Comparison of Real-time CNN-based Methods for Finger-level Hand Segmentation

Gibran Benitez-Garcia[a], Natsuki Takayama[a], Jesus Olivares-Mercado[b], Gabriel Sanchez-Perez[b], and Hiroki Takahashi[a,c]

[a]Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, Japan
[b]Instituto Politecnico Nacional, ESIME Culhuacan, Mexico City, Mexico
[c]Artificial Intelligence eXploration Research Center, The University of Electro-Communications, Tokyo, Japan

ABSTRACT

Hand segmentation is usually considered a pixel-wise binary classification problem, where the foreground hand is meant to be recognized in an input image. However, we envision that finger-level hand segmentation is more useful for applications like hand gesture and sign language recognition. Therefore, in this paper, we compare five state-of-the-art (SOTA) real-time semantic segmentation methods for the task of finger-level hand segmentation. To do that, we introduce two subsets consisted of 1,000 images manually annotated pixel-wise selected from new proposed datasets of hand gesture and world-level sign language recognition. With these subsets, we evaluate the accuracy of the recent SOTA methods of DABNet, FastSCNN, FC-HardNet, FASSDNet, and DDRNet. Since each subset has relatively few images (500), we introduce a simple yet effective loss function to train with synthetic data that includes the same annotations. Finally, we present a real-time performance evaluation of the five algorithms on the NVIDIA Jetson family of GPU-powered embedded systems, including Jetson Xavier NX, Jetson TX2, and Jetson Nano.

Keywords: Hang segmentation, finger segmentation, real-time CNN

1. INTRODUCTION

Hand segmentation is a dense prediction problem that detects every pixel that belongs to a hand (binary segmentation), where in some cases, these are disambiguated among left and right hands.[1] Several works employ hand segmentation as a preprocessing step for other tasks, such as hand gesture recognition and human behavior analysis.[2] However, some specific applications like hand gesture recognition might be easy to perform if the hand segmentation is achieved at the finger level. For example, the differences between gestures based on one or two fingers are clear if we use finger-level hand segmentation masks. On the contrary, the conventional left and right hand detections are not enough to exhibit distinct characteristics of each gesture, as shown in Fig. 1. Therefore, in this paper, we manually annotate pixel-wise finger labels of 1,000 images selected from two datasets of hand gesture and sign language recognition, respectively. With these datasets, we compare state-of-the-art (SOTA) segmentation algorithms for finger-level hand segmentation. Besides, since each subset has relatively few images, we introduce a simple yet effective loss function to train with a huge synthetic dataset[3] that includes the same finger-level annotations.

In order to use finger-level hand segmentation as a preprocessing step, faster than real-time performance is necessary for time-critical tasks. Common techniques used to fulfill the real-time requirements for semantic segmentation include network quantization, network compression, factorization of standard convolution, and efficient redesign of CNN-based architectures. This paper mainly focuses on the last, which is a common technique used in the most recent SOTA for real-time semantic segmentation. Thus, we compare the real-time performance of five algorithms: DABNet,[4] FastSCNN,[5] FC-HardNet,[6] FASSDNet,[7] and DDRNet.[8] Besides, we also test their capability to be implemented on low-power consumption embedded systems, such as Jetson Nano.

Further author information: (Send correspondence to G.B-G.)
G.B-G.: E-mail: gibran@ieee.org, Telephone: 042 443 5000

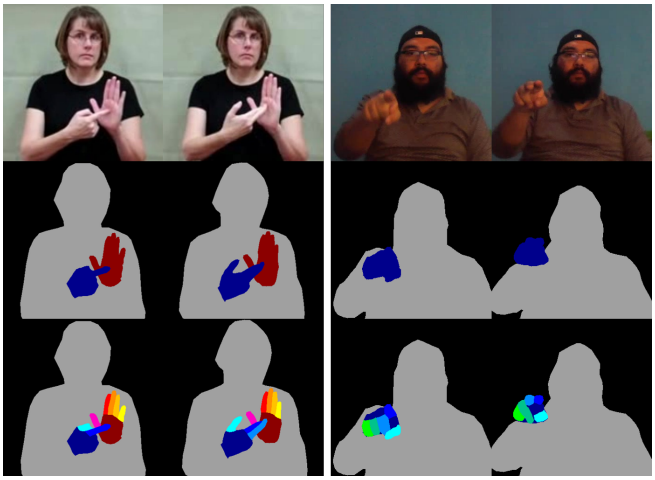

Figure 1. Comparison between one and two finger-based gestures using hand segmentation at finger level. The first and third rows show the RGB input and the finger-level hand segmentation, respectively. The second row shows the conventional right&left hand segmentation as a baseline.

2. RELATED WORK

In recent years, hand segmentation has being an active research topic.[9] Some of its main applications include hand gesture recognition (HGR),[10,11] RGB-based hand pose estimation,[3] and analysis of egocentric interactions.[1] For example, in[10] and,[11] the HGR is achieved in two steps; the first is focused on hand segmentation, while the second uses RGB and hand masks as input for the final classification. Binary hand segmentation has been extensively used as a preprocessing step for hand pose estimation. The authors of[12] proposed an end-to-end trainable 3D hand pose estimation framework based on foreground region supervision and 2D skeletal joint estimation. Since the output is a 3D mesh, it is possible to obtain approximated hand segmentation masks. On the other hand, it has been demonstrated that a good hand segmentation mask can be sufficient for recognizing actions and activities involving the hands in an egocentric visions. Thus, several methods rely on robust hand segmentation methods before activity recognition.[1] The recent work of[13] advances the SOTA accuracy of binary hand segmentation by proposing a Bayesian CNN-based model adaptation framework that can generalize between different domains. However, none of the previous works explore hand segmentation at finger level, as proposed in this paper.

3. REAL-TIME SEMANTIC SEGMENTATION

High-accuracy semantic segmentation methods usually rely on maintaining high-resolution features while applying convolution with large dilations rates to enlarge receptive fields.[14,15] However, the expensive computational resources required for these practices, in conjunction with the heavy pyramidal pooling techniques, limit the algorithms to achieve faster performance than real-time. Conversely, real-time segmentation algorithms build upon lightweight encoder-decoder[6,7] or bilateral pathway[5,8] architectures that usually employ compact pyramidal pooling modules and depth-wise convolutions.[4] In this way, we analyze five of the most efficient and balanced algorithms between accuracy and speed.

DABNet[4] builds upon Depth-wise Asymmetric Bottleneck modules designed to alleviate the dense number of parameters required in huge architectures. Each module employs depth-wise factorized convolutions with a bottleneck structure, which can extract local and contextual information jointly without a need for a pyramidal pooling module. The compact structure and the input resizing instead of convolving significantly reduce the parameters of the network.

FastSCNN[5] is based on a bilateral pathway aimed to combine spatial detailed high-resolution features and deep features at lower resolution. The two branches share the first convolutional layers which maintain relatively high-resolution features by only using depth-wise separable convolutions. The low-resolution path uses inverted

residual bottleneck blocks with a pyramidal pooling module (PPM) from PSPNet.[14] The fusion of both paths is based on a simple addition of features. This architecture presents the least number of parameters within the SOTA methods analyzed in this paper.

FC-HardNet[6] is a classic encoder-decoder architecture without pyramidal pooling and complex fusion modules. Its core contribution relies on the Harmonic Dense Blocks (HDBs), which are specifically designed to address the memory traffic problems and the density of computations from the dense blocks proposed by DenseNet. Each HDB reduces the layer connections and balances the input/output channel ratio based on the width of each layer according to its connections. This architecture focuses on improving the throughput of the feature maps by avoiding unnecessary DRAM accesses.

FASSDNet[7] employs a similar encoder-decoder architecture based on HDBs. However, it includes two key modules aimed to design a high-performance decoder. Dilated Asymmetric Pyramidal Fusion (DAPF) increases the receptive field on the last encoder by combining features at different scales. The second module, Multi-resolution Dilated Asymmetric (MDA), fuses and refines detail and contextual information from the early and deeper stages of the network. Both modules are designed to keep a low computational complexity by using asymmetric dilated convolutions.

DDRNet[8] is a recently proposed approach based on a two-pathway network. Its architecture is similar to FastSCNN with relatively high- and low-resolution branches. However, there are one-by-one corresponding relations between both resolution paths, defined with a bilateral fusion. This fusion includes a high-to-low and low-to-high fusion, emulating the different resolution combinations of complicated architectures. Besides, a new Deep Aggregation Pyramid Pooling Module (DAPPM) is introduced as an improvement of PPM. So that, DAPPM presents more (x4) context size combinations than the original PPM.[14] Note that this method uses deep supervision by adding an auxiliary loss at the middle of the architecture.

4. FINGER-LEVEL HAND DATASETS

Only a few available hand datasets include annotations other than binary pixel labels. For instance, EgoHands,[16] a large egocentric hand dataset for activity recognition, includes a subset of 4800 pixel-wise annotated images. The annotations include labels from 4 classes regarding own and others' left and right hands. Similarly,[11] presented 500 frames with left and right hand annotations from videos of gestures designed for touchless screen interactions. WorkingHands[17] recently presented the largest dataset with left and right segmented hands, comprising more than 400 thousand frames of "hands using tools" captured by thermal and RGB-D cameras. In contrast, the rendered hand pose dataset (RHD)[3] contains 43,986 synthetically generated images for hand pose estimation. RHD is the only publicly available dataset with finger-level pixel-wise annotations to the best of our knowledge.

Due to the lack of real-world finger-level annotated frames, we annotate two subsets from available datasets. We refine the 500 frames of the IPN hand dataset[18] chosen from,[11] which come from hand gesture videos with interactions of touchless screens. Besides, we define a subset of 500 images from the recently proposed Word-Level American Sign Language (WLASL) video dataset.[19] We manually choose the most representative frames that show different finger positions and a significant variety of backgrounds and subjects. We annotate 13 classes, which include palm and five fingers per hand and the person's shape.

5. EXPERIMENTS

We evaluate all methods using the RHD[3] synthetic dataset and the annotated subsets of IPN hand[18] and WLASL[19] datasets. We use the standard data split of RHD, which includes 41,258 images for training and 2,728 for testing. We randomly choose 400 images for training and 100 for testing for each subset. The performance evaluation is measured in mean intersection-over-union accuracy (mIoU) and frames per second (FPS). Finally, we report the number of parameters and computational complexity in GFLOPs.

Table 1. Quantitative results (mIoU)

Method	Params.	FPS	RHD	IPN	WLASL
DDRNet	5.73M	188.6	69.03	64.45	62.95
FASSDNet	2.85M	121.4	68.50	64.19	62.23
FC-HarDNet	4.12M	129.0	68.59	63.81	61.56
DABNet	0.76M	171.0	61.22	56.89	57.30
FastSCNN	1.14M	259.7	55.05	55.33	55.00

Table 2. Results of IPN subset with different training strategies

Method	Synthetic	Scratch	Fine-tune	cJointly	αJointly
DDRNet	32.21	50.56	61.48	62.25	64.45

5.1 Training Strategies

Since the subsets of annotated real-world images have relatively few images, common transfer learning techniques are beneficial in our experiments. The most straightforward strategy consists of training with a huge dataset to transfer the learned knowledge by fine-tuning the model with the small target subset. On the other hand, previous works have demonstrated that high accuracy can be achieved by jointly training on real and synthetic data.[20] Therefore, we propose a simple loss function to train the hand segmentation models with the RHD dataset and each real-world subset. The final loss is defined as a weighted sum of the desired semantic loss function, which can be expressed as $L_f = L_r + \alpha L_s$, where L_f, L_r, L_s represent the final loss, real-world-based loss, and the synthetic-based loss, respectively, and α denotes the weight assigned to regulate the contribution of L_s, which is empirically set as 0.4 in this paper.

5.2 Implementation Details

We use Python 3.6 and PyTorch 1.5 for the experiments. For a fair comparison, the same training setting is used for all models, where Stochastic Gradient Descent (SGD) is used as the optimizer. We employ the "poly" learning rate strategy and the cross-entropy loss by following the online bootstrapping strategy. Data augmentation consists of random horizontal flip, random scale, and random crop with 480×480 crops. The RHD images were rescaled to 640×640, while the IPN and WLASL were processed at the original resolution of 640×480. We trained all models with batch size 32 for 90,000 iterations when using RHD or the joint training (real+synthetic data), and 40,000 iterations when training the subsets from scratch. The fine-tuning from RHD models were trained for 30,000 extra iterations.

5.3 Experimental Results

Quantitative results of RHD, IPN, and WLASL test sets are shown in Table 1. FPSs were measured on an Intel Core i7-9700K desktop with a single NVIDIA GTX 1080Ti GPU and 64 GB of RAM. From this table, we can see that DDRNet achieves the highest results. Interestingly, FASSDNet presents better results on the real-world subsets than FC-HarDNet. In general, we can assume that the top 2 and 3 accuracy correspond to these methods. Fig. 2 qualitatively shows the superiority of the top 3 approaches over DABNet and FastSCNN. From a per-class evaluation, we found that the person class is the easiest to recognize, as shown in Fig. 2. In contrast, the left thumb and right medium fingers are, in general, the most difficult classes for IPN and WLASL datasets, respectively. Furthermore, Table 1 reaffirms that the number of parameters does not correlate with the inference speed. Being FastSCNN the fastest network with about 260 FPS. Note that all methods fulfill the requirement for a preprocessing step since they surpass the real-time performance.

The IPN and WLASL results were obtained following the joint training (α*Jointly*). Table 2 shows the superiority of our proposal concerning different training strategies. α*Jointly* overcomes the conventional joint

Figure 2. Qualitative results with a frame from the WLASL subset.

Table 3. Inference speed (FPS) on the NVIDIA Jetson family

Method	GFLOPs	Storage	Xavier	TX2	Nano
DDRNet	5.55	22MB	159.35	55.02	28.33
FASSDNet	6.60	11MB	60.0	26.8	12.11
FC-HarDNet	5.19	16MB	83.44	46.31	17.54
DABNet	6.12	3MB	30.5	10.9	4.81
FastSCNN	1.03	4.5MB	204.04	59.68	30.51

training (*cJointly*), which equally treats the losses of synthetic and real-world samples. We can see that the model trained only with the RHD dataset (*Synthetic*) is not reliable on real-world data (obtaining only 32% of mIoU). On the other hand, the fine-tuned strategy significantly overcomes the model trained from scratch. In this way, we prove that synthetic data is beneficial when the real-world dataset is relatively small.

Finally, Table 3 shows the efficiency results of each method when implemented on GPU-powered embedded systems. We employ the TensorRT tool for better optimization in the GPU. The inference speed was calculated from the average FPS rate of 10,000 iterations with $640 \times 480 \times 3$ images. The same resolution is used to obtain the computational complexity (GFLOPs). From Table 3, we see that only FastSCNN is capable of achieving real-time performance on the three embedded systems. Interestingly, DABNet presents the worst inference speed, while its model requires the smallest memory storage.

6. DISCUSSION AND CONCLUSION

Finger-level hand segmentation is more challenging than binary hand segmentation since multiple classes with significantly unbalanced sizes must be handled. Thus, semantic segmentation methods need to pay special attention to detail and contextual information on small classes, such as pinky and thumb fingers. In this paper, we particularly analyze five approaches designed to fulfill the mentioned requirements. Our analysis suggests that, the encoder-decoder methods (FC-HarDNet and FASSDNet) are slower but more accurate than the two-pathway approaches (FastSCNN and DDRNet). However, the bilateral fusion of DDRNet shows an effective solution. On the other hand, the multi-resolution-input structure of DABNet presents several downsides on accuracy and efficiency. The real-time requirements are achieved by all methods implemented on a common GPU. For the embedded systems, only the two-pathway approaches can fulfill them. Nonetheless, the Jetson Nano presents a challenge that the current methods cannot overcome.

In summary, we took a deep look into the possibility to achieve faster than real-time hand segmentation at finger-level, introduced two real-world subsets to carry full supervision, proposed a simple loss function to employ synthetic data, and identified the current SOTA methods that can also be applied on embedded systems.

REFERENCES

[1] Urooj, A. and Borji, A., "Analysis of hand segmentation in the wild," in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 4710–4719 (2018).

[2] Rangesh, A. and Trivedi, M. M., "Handynet: A one-stop solution to detect, segment, localize & analyze driver hands," in [*The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*], 1103–1110 (2018).

[3] Zimmermann, C. and Brox, T., "Learning to estimate 3d hand pose from single rgb images," in [*The IEEE International Conference on Computer Vision (ICCV)*], 4903–4911 (2017).

[4] Li, G. and Kim, J., "DABNet: Depth-wise Asymmetric Bottleneck for Real-time Semantic Segmentation," in [*British Machine Vision Conference (BMVC)*], (2019).

[5] Poudel, R. P., Liwicki, S., and Cipolla, R., "Fast-scnn: fast semantic segmentation network," in [*British Machine Vision Conference (BMVC)*], (2019).

[6] Chao, P., Kao, C.-Y., Ruan, Y.-S., Huang, C.-H., and Lin, Y.-L., "HarDNet: A Low Memory Traffic Network," in [*The IEEE International Conference on Computer Vision (ICCV)*], (October 2019).

[7] Rosas-Arias, L., Benitez-Garcia, G., Portillo-Portillo, J., Sanchez-Perez, G., and Yanai, K., "Fast and accurate real-time semantic segmentation with dilated asymmetric convolutions," in [*The 25th International Conference on Pattern Recognition (ICPR2020)*], 2264–2271, IEEE (2021).

[8] Hong, Y., Pan, H., Sun, W., Jia, Y., et al., "Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes," *arXiv preprint arXiv:2101.06085* (2021).

[9] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., and Garcia-Rodriguez, J., "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing* **70**, 41 – 65 (2018).

[10] Dadashzadeh, A., Targhi, A. T., Tahmasbi, M., and Mirmehdi, M., "Hgr-net: a fusion network for hand gesture segmentation and recognition," *IET Computer Vision* **13**(8), 700–707 (2019).

[11] Benitez-Garcia, G., Prudente-Tixteco, L., Castro-Madrid, L. C., Toscano-Medina, R., Olivares-Mercado, J., Sanchez-Perez, G., and Villalba, L. J. G., "Improving real-time hand gesture recognition with semantic segmentation," *Sensors* **21**(2), 356 (2021).

[12] Baek, S., Kim, K. I., and Kim, T.-K., "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in [*The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 1067–1076 (2019).

[13] Cai, M., Lu, F., and Sato, Y., "Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation," in [*The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 14392–14401 (2020).

[14] Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J., "Pyramid Scene Parsing Network," in [*The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (July 2017).

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

[16] Bambach, S., Lee, S., Crandall, D. J., and Yu, C., "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in [*The IEEE International Conference on Computer Vision (ICCV)*], 1949–1957 (2015).

[17] Kim, S., Chi, H.-g., Hu, X., Vegesana, A., and Ramani, K., "First-person view hand segmentation of multi-modal hand activity video dataset," in [*British Machine Vision Conference (BMVC)*], (2020).

[18] Benitez-Garcia, G., Olivares-Mercado, J., Sanchez-Perez, G., and Yanai, K., "Ipn hand: A video dataset and benchmark for real-time continuous hand gesture recognition," in [*The 25th International Conference on Pattern Recognition (ICPR2020)*], 4340–4347, IEEE (2021).

[19] Li, D., Rodriguez, C., Yu, X., and Li, H., "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in [*The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*], 1459–1469 (2020).

[20] Richter, S. R., Vineet, V., Roth, S., and Koltun, V., "Playing for data: Ground truth from computer games," in [*The European Conference on Computer Vision (ECCV)*], 102–118, Springer (2016).