Efficient Models for Real-time Person Segmentation on Mobile Phones

Julian Strohmayer
Computer Vision Lab
TU Wien
Vienna, Austria
jstrohmayer@cvl.tuwien.ac.at

Jakob Knapp
Computer Vision Lab
TU Wien
Vienna, Austria
jknapp@cvl.tuwien.ac.at

Martin Kampel
Computer Vision Lab
TU Wien
Vienna, Austria
martin.kampel@tuwien.ac.at

***Abstract*—Despite constantly evolving mobile hardware, real-time person segmentation on mobile phones is challenging due to the limited computational resources. To address this problem, we introduce a novel UNet-like network architecture based on MobileNetV3, which enables the segmentation of persons in images and videos on mobile phones. Our model, which is not limited to a specific shot type, outperforms specialized models in their respective domains and runs with 35 fps on a Google Pixel 4 mobile phone. Moreover, we demonstrate how the segmentation accuracy can be further improved by exploiting the temporal coherence of consecutive frames in videos.**

***Index Terms*—real time, person segmentation, mobile phone**

I. Introduction

The segmentation of persons in images or videos captured on mobile phones is a central task in many person-centric mobile applications. Examples include background removal or replacment [1], hair recoloring [2], virtual makeup [3], hand gesture recognition [4] or portrait stylization [5]. Although current mobile phones feature powerful computing hardware, the acquisition of high quality segmentation masks in real time is still challenging. Without increasing the computational resources, the only way to address this problem is the development of efficient person segmentation models that can cope with resource-constrained environments. In search of such a model, we evaluate 5 state-of-the-art network architectures with regard to their suitability for real-time segmentation of persons on mobile phones. Relevant components from candidate architectures are selected, new network architectures are developed and corresponding person segmentation models are trained and evaluated on both custom and publicly available datasets. Based on this evaluation, we propose a novel UNet-like network architecture that uses MobileNetV3 building blocks. The resulting model achieves state-of-the-art segmentation accuracy on portrait shots and compares favorably to much larger models specialized in full body shots, while not being limited to a specific shot type. Moreover, the model achieves an average inference time of 27.9 ms (>35 fps) on a Google Pixel 4 mobile phone.

The remainder of this work is structured as follows. In Section II, we discuss the state of the art regarding person segmentation and mobile network architectures. In Section III, potential encoder architectures are compared and we propose new network architectures for person segmentation on mobile phones. The composition of our datasets and model training is discussed in Section IV. Finally, quantitative and qualitative evaluation results are given in Section V.

II. Related Work

In recent years, network architectures and models for person segmentation that focus on a specific shot type (e.g. portait or full-body shot) have been proposed. In [6], Shen et al. present their network architecture for portrait segmentation called PortraitFCN+ and the EG1800 dataset, which is frequently used for comparing portrait segmentation models. To our knowledge, the best results on the EG1800 benchmark to date are achieved by Wadhwa et al. [7]. PortraitNet [8], which is a UNet-like [9] network architecture with a MobileNetV2 [10] backbone, achieves comparable results on the same benchmark. Small portrait segmentation models like SINet+ [11] and HLB [12] with less than one million parameters achieve frame rates of over 30 fps on current mobile phones, while sacrificing some accuracy. Even lower latencies can be achieved with network architectures that are specifically optimised for mobile environments. These include the MobileNet family [15], ShuffleNet [16], GhostNet [17], MNas-Net [18] or EfficientNet [19]. With inference times well below 50 ms, these network architectures are ideal candidates for the development of efficient person segmentation models for mobile phones. In the context of person segmentation, BowtieNet [14] should also be mentioned, which currently achieves the highest segmentation accuracy on the Baidu people segmentation dataset, consisting mainly of full body images. With a framerate of 39 fps on an Nvidia Titan X GPU, this model is not suitable for mobile environments, but we can still compare its segmentation accuracy to that of our own models.

III. Network Architecture

The proposed network architecture follows the encoder-decoder principle that was popularized by UNet [9] and has since been proven effective for various segmentation problems [20]. The encoder-decoder architecture consist of two stages, a contracting path (encoder) and an expanding path (decoder).

TABLE I
PERFORMANCE METRICS OF POTENTIAL ENCODER ARCHITECTURES.
*INFERENCE TIME ON GOOGLE PIXEL 4 (4 CPU CORES)

Network Architecture	params	top-1	ms*	S
MobileNetV3 large	5.4M	75.2	32	**0.66**
MobileNetV3 small	2.5M	67.4	13	**0.67**
ShuffleNetV2 1.0x	2.3M	69.4	17	**0.68**
ShuffleNetV2 0.5x	1.4M	60.3	10	0.50
MnasNet-A1	3.9M	75.2	40	0.55
EfficientNet-B0	5.3M	77.3	49	0.50
EfficientNet-Lite0	4.7M	75.1	45	0.49
GhostNet 1.0	5.2M	73.9	33	**0.61**
GhostNet 0.5	2.6M	66.2	21	0.53

The encoder extracts feature maps of decreasing size through consecutive convolution and pooling operations up to a sufficiently small feature map resolution. The decoder then brings the output of the encoder back to the original input resolution by concatenating encoder and decoder feature maps via skip connections and repeatedly applying transposed convolutions (or a combination of interpolation and convolution operations). Because encoder and decoder are, apart from skip connections, self-contained components that can be exchanged, the encoder-decoder architecture is an ideal evaluation platform.

Encoders for mobile phones have to cope with prevailing limitations regarding memory capacity and processing power, while still being able to run in real time. State-of-the-art encoder architectures, as listed in Table I, achieve this by using efficient convolution blocks. In order to select potential encoders for our own network architecture, encoders are ranked according to their top-1 ImageNet classification accuracies and their average inference times on a Google Pixel 4 mobile phone. From these two measurements, the efficiency score $S \in [0, 1]$ is calculated as in Equation 1, where $\widehat{top\text{-}1}$ and $\widehat{ms}$ are the top-1 classification accuracies and inference times, given in Table I, rescaled to $[0, 1]$.

$$S = (\widehat{top\text{-}1} + (1 - \widehat{ms})) * 0.5 \tag{1}$$

As given in Table I, the three top performing encoder architectures, with respect to S, are ShuffleNetV2 1.0x, MobileNetV3 large and MobileNetV3 small, with values of 0.68, 0.67 and 0.66, respectively. Since both MobileNetV3 variants share the same encoder architecture, we also include GhostNet 1.0, achieving an efficiency score of 0.61, in our evaluation.

To carry the efficiency of the encoders over to the full network architecture, we use their respective characteristic bottleneck blocks to build a matching decoder. These are inverted residual blocks with squeeze-and-exitation (MobileNetV3), ShuffleNetV2 units, and Ghost bottlenecks (GhostNet). Encoder feature maps are concatenated with decoder feature maps of the same level via skip connections, fed to the bottleneck block and up-scaled bilinearly to match the resolution of the skip connection in the subsequent level. By using bilinear interpolation instead of transposed convolution, we avoid checkerboard artifacts and keep the number of parameters of the decoder low. Furthermore, because convolutions on small

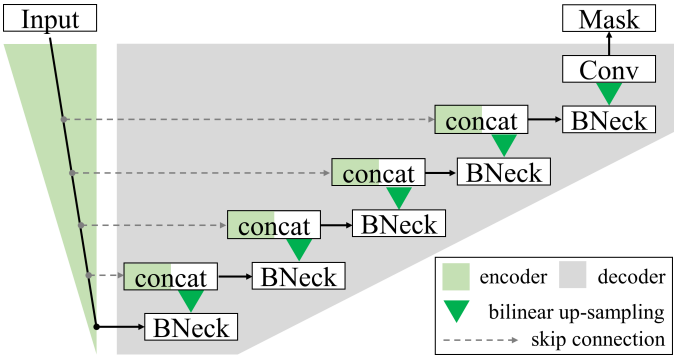


Fig. 1. Generalized network architecture, showing the encoder-decoder-structure and the use of the characteristic bottleneck blocks in the decoder.

feature maps are cheaper and bilinear interpolations do not add new information, we place the bilinear interpolations after the bottleneck blocks. The generalized network architecture, upon which all our person segmentation network architectures are built, is visualized in Figure 1, which shows the encoder-decoder structure and the placement of the characteristic bottleneck blocks in the decoder. A detailed description of the network architectures presented can be found in the supplementary material[1]. Four person segmentation models are trained and evaluated, called MobSegS, MobSegL, ShuSegL, and GhoSegL in the following. The naming convention is based on the different encoders, which are MobileNetV3 small, MobileNetV3 large, ShuffleNetV2 1.0x and GhostNet 1.0 respectively.

IV. DATA AND TRAINING

Both image and video data were considered for training our person segmentation models. However, the availability of suitable video datasets with dense annotation and sufficient image quality is limited [21]. For this reason, we decide to train on image-mask pairs exclusively, as there is an abundance of such data available online. For the evaluation, however, both image and video datasets are utilized. Links to all the datasets used are given in Table VI.

A. Data

We construct a dataset from eight image collections containing people in different poses and environments. Besides semantic segmentation, they stem from domains such as natural image matting, human parsing, or pose estimation. Apart from portrait shots, half and full body shots are included as well, which facilitates the generalization to a wide spectrum of shot types. A complete list of datasets used, with the respective number of images and contained shot types, is given in Table II. The combined image dataset contains a total of 84,826 image-mask pairs, composed of portrait, full body and other shots with a ratio of 2:2:1.

B. Video Data

In addition to single image inference, we exploit the temporal coherence of consecutive frames. For this, the generalized

TABLE II
LIST OF IMAGE DATASETS USED, WITH THE NUMBER OF IMAGES AND THE CONTAINED SHOT TYPES PORTRAIT (P), HALF-BODY (HB), AND FULL-BODY (FB).

Dataset	Images	Shot Types
Human Matting Dataset	34,426	P
Baidu People Segmentation	5,387	FB
Dark Complexion Portrait Segmentation	12,165	P, HB, FB
ICCV15 Human Parsing Dataset [22], [23]	17,706	FB
UTP - Leeds Sports Pose [24]	2,000	FB
UTP - Leeds Sports Pose Ext. [24]	8,642	FB
PicsArt Hackathon Dataset	2,500	P, HB, FB
Deep Automatic Portrait Matting [25]	2,000	P
Total	84,826	

TABLE III
LIST OF VIDEO DATASETS USED, WITH THE CORRESPONDING NUMBER OF FRAMES, NUMBER OF MASKS AND THE ANNOTATION TYPE.

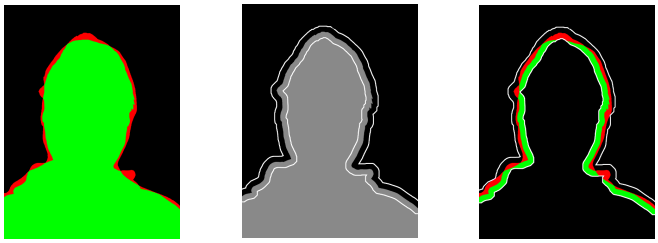
Dataset	Frames	Masks	Annotation Type
SegTrack v2 [26]	53	53	dense
DAVIS 2017 [27]	762	762	dense
LASIESTA [28]	1,794	1,794	dense
VSB100 [29]	525	66	sparse
A2D [30]	2,584	74	sparse
Total	5,718	2,749	

network architecture is modified, such that it accepts an additional fourth input channel, being the predicted mask of the previous frame. It has been shown that this modification can improve the segmentation quality on videos [2]. For comparing the standard models with three input channels (RGB) to the described temporal model, annotated videos are required, which are sourced from the five publicly available video datasets listed in Table III. From these datasets, videos containing people are selected (e.g. DAVIS 2017 parkour). The resulting video test dataset contains 50 different videos with 2,749 annotated frames.

C. Model Training

To prepare training, validation and test datasets, the 84,826 images are split in a 8:1:1 ratio. We use the Adam optimizer (lr=0.01), Binary-Cross-Entropy (BCE) loss, a batch size of 4 and train for 150 epochs. RGB images are re-scaled to 224x224 and normalized using channel means and standard deviations, derived from the training dataset. Data augmentations, such as random translations ($\pm 0.1h$, $\pm 0.1w$), rotations ($\pm 10°$), and scaling ($\pm 0.1h$, $\pm 0.1w$) operations, are applied to RGB image and mask (h and w are image height and width in pixels, respectively).

Apart from the four models that receive RGB images as input (MobSegS, MobSegL, ShuSegL and GhoSegL), we train a second version of MobSegS, called MobSegS+, which is the aforementioned temporal model that takes the previous mask as additional input. As we train on image-mask pairs, we do not have access to this mask. However, it can be simulated by applying transformations to the ground truth mask of the current frame, as described in [2]. For this, random translations ($\pm 0.15h$, $\pm 0.15w$), rotations ($\pm 5°$), scaling ($\pm 0.2h$, $\pm 0.2w$), and shearing ($\pm 5°$) operations are applied after the initial augmentations. Furthermore, we randomly pass an empty previous mask in 75% of cases, which trains the model for first-frame inference where no previous mask is available.



(a) IoU (b) boundary (c) rIoU

Fig. 2. Visual comparison between IoU and rIoU, with TP pixels highlighted in green and FP and FN pixels in red. (a) Shows the standard IoU, (b) the extraction of the boundary region by shrinking and expanding the ground truth mask (grey) and (c) rIoU.

V. EVALUATION

To evaluate the models, we collect both quantitative and qualitative data from the image and video test datasets. Performance metrics for MobSegS, MobSegL, ShuSegL and GhoSegL on the image test dataset are derived and we compare the performance of MobSegS and the temporal model MobSegS+ on the video test dataset to assess whether the exploitation of temporal coherence is beneficial.

A. Metrics

The performance of a segmentation model is determined by measuring the similarity or dissimilarity between the predicted segmentation mask and the corresponding ground truth mask. Standard metrics, such as segmentation accuracy (ACC), intersection over union (IoU), and the Sørensen-Dice coefficient (Dice), which we use in accordance with scientific literature, facilitate this.

Additionally, we introduce the regional intersection over union (rIoU) metric, which, as opposed to global metrics, allows one to place a stronger emphasis on segmentation errors in the border region between foreground and background. If the foreground occupies the majority of the image, as is the case with portraits, for example, these errors are difficult to detect with global metrics such as IoU because their contribution to the total error is so small. We compute rIoU as the IoU of the border region between foreground and background. For this, independent morphological opening and closing operations are applied to the binary ground truth mask. This yields enlarged and shrunken versions, which represent the inner and outer borders of the boundary region, as given in Figure 2 (c). The binary mask of the border region is then acquired by subtracting the shrunken from the enlarged mask. Finally, all pixels that fall within the border region are extracted and taken into account for the IoU computation, which results in the rIoU. The difference between IoU and rIoU is visualized in Figure 2 (a)(c), which shows the global scope of IoU and the local scope of rIoU.

TABLE IV
PERFORMANCE METRICS FOR THE IMAGE TEST DATASET. *INFERENCE TIME ON GOOGLE PIXEL 4 (4 CPU CORES) FOR 224X224 INPUT IMAGES.

Model	ACC	Dice	IoU	rIoU	params	ms*
MobSegL	**98.6**	**97.0**	**94.6**	**86.4**	7.2M	51.4
MobSegS	98.3	96.3	93.5	84.3	**2.4M**	**27.9**
ShuSegL	98.3	96.2	93.5	84.9	2.6M	37.2
GhoSegL	98.6	96.9	94.5	86.3	5.9M	48.2

TABLE V
COMPARISON WITH STATE-OF-THE-ART PORTRAIT AND FULL-BODY SEGMENTATION MODELS. IOU IS MEASURED ON THE EG1800 (PORTRAIT) AND BAIDU PEOPLE SEGMENTATION (FULL-BODY) DATASET. *REAL-TIME PERFORMANCE (>30 FPS) ON A GOOGLE PIXEL 4 MOBILE PHONE.

Model (portrait)	IoU	rt*	Model (full-body)	IoU	rt*
HLB [12]	94.9	✓	DLV2-VGG [14]	91.6	x
SINet+ [11]	95.3	✓	DLV2-ResNet [14]	92.7	x
PortraitNet [8]	96.6	✓	DLV3+ [14]	92.8	x
Wadhwa et al. [7]	97.7	✓	BowtieNet [14]	93.6	x
MobSegS (ours)	97.4	✓	MobSegS (ours)	91.6	✓

B. Results

The performance metrics on the image test dataset are given in Table IV. We observe that the largest model MobSegL outperforms the other models across all metrics, while having the highest inference time. However, when segmenting fast-moving persons, the advantage of a higher frame rate outweighs the advantage of a slightly better segmentation accuracy as soon as the mask lags behind visibly. Furthermore, the perceived segmentation quality is similar across all models. For this reason, we favor models with a small inference time over slower but more accurate models. Regarding accuracy, MobSegS and ShuSegL show similar performance. However, MobSegS is the only model to meet the real-time requirement of 30 fps. From this, we conclude that MobSegS is best suited for person segmentation on mobile phones out of all the models evaluated. Qualitative segmentation results of MobSegS are given in Figure 3, which show the consistent segmentation quality across different shot types. Moreover, we can observe that the model is invariant to rotations, as shown by the backflip example.

Despite being trained for general purpose person segmentation, MobSegS still compares favorably to state-of-the-art segmentation models that focus on a particular shot type, as given in Table V. On the EG1800 dataset [6], MobSegS (97.4% IoU) is second to the model proposed by Wadhwa et al. [7] by only 0.26% IoU. To our knowledge, this is the most accurate real-time portrait segmentation model to date and we achieve comparable performance with one tenth of the training data. Furthermore, MobSegS beats PortraitNet [8] (96.6% IoU) and outperforms compact portrait segmentation models, such as SINet+ [11] (95.3% IoU) and HLB [12] (94.9% IoU), while still operating at more than 30 fps on a Pixel 4 mobile phone. The same holds true for the segmentation of full body shots. In [14], Zhao et al. compare their BowtieNet model to several other segmentation models based on their performance on the Baidu People Segmentation Dataset. Moreover, MobSegS (91.6% IoU) outperforms much larger models, such as DeepLabV2-VGG [14] (91.6% IoU). While BowtieNet is more accurate (93.4% IoU), it processes images of size 256x256 at only 39.1 fps on an Nvidia Titan X GPU. In our experiments, MobSegS achieves speeds beyond 110 fps for images of size 224x224 on the same hardware, which makes it more suitable for use on mobile phones.

Fig. 3. Qualitative comparison between MobSegS prediction and ground truth for a portrait, half body and full body shot example from the Human Matting Dataset, Baidu People Segmentation Dataset and Leeds Sports Pose Dataset, respectively.

To evaluate how the incorporation of additional temporal information affects segmentation accuracy, we compare the performance of MobSegS and MobSegS+ on the video test dataset. Here, MobSegS achieves 75.7% IoU and 70.4% rIoU. MobSegS+ achieves 77.7% IoU and 72.9% rIoU, which is an improvement of 2.0% and 2.5%, respectively. Due to the additional input channel we lose 3.5 fps, which brings the frame rate down to 32.3 fps. The real-time requirement of 30 fps is still exceeded, however. From this we conclude that the incorporation of temporal information improves segmentation accuracy while real-time frame-rates are still achieved.

VI. CONCLUSION

In this work, we have developed and evaluated four efficient network architectures for real-time person segmentation on

mobile phones. From this, a MobileNetV3-based model, MobSegS, emerged which achieved state-of-the-art segmentation accuracy in the EG1800 portrait segmentation benchmark and could compete with much larger models for full-body segmentation in the Baidu segmentation challenge without being limited to a specific shot type. Moreover, experiments with MobSegS+ demonstrated that by adding temporal information, one can further improve the segmentation accuracy by 2% IoU. With frame rates of 35.8 fps and 32.3 fps, respectively, on a Google Pixel 4 mobile phone, both MobSegS and MobSegS+ are well suited for mobile applications, relying on real-time person segmentation in images or videos.

REFERENCES

[1] B. Zhu, Y. Chen, J. Wang, S. Liu, B. Zhang, and M. Tang, "Fast deep matting for portrait animation on mobile phone," *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

[2] A. Tkachenka, G. Karpiak, A. Vakunov, Y. Kartynnik, A. Ablavatski, V. Bazarevsky, and S. Pisarchyk, "Real-time hair segmentation and recoloring on mobile gpus," *ArXiv*, vol. abs/1907.06740, 2019.

[3] X. Jin, R. Han, N. Ning, X. Li, and X. Zhang, "Facial makeup transfer combining illumination transfer," *IEEE Access*, vol. 7, pp. 80 928–80 936, 2019.

[4] H. Lahiani, M. Elleuch, and M. Kherallah, "Real time hand gesture recognition system for android devices," in *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2015, pp. 591–596.

[5] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs, "Automatic portrait segmentation for image stylization," *Computer Graphics Forum*, vol. 35, no. 2, pp. 93–102, 2016.

[6] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs, "Automatic portrait segmentation for image stylization," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 93–102.

[7] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy, "Synthetic depth-of-field with a single-camera mobile phone," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–13, 2018.

[8] S.-H. Zhang, X. Dong, H. Li, R. Li, and Y.-L. Yang, "Portraitnet: Real-time portrait segmentation network for mobile device," *Computers & Graphics*, vol. 80, pp. 104–113, 2019.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[11] H. Park, L. Sjosund, Y. Yoo, N. Monet, J. Bang, and N. Kwak, "Sinet: Extreme lightweight portrait segmentation networks with spatial squeeze module and information blocking decoder," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2066–2074.

[12] Y. Li, A. Luo, and S. Lyu, "Fast portrait segmentation with highly light-weight network," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1511–1515.

[13] Z. Wu, Y. Huang, Y. Yu, L. Wang, and T. Tan, "Early hierarchical contexts learned by convolutional networks for image segmentation," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1538–1543.

[14] X. Zhao, F. Tang, and Y. Wu, "Real-time human segmentation by bowtienet and a slam-based human ar system," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 5, pp. 511–524, 2019.

[15] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

[16] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Computer Vision – ECCV 2018*. Springer International Publishing, 2018, pp. 122–138.

[17] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1577–1586, 2020.

[18] M. Tan, B. Chen, R. Pang, V. Vasudevan, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2815–2823, 2019.

[19] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *ArXiv*, vol. abs/1905.11946, 2019.

[20] S. Minaee, Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *ArXiv*, vol. abs/2001.05566, 2020.

[21] R. Kalsotra and S. Arora, "A comprehensive survey of video datasets for background subtraction," *IEEE Access*, vol. 7, pp. 59 143–59 171, 2019.

[22] X. Liang, S. Liu, X. Shen, J. Yang, L. Liu, J. Dong, L. Lin, and S. Yan, "Deep human parsing with active template regression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 12, pp. 2402–2414, 2015.

[23] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, and S. Yan, "Human parsing with contextualized convolutional neural network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1386–1394.

[24] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3d and 2d human representations," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[25] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, "Deep automatic portrait matting," in *European conference on computer vision*. Springer, 2016, pp. 92–107.

[26] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, "Video segmentation by tracking many figure-ground segments," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2192–2199.

[27] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. V. Gool, "The 2017 DAVIS challenge on video object segmentation," *ArXiv*, vol. abs/1704.00675, 2017.

[28] C. Cuevas, E. M. Yáñez, and N. García, "Labeled dataset for integral evaluation of moving object detection algorithms: Lasiesta," *Computer Vision and Image Understanding*, vol. 152, pp. 103–117, 2016.

[29] F. Galasso, N. Shankar Nagaraja, T. Jimenez Cardenas, T. Brox, and B. Schiele, "A unified video segmentation benchmark: Annotation, metrics and analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3527–3534.

[30] C. Xu, S.-H. Hsieh, C. Xiong, and J. J. Corso, "Can humans fly? action understanding with multiple classes of actors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2264–2273.

APPENDIX

TABLE VI
DATASETS AND SUPPLEMENTARY MATERIAL.

Dataset	URL (Accesssed: 7.12.2020)
HMD	https://github.com/aisegmentcn/matting_human_datasets
BPS	http://www.cbsr.ia.ac.cn/users/ynyu/dataset
DCPS	https://competitions.codalab.org/competitions/24206
HPD	https://github.com/lemondan/HumanParsing-Dataset
UTP LSP	http://files.is.tuebingen.mpg.de/classner/up
UTPLSP ext.	http://files.is.tuebingen.mpg.de/classner/up
PHD	https://github.com/gasparian/PicsArtHack-binary-segmentation
DAPM	http://www.cse.cuhk.edu.hk/~leojia/projects/automatting
SegTrackV2	http://web.engr.oregonstate.edu/~lif/SegTrack2/dataset
DAVIS 2017	https://davischallenge.org/davis2017/code.html
LASIESTA	https://www.gti.ssr.upm.es/data/lasiesta_database.html
VSB100	https://www.mpi-inf.mpg.de/departments/computer-vision-and-machine-learning/research/video-segmentation
A2D	https://web.eecs.umich.edu/~jjcorso/r/a2d
supp. mat.[1]	https://cvl.tuwien.ac.at/uncategorized/seg2021/