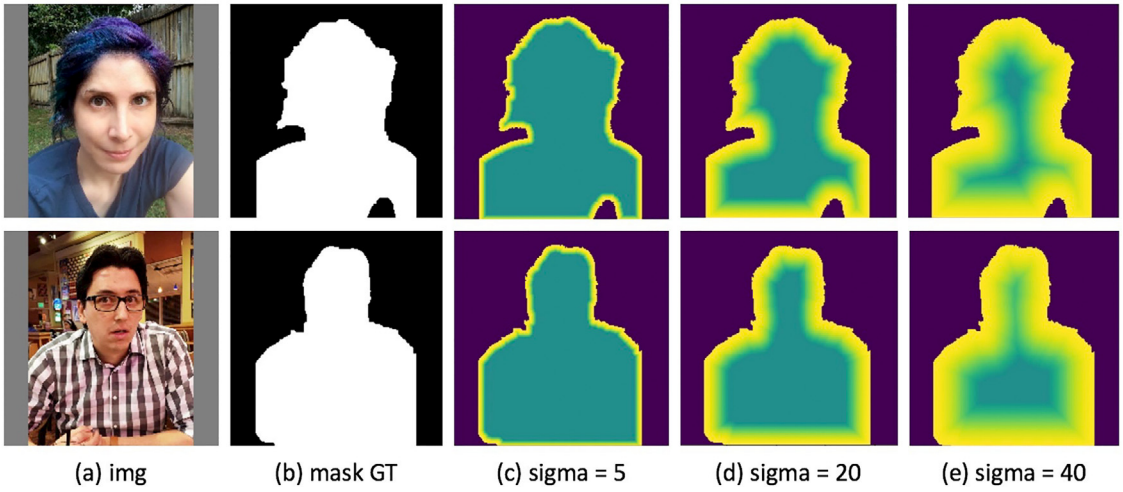

Fig. 8. New metric for evaluating boundary precision. (a) Original images. (b) The corresponding segmentation. (c)–(e) The weight masks under the new metric with different σ in Eq. (12).

Table 2
Accuracy comparison of PortraitNet with new mean IOU metric on EG1800 test dataset.

Sigma	Single model (%)	Edge model (%)	Increase (%)
3	91.56	92.34	+0.78
5	93.80	94.44	+0.64
10	96.11	96.55	+0.44
20	97.57	97.86	+0.29
40	98.23	98.44	+0.21
80	98.43	98.63	+0.20

We also propose a specific metric to better evaluate the model performance on portrait boundary than mean IOU. The new metric is similar to mean IOU with emphasized weight on boundaries over inner pixels:

$$mean\ edge\ IOU = \frac{1}{N} \times \sum_{i=1}^{N} \frac{w(x)_i (maskPD_i \cap maskGT_i)}{w(x)_i (maskPD_i \cup maskGT_i)}, \tag{11}$$

where $w(x)_i$ represents the weight of pixel x in the ith image. More specifically, the weight $w(x)_i$ declines continuously from boundaries to inside as in the following equation:

$$w(x)_i = \begin{cases} e^{-\frac{dis(x)^2}{2\sigma^2}}, & x \in maskGT\ and\ y(x) = 1 \\ 0, & x \in maskGT\ and\ y(x) = 0 \end{cases} \tag{12}$$

where $dis(x)$ represents the distance from pixel x to portrait boundary, and σ indicates the decline rate. An illustration of the new metric with different σ is shown in Fig. 8. Based on the new metric, we compare the performance of the two networks with different σ on EG1800 dataset, the results are shown in Fig. 9 and Table 2. It can be seen that that the performance enhancement is larger when σ is smaller, since the metric emphasize more on boundary pixels when σ is small. This demonstrates the effectiveness of boundary loss in improving the precision of segmentation boundaries.

4.4.2. Consistency constraint loss

Due to the complication of lighting conditions when taking selfies on smartphone, we use texture augmentation methods to improve the robustness of PortraitNet. Meanwhile, we also find that soft label could further improve the segmentation precision with the help of consistency constraint loss. We conduct a contrast test to verify its effectiveness. PortraitNet-M is the same as discussed before, we train PortraitNet-C with both segmentation loss and consistency constraint loss. We set $\alpha = 2$ to balance the two losses, and T is set to 1. The evaluation with mean IOU on EG1800 dataset and Supervise-Portrait dataset is reported in Table 1. Fig. 10 shows some segmentation results generated by PortraitNet-M and PortraitNet-C, respectively.

To validate our proposed model, we perform data augmentation on the test dataset to capture illumination variations. We evaluate the two networks with mean IOU metric on the new test dataset

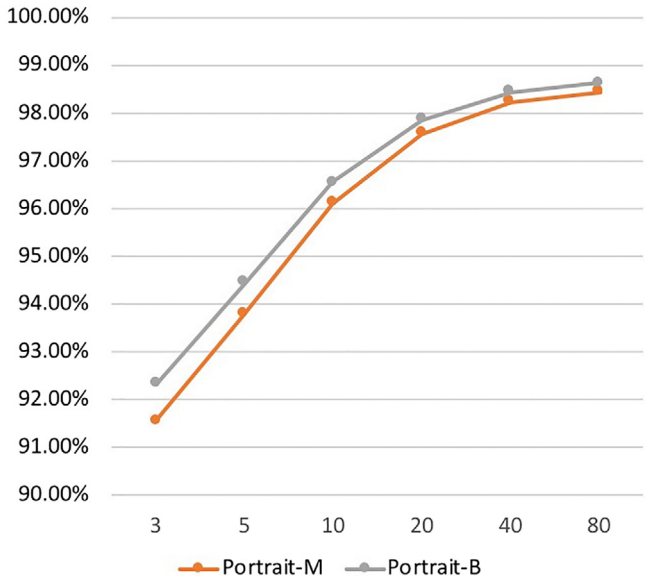

Fig. 9. The precision comparison in mean edge IOU metric between Portrait-M and Portrait-B. Portrait-M trained model only with mask loss, Portrait-M model trained with mask loss and boundary loss, parameter $\lambda = 0.1$.

Table 3
Accuracy comparison of PortraitNet-M and PortraitNet-C with new EG1800 test dataset using the mean IOU metric.

Method	EG1800 (%)	Supervise.ly (%)
PortraitNet-M(ours, Exp.1)	95.78	91.51
PortraitNet-C(ours, Exp.3)	96.24	91.98

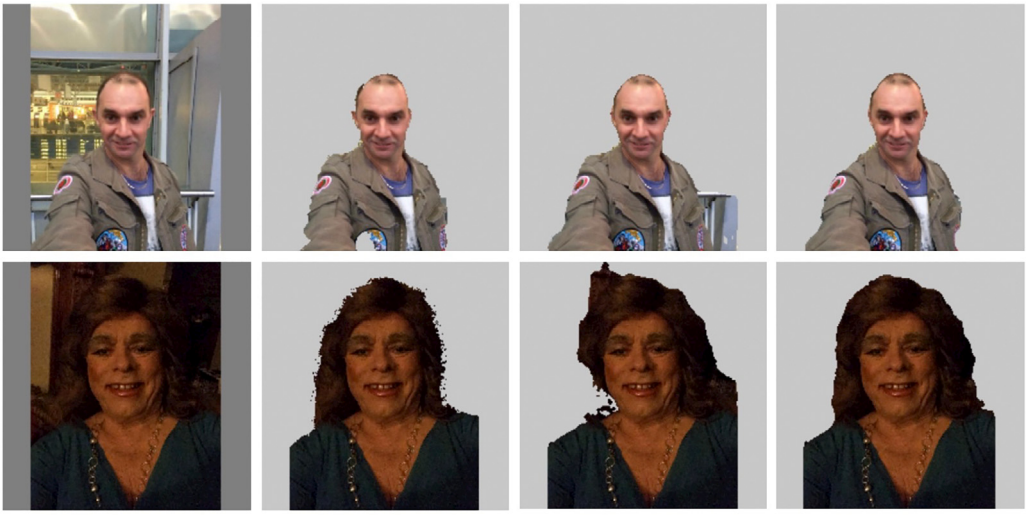

Fig. 10. Results generated by PortraitNet-M and PortraitNet-C.

Table 4
Accuracy comparison with the state-of-the-art real-time segmentation methods using the mean IOU metric. The numbers in brackets represent the image size for inference. Horizontal flip or image resizing are not used in testing.

Method	EG1800 (%)	Supervise.ly (%)
PortraitFCN + (800 × 600)	95.91	92.78
ENet(224 × 224)	96.00	92.38
BiSeNet(448 × 448)	95.79	92.56
BiSeNet(224 × 224)	95.25	91.25
BiSeNet + (224 × 224)	95.55	91.76
PortraitNet(ours, 224 × 224)	**96.62**	**93.43**

Table 5
Quantitative performance comparison. FLOPs are estimated with the size in brackets.

Method	FLOPS (G)	Parameters (M)
PortraitFCN + (6 × 224 × 224)	62.89	134.27
ENet(3 × 224 × 224)	0.44	0.36
BiSeNet(3 × 448 × 448)	9.52	12.4
BiSeNet(3 × 224 × 224)	2.38	12.4
PortraitNet(ours, 3 × 224 × 224)	0.51	2.1

Table 6
Speed comparison with the state-of-the-art real-time segmentation models on NVIDIA 1080Ti graphic card.

Method	NVIDIA 1080Ti (ms)
PortraitFCN + (6 × 224 × 224)	19.04
ENet (3 × 224 × 224)	12.53
BiSeNet (3 × 448 × 448)	5.15
BiSeNet (3 × 224 × 224)	3.11
PortraitNet(ours, 3 × 224 × 224)	4.92

as in Table 3. The model with consistency constraint loss is more robust to illumination condition change.

4.4.3. Accuracy analysis

PortraitNet is specifically designed for mobile device compared with other real-time segmentation networks. We choose PortraitFCN+ [7], ENet [13] and BiSeNet [15] as baselines, since PortraitFCN+ [7] is one of the iconic methods for portrait segmentation, ENet [13] and BiSeNet [15] are the state-of-the-art. In our experiments, the backbone of BiSeNet is ResNet18.

For real-time inference on mobile device, we use MobileNet-v2 [19] as our backbone to extract features form original images, and we use U-shape architecture to generate sharp segmentation boundaries. Depthwise separable convolutions are used in PortraitNet to gain running speed. In encoder modules, the down-sampling rate is $32\times$. We use large receptive field to utilize global information to help deduce the segmentation mask, which is necessary for portrait images. In decoder modules, we use skip lines from encoder modules to reconstruct the spatial information for better segmentation details. To employ segmentation networks on mobile device, we set the input image size of 224×224 for real-time inference. We train PortraitNet model with mask loss and two auxiliary losses as the following:

$$L = L'_m + \alpha \times L_c + \beta \times L_e, \tag{13}$$

where L'_m, L_c, L_e are defined in Eq. (5), Eq. (6), Eq. (2) respectively, and $\alpha = 2,\ \beta = 0.3,\ T = 1$.

The performance on EG1800 and Supervise-Portrait datasets is shown in Table 4. To further verify the performance of the two auxiliary losses, we test a new model called BiSeNet+, which is BiSeNet with our two auxiliary losses. The experiments show that the two auxiliary losses also improve the result of BiSeNet. Fig. 11 shows several difficult portrait segmentation results generated by different methods.

4.4.4. Speed analysis

Inference efficiency is crucial for portrait segmentation on mobile device. We evaluate the FLOPs (float point operations) and the scale of parameters on different models (see Table 5). We also test the actual running speed on NVIDIA 1080Ti graphic card compared with other methods (see Table 6). For fair comparison, we use bilinear interpolation based up-sampling instead of de-convolution in PortraitNet. We find that PortraitNet achieves a good balance between accuracy and efficiency. Moreover, we adapt PortraitNet from Pytorch [29] to Coreml [30] and test the inferring time on IOS. For image size of 224×224, the cost of PortraitNet processing one image is around 32 ms, while other real-time segmentation methods cannot directly run on iPhone without modification.

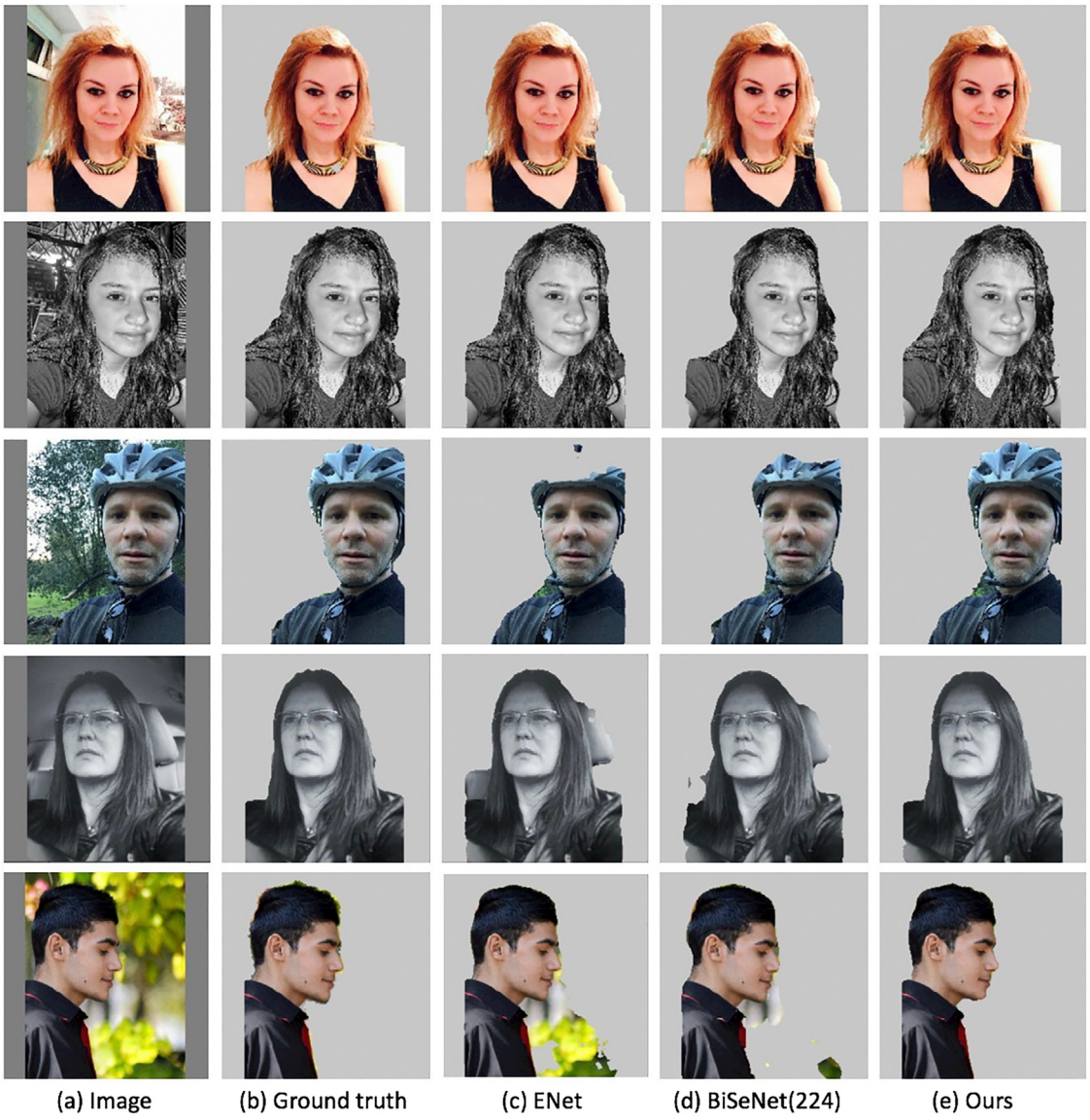

Fig. 11. Segmentation results of challenging portrait images generated by different methods. The first row shows images with strong illumination. The second and fourth rows show images with background color close to foreground portrait. The third row shows the portrait image with helmet. The last row shows the portrait from a side view.

5. Conclusion

In this paper, we present PortraitNet, a specifically designed lightweight model for segmenting portrait images on mobile device. We propose to add two auxiliary losses to assist training without additional cost for segmentation inference. The boundary loss helps to generate sharper boundaries, and the consistent constraint loss improves the robustness with respect to lighting variations. The experimental results demonstrate both high accuracy and efficiency of our approach, verifying that PortraitNet could serve as a lightweight tool for real-time portrait segmentation on mobile device.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Technology R&D Program (2016YFB1001402), the Natural Science Foundation of China (61772298 and 61832016), Research Grant of Beijing Higher Institution Engineering Research Center and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–40.
[2] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 2017;39(12):2481–95.
[3] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015. p. 234–41.
[4] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv e-prints 2014.

[5] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 2018;40(4):834–48.
[6] Chen L-C, Papandreou G, Schroff F, Adam H. Rethinking atrous convolution for semantic image segmentation. arXiv e-prints 2017.
[7] Shen X, Hertzmann A, Jia J, Paris S, Price B, Shechtman E, et al. Automatic portrait segmentation for image stylization. In: Computer Graphics Forum, 35. Wiley Online Library; 2016. p. 93–102.
[8] Du X, Wang X, Li D, Zhu J, Tasci S, Upright C, et al. Boundary-sensitive network for portrait segmentation. arXiv e-prints 2017.
[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
[10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097–105.
[11] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv e-prints 2014.
[12] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1–9.
[13] Paszke A, Chaurasia A, Kim S, Culurciello E. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:160602147 2016.
[14] Zhao H, Qi X, Shen X, Shi J, Jia J. Icnet for real-time semantic segmentation on high-resolution images. arXiv e-prints 2017.
[15] Yu C, Wang J, Peng C, Gao C, Yu G, Sang N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: European Conference on Computer Vision. Springer; 2018a. p. 334–49.
[16] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, et al. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 3213–23.
[17] Yu C, Wang J, Peng C, Gao C, Yu G, Sang N. Learning a discriminative feature network for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018b. p. 1857–66.
[18] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv e-prints 2017.
[19] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–20.
[20] Hluchyj MG, Karol MJ. Shuffle net: An application of generalized perfect shuffles to multihop lightwave networks. Journal of Lightwave Technology 1991;9(10):1386–97.
[21] Ma N, Zhang X, Zheng H-T, Sun J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 116–31.
[22] Zheng S, Song Y, Leung T, Goodfellow I. Improving the robustness of deep neural networks via stability training. In: Proceedings of the ieee conference on computer vision and pattern recognition; 2016. p. 4480–8.
[23] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv e-prints 2015.
[24] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv e-prints 2015.
[25] Canny J. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence 1986(6):679–98.
[26] Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2980–8.
[27] Anil R, Pereyra G, Passos A, Ormandi R, Dahl GE, Hinton GE. Large scale distributed neural network training through online distillation. arXiv e-prints 2018.
[28] Supervise.ly. https://supervise.ly/; 2017.
[29] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in pytorch 2017.
[30] Coreml. https://developer.apple.com/documentation/coreml; 2017.