WorkingHands: A Hand-Tool Assembly Dataset for Image Segmentation and Activity Mining

Roy Shilkrot[13]
roys@cs.stonybrook.edu
Supreeth Narasimhaswamy[1]
sunarasimhas@cs.stonybrook.edu
Saif Vazir[1]
svazir@cs.stonybrook.edu
Minh Hoai[12]
minhhoai@cs.stonybrook.edu

[1] Computer Science Department
Stony Brook University
Stony Brook, NY 11794, USA

[2] VinAI Research
Hanoi, Vietnam

[3] Tulip Interfaces Inc.
Somerville, MA 02143, USA

Abstract

Computer vision in manufacturing is a decades long effort into automatic inspection and verification of the work pieces, while visual recognition focusing on the human operators is becoming ever prominent. Semantic segmentation is an exemplary vision task that is key to enabling crucial assembly applications such as completion time tracking and manual process verification. However, focus on segmentation of human hands while performing complex tasks such as manual assembly is still lacking. Segmenting hands from tools, work pieces, background and other body parts is difficult because of self-occlusions and intricate hand grips and poses. In this paper we introduce WorkingHands, a dataset of pixel-level annotated images of hands performing 13 different tool-based assembly tasks, from both real-world captures and virtual-world renderings, with RGB+D images from a high-resolution range camera and ray casting engine. Moreover, using the dataset, we can learn a generic Hand-Task Descriptor that is useful for retrieving hand images and video performing similar operations across different non-annotated datasets.

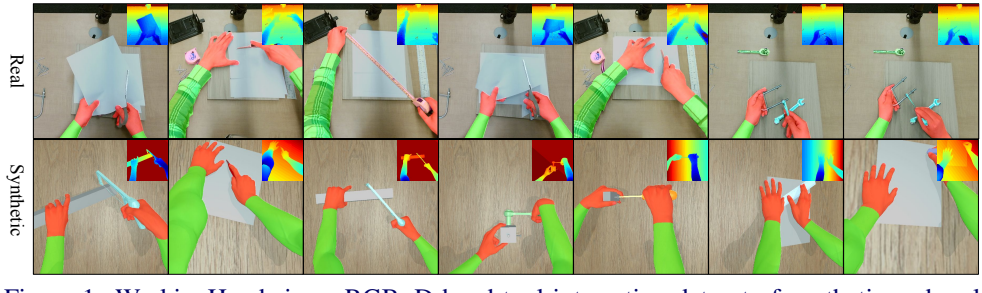

Figure 1: WorkingHands is an RGB+D hand-tool interaction dataset of synthetic and real data, with semantic segmentation annotations for 16 classes.

Hand Segmentation				Hand Grasp, Action and Pose				
Work	**#frames**	**Depth**	**Annotation**	**Work**	**#Syn**	**#Real**	**Seg.**	**#Obj.**
EgoHands [2]	4.8K	N	Man.	GUN-71 [27]	0	12K	N	28
Handseg [21]	210K	Y	Auto.	MIT CBMM [29]	0	12K	N	148
NYU Hand Pose [32]	6.7K	Y	Auto.	Choi et al. [5]	330K	0	N	600
HandSegNet [33]	44K	Y	Syn.	SynthHands [23]	220K	3.2K	Y†/N	7
HandNet [34]	213K	Y	Auto.	InterSegHands [6]	0	52K	Y††	28
EGTEA Gaze+ [18]	14K	N	Man.	Yale [6]	0	18K	N	11
Khan et al. [16]	1.6K	N	Man.	Garcia et al. [12]	0	105K	N	26
TV-Hand [25]	9.5K	N	Man.					
Ours	7.9K†††	Y	Semi-auto	Ours	4.2K	3.7K	Y/Y	13

Table 1: Comparison of hand analysis datasets. We compare datasets geared towards segmentation, and others towards hand pose and grasp that may also have segmentation annotation. †: [23] contains only *background* segmentation and doesn't separate hand from object. ††: [6] contains *only depth* images without color. Fig. 2 illustrates these shortcomings. †††: Note this is the number of raw frames *without* augmentation.

1 Introduction

Computer vision is now used in many of the manufacturing and fabrication fields. Manufacturers are using high-end machine vision for part inspection and verification, as well as means to track the workers and the work pieces to gain crucial insight into the efficiency of their assembly lines. Small-scale fabrication, on the other hand, happens virtually anywhere, even at home, at school, or in personal fabrication shops. Still all kinds of fabrication, mass- or small-scale, share a commonality: *manual assembly tasks performed by humans*. This comes as a stark contrast to the minor offering of computer vision methods to understand manual assembly scenes. To this end we offer a dataset of fully annotated images of assembly tasks with manual tools, named *WorkingHands*. This dataset includes both real-world and virtual-world samples, and the dataset is useful for various computer vision tasks including semantic segmentation and activity retrieval.

Dataset uniqueness. Segmentation of arms, hands and tools can enable very appealing applications in manufacturing, such as tracking human operators motions, precise actions, utilization of tools, and verify a correct, safe occupancy of the workstation area. There are several large-scale datasets to assist in segmentation algorithm development, such as ImageNet [10], COCO [19], SUN [35], PASCAL [11], and ADE20K [37]; but these datasets are not explicitly developed for hand analysis. Hand image analysis datasets [2, 16, 21, 22, 38] were proposed for segmentation of hands, but many do not involve hand-object interactions or other body parts. In parallel, hand-object pose and grasp estimation is another important topic of research in robotics, and datasets abound [9, 12, 27], however most offerings generally exclude segmentation annotation data, or provide segmentation data of low quality (e.g., no hand-tool or hand-arm separation). Table 1 and Figure 2 compare existing datasets; most are unsuitable for segmenting hands during assembly tasks with hand-held tools.

Dataset composition. Manually annotating distinct semantic parts in images is tedious, error-prone, and may be prohibitively expensive. However, a thorough segmentation annotation of the image can enable more powerful downstream applications, such as object and activity detection. Many recent hand image analysis works (e.g. hand pose, grasp, segmentation and others) rely on big synthetic data to bootstrap deep learning [12, 23, 38] and follow

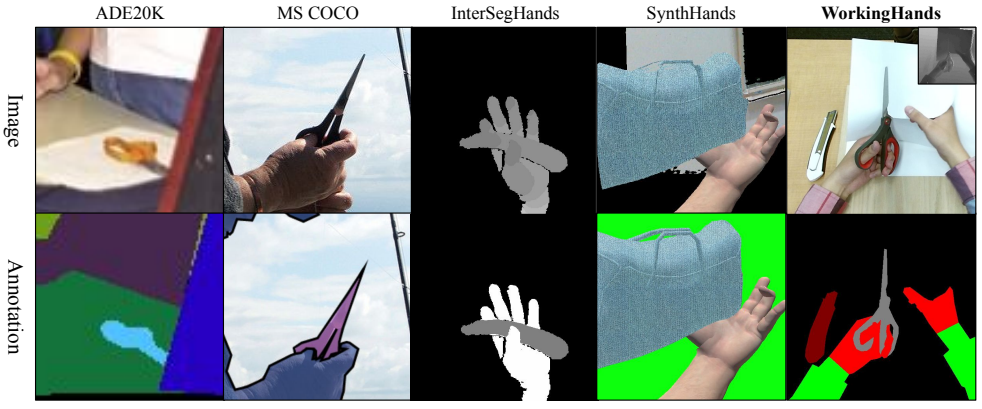

Figure 2: Qualitative comparison of annotation quality in our dataset vs. [6, 19, 23, 37]. Our annotation is more complete and precise in terms of polygon quality, depth and background information, arm and hand separation, etc. In addition, other datasets have a far smaller amount of instances in most object categories (see Table 2).

Tool:	Scrwdrv.	Wrench	Pliers	Pencil	Scissors	Cutter†	Hammer	Ratchet	Tape	Saw	Eraser	Glue	Ruler
COCO	0	0	0	0	**975**	**4507**	0	0	0	0	0	0	0
SUN	1	64	1	7	3	63	4	0	0	1	0	0	2
ADE20K	2	2	1	8	16	161	5	0	1	1	0	0	4
WorkingHands	**1616**	**2051**	**1586**	**2320**	864	2021	**1066**	**967**	**796**	**1183**	**846**	**650**	**2428**

Table 2: Comparison of number of pixel-level annotated object instances among prominent segmentation datasets and our own. † "Knife" also considered as "Cutter" in other datasets.

up with training on a smaller real-world annotated set. Segment annotation can be easily extracted in synthetic data and 3D models can be parameterized to augment the scene in a multitude of novel situations, which led to the creation of specialized generation packages [36]. WorkingHands contains both real and synthetic high quality annotations not only of the hands, arms and tools, but also of any unused tools placed on the work desk. To the best of our knowledge, ours is also the first segmentation dataset that concentrates on small-scale *manual assembly*. A sample of our annotated dataset is presented in Fig.1. The entire dataset is available at `http://hi.cs.stonybrook.edu/workinghands`.

2 The WorkingHands Dataset

We chose to deliver two types of image data in WorkingHands, real-world and synthetic, so together they can provide a generalized and practical database for semantic segmentation for small-scale assembly works.

The structure of the dataset is designed following PASCAL [11], which includes color images and segmentation class labels (See Fig. 4). The pixel-value of the segments in the label image ranges from 0 to $N-1$, where N is the number of classes. In addition, we include depth images to provide extra information, since depth has been shown to be useful for understanding human body parts [30, 32]. RGB information is also very hard to generalize properly. In real world situations there is immense color variability, for example shirt, tool,

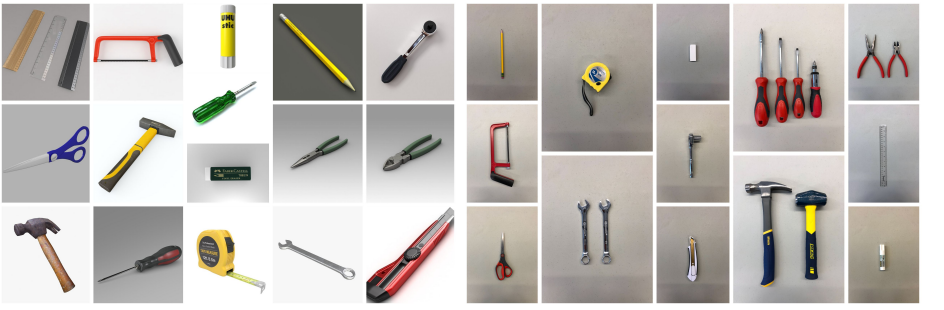

Figure 3: The tools used in WorkingHands. Left: synthetic tool models, Right: real tools.

Tool	Task	RGB	Tool	Task	RGB
Screwdriver	Tight screws	FFFF00	Cutter	Cut paper	800000
Wrench	Tight hex nuts	00FFFF	Hammer	Drive nail	808000
Tape measure	Measure object	800080	Ratchet	Tight hex nuts	008000
Pencil	Sketch on paper	C0C0C0	Pliers	Cut wires	FF00FF
Eraser	Erase a sketch	000080	Saw	Cut a board	008080
Scissors	Cut paper	808080	Glue	Glue papers	CD853F
Ruler	Draw a line	4682B4			

Table 3: Tools used in WorkingHands with their tasks and mask RGB values (as seen in Fig. 1,2,4,7). RGB for Hand is FF0000, Arm is 00FF00 and Background is 000000.

background or skin colors, let alone variation in lighting. Depth images circumvent these problems while the added cost of obtaining them is not high.

Tools and Tasks Selection. We aim to create a dataset for most small-scale assembly works. However, assembly is a widely diverse action with many goals that uses a large class of tools. We chose to feature common tools that exist in most households and manual assembly pipelines, such as the 13 hand-held tools listed in Table 3. Pictures of the collection of tools used in our recordings can be seen in Fig. 3. We staged a small workstation with wooden and paper craft pieces to be used for work pieces, and instructed the "workers" to perform simple assembly tasks (see Table 3).

Capturing Real Data. Data was captured using a standard Kinect V2 camera, capturing at 1920×1080 resolution for RGB and 512×424 for depth at 7 FPS. Depth and RGB streams are pixel-aligned using the provided SDK and the camera intrinsic and extrinsic parameters. The camera is mounted above the desk to provide first-person perspective effects. This was done to allow our data to be used both for segmentation of images from head-mounted gear as well as top-view cameras in a workbench, which are becoming ubiquitous in the manufacturing world. During the recording, the real time video output was displayed so that the workers could adjust their postures to avoid excessive occlusion. Given the instructions as shown in Table 3, three volunteers were recruited (gender: one female, two males; skin pigment complexion: one Caucasian, two Asians). Multiple tools are allowed to use in one task in order to help complete the work. Per each task, the camera started to capture images after the workers began their work, and stopped automatically after recording 150 frames. A total of 39

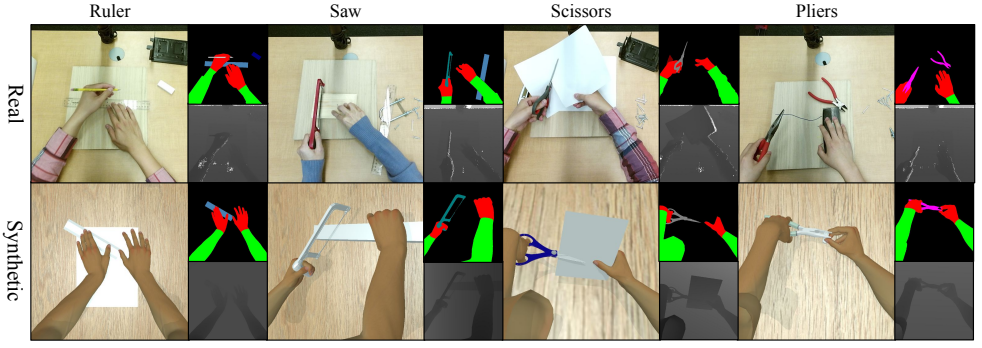

Figure 4: Sample annotations, color, and depth data, real and synthetic.

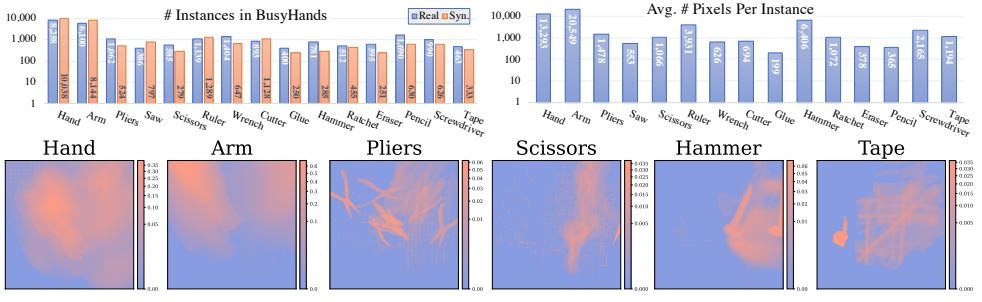

Figure 5: Top-left: number of class instances in the WorkingHands dataset; Top-right: average number of pixels for an instance of each class (e.g., *Hand* instances cover roughly 13,300 pixels on average). Note the logarithmic scale. Bottom: heatmap illustration of the pixel-position of a few classes in the Real part of the dataset.

films were captured, of which 26 were fully annotated with segmentation information. We employed Python-LabelMe (https://github.com/wkentaro/labelme), an open source image annotation software based on the original LabelMe project from MIT [28], to annotate different semantic parts and assign appropriate labels to them. The results can be seen in Fig. 1.

Rendering Synthetic Data. To enrich our dataset with a large number of samples, we adopted using synthetic data. Recently, tools were presented to create hand pose synthetic data [36], however not for segmentation. To generate realistic data to be on a par with real data, we purchased high quality 3D models of tools (see Fig. 3) as well as a highly realistic pair of hands, and loaded them in the Blender software (http://www.blender.org). All the manual tasks (or instructions) were simulated by creating realistic key-frame animations mimicking human motion by observation. To increase the generality of the dataset, so it can be applied in various physical environments, we use five camera perspectives in the synthetic dataset.

Unlike real-world captures, annotating semantic parts in a virtual environment is very straightforward. In Blender, we unwrapped the meshes of tools, hands, and arms to 2D UV maps, then painted the UV maps using solid colors. Each color is one-to-one mapped to one class label in our dataset according to the RGB-codes dictionary (see Table 3). Later, we utilize these colors to retrieve corresponding label numbers. Given a mapped texture

Figure 6: Augmentations enabled in our dataset. Background can be replaced since all tools in the scene are annotated. Skin tone and "plumpness" of the synthetic hands are parametric.

in Blender, the software will output rendered images of RGB and semantic labels for all the designed animation frames automatically. A synthetic depth map can be obtained by outputting the virtual camera's z-buffer, and is pixel-aligned to the other streams.

Dataset Parts. The real part of the dataset has 3695 labeled images, while in the synthetic part has 4170 images. Instances wise, we have 9505 instances of tools in the real dataset, and 4170 instances of tools in the synthetic parts. Proportions of each tool in both real data and synthetic data are charted in Fig. 5. Potential data augmentations are shown in Fig. 6.

2.1 Semantic Segmentation Labeling Evaluation

Related work. Semantic segmentation has long been a central pursuit in the computer vision research agenda, with compelling applications in autonomous navigation, security, image-based search and manufacturing, to name a few. Recently, segmentation research has seen a tremendous boost in offerings of deep convolutional network architectures, marked roughly by Long et al's Fully-Convolutional Networks (FCN) work [20] as the new era of semantic segmentation. The key insight in [20], which still resonates in most state-of-the-art contributions today, is using a pre-trained powerful visual feature-extracting network (such as VGG [31], ResNet [13], or a standalone one) and layer on top of it a decoding and unpooling mechanism to predict a class for each pixel at the original resolution.

Evaluated Segmentation Methods. We experimented with the following semantic segmentation algorithms from the latest literature: SegNet [3], Mobile UNet [14], Full-Resolution Residual Networks (FRRN) [26], AdapNet [33], DeepLab [7, 8]. All algorithms were implemented with Tensorflow [1], forking the Semantic Segmentation Suite project [29].

Metrics. We use a standard metric to evaluate labeling performance. The most adopted is the *intersection-over-union* metric $IoU = \frac{TP}{TP+FP+FN}$, where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels, respectively [11]. We average over all classes and then over samples to obtain the mean intersection over union (mIOU).

Evaluation Results. The results of training and testing with the selected evaluation methods (listed in §2.1) are given in Table 4. We notice that full-resolution residual networks (FRRNs) are mostly superior under all categories, followed by the SegNet with skip connections. In Figure 7 we show example results on the Real test set with FRRN-B and SegNet-Skip.

	Algorithm								
Train → Test	AdapNet	D.LabV3	D.LabV3+	SegNet	SegNet-Sk	FRRN-A	FRRN-B	Mob.UNet	Mob.UNet-Sk
Rl. → Rl.	0.174	0.113	0.139	0.257	**0.336**	0.316	0.283	0.234	0.22
Syn. → Syn.	0.714	0.532	0.584	0.782	0.856	0.856	**0.858**	0.759	0.842
Syn.+Rl. → Rl.	0.291	0.212	0.227	0.328	0.494	0.502	**0.589**	0.216	0.388
Syn.+Rl. → Syn.	0.623	0.367	0.313	0.591	0.641	**0.776**	0.763	0.547	0.713

Table 4: Results of the baseline methods on the WorkingHands dataset, in terms of *mIOU*. The first column marks training vs. testing, e.g. 'Syn.+Rl. → Rl.' means training on both synthetic and real images (training set) and testing only on real images (test set held out). 'Sk' indicates the use of skip connections in the network.

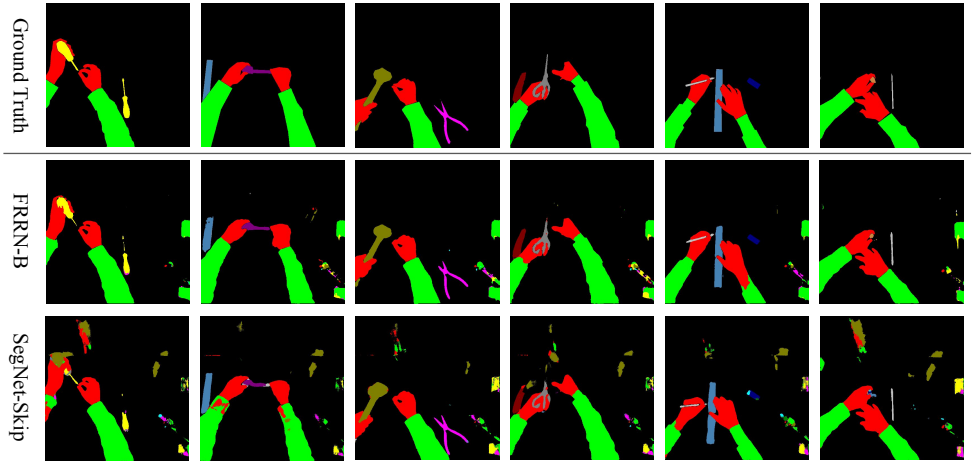

Figure 7: Results of running FRRN-B [26] and SegNet-Skip [3] on a number of samples from the Real test dataset. The top row is the ground truth annotation.

Some post processing cleanup on the segmentation result, in particular blob geometry analysis (which we did not attempt), could potentially alleviate the level of noise. Another insight is that the existence of synthetic data dramatically increases the power of the learners in accuracy over Real data. In the case of FRRN-A, for example, mIOU over the Real test set shot up from 0.336 when training just with Real images up to 0.502 with synthetic data.

3 A Hand-Task Descriptor for Activity Mining

Knowing which tool is being used at a given moment is useful for both hand pose estimation and activity recognition. We describe here a Hand-Task Descriptor that can be used to represent the assembly activity being performed. To obtain the Hand-Task descriptor, we train a classifier to predict the tool(s) being used by the hands based on the motion of the hands and the arms. This training procedure requires a dataset with pixel-wise annotations for the arms, the hands, and the tools being used; and the WorkingHands dataset is particularly suitable for this purpose, unlike many existing datasets for hand pose estimation (e.g., [17]) or hand-object interaction (e.g., [23]) that do not have annotation for the tools being used. As will be seen, the learned descriptor is useful for retrieving video instances with similar hand configurations and movements. Figure 8 illustrates this proposed pipeline.

Retrieved From	Synthetic		Synthetic+Real	
	AP@5	AP@10	AP@5	AP@10
Pretrained 3D ResNet-152	0.0836	0.0836	0.6672	0.5168
HandDescriptor-512 (proposed)	0.5145	0.5268	0.9236	0.8913
HandDescriptor-1024 (proposed)	**0.5481**	**0.5740**	**0.9322**	**0.9563**

Table 5: Average Precision at k (AP@k) for video retrieval. Query videos are real data. Two cases are considered: 1) the retrieved videos are from the Synthetic data, and 2) the retrieved data come from both the Synthetic and Real data. Both Hand-Descriptors of 512 and 1024 dimensions outperform the descriptor obtained using the pretrained network.

	ScrwDrvr.	Wrench	Pliers	Pencil	Eraser	Scissors	Cutter	Hammer	Ratchet	TapeMeas.	Saw	Glue	Ruler
AP	0.358	0.178	0.327	0.109	0.130	0.110	0.145	0.126	0.063	0.586	0.121	0.162	0.124

Table 6: Average Precision (AP) results for the tool-being-used classifier trained on synthetic data and tested on the real data of the WorkingHands dataset. Mean AP = 0.1954.

Learning the descriptor. To obtain a hand-task descriptor, we learn a classifier to predict the tools being used by the hand based on the motion of the hands and arms. Since more than one tool can be used at a time, we pose this as a multi-label classification problem. The target output of the classifier is a 13-dimensional binary vector for 13 different tools. The input to the classifier is a block of 16 frames, encoding the binary masks for the hands and arms. We use the synthetic data and real data of the WorkingHands dataset for training and testing, respectively. We also perform data augmentation by flipping the frames horizontally and vertically. The training set has 940 videos and the test set has 880 videos. The classifier is a 3D Resnet-152, and it is initialized by as a pretrained network on the Kinetics dataset [15]. Once the classifier has been trained, the activation values at the penultimate layer is taken as the feature descriptor for the input video.

Using the descriptor. We can use the Hand-Task descriptor to retrieve videos with similar hand configuration and movement. Specifically, given a query video, we can extract the feature vector (from the penultimate layer of the trained 3D Resnet-152) and retrieve other videos with similar feature vectors (using cosine similarity).

Table 5 reports the quantitative performance of the Hand-Task descriptor, evaluated on the WorkingHands dataset. We use the Average Precision at k (AP@k) as the performance metric for the retrieval task. AP@k can be computed as follows. For each query video, we retrieve the top k videos that are similar to the query video and then compute the precision value (based on whether the retrieved videos use the same tool(s) as the query video). Finally, we average the precision values over all query videos. As a baseline, we also evaluate the performance of a 3D ResNet-152 pretrained on the Kinetics dataset [15]. We also experiment with different sizes (feature dimensions) for the Hand-Descriptor. As can be seen from Tab. 5, the Hand-Descriptors outperforms the baseline method, and the former perform well even when the query video is real and the database videos are synthetic.

Table 6 shows the Average Precision for detecting activity classes. Some activities are easier to detect than others, and the mean AP is 0.1954. This is a challenging task in general, given the classifier is trained on the synthetic data and evaluated on the real data.

Figure 9 shows some qualitative results for using the Hand-Task descriptor to retrieve

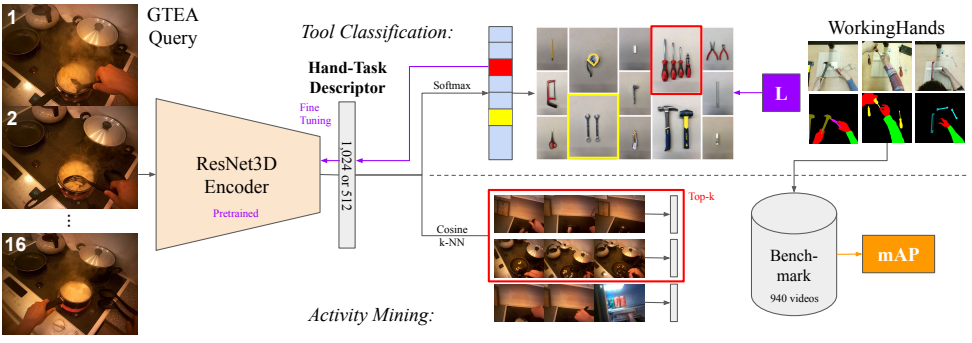

Figure 8: Our proposed activity mining pipeline. A tool classifier is trained to bootstrap the Hand-Task Descriptor, which is useful for mining for similar hand-tool activities from other datasets without annotation. Purple lines denote gradient flow. 'L' - cross-entropy loss, 'mAP' - mean Average Precision.

similar videos on both the WorkingHands and EGTEA Gaze+ datasets. Although the Hand-Task descriptor was not trained or fine-tuned on the EGTEA Gaze+ dataset, it can accurately retrieve videos with similar activities. Note also that the EGTEA Gaze+ dataset involves different activities, and hands in this dataset interact with different objects than the ones in the WorkingHands dataset. This demonstrates the generalization capability of the Hand-Task descriptor, and also proves the usefulness of the WorkingHands dataset.

4 Conclusions

To advance the field of computer vision methods for assembly operations, we contribute WorkingHands, a high-quality fully annotated segmentation dataset with both real and synthetic image data. We present an evaluation of numerous leading segmentation algorithms on our dataset as a baseline for other researchers. The WorkingHands dataset also lends itself to create a hand-task descriptor that can predict which object is being used as well as retrieve similar manual tasks in large corpora without such annotation. We demonstrate this capability on the EGTEA Gaze+ dataset [13].

Acknowledgments. We would like to thank the Nvidia corporation for their generous donation of a Titan Xp and Quadro P5000 GPUs, which were used in this project.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: a system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation*, 2016.

[2] M. Afifi. Gender recognition and biometric identification using a large dataset of hand images. *CoRR*, abs/1711.04322, 2017.

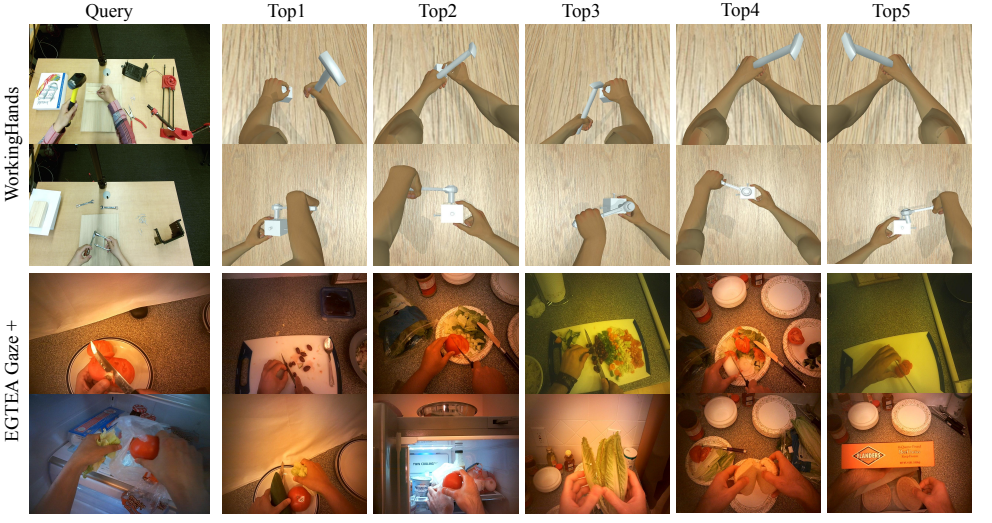

Figure 9: Qualitative results for retrieving similar activities on the WorkingHands and EGTEA Gaze+ datasets. We visualize one frame from a query video and one frame from each of the top-5 retrieved videos. The first two rows corresponds to WorkingHands data and the last two rows corresponds to EGTEA Gaze+ dataset.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[4] S. Bambach, S. Lee, D. J. Crandall, and C. Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proc. ICCV*, 2015.

[5] Z. Bo, H. Zhang, J. Yong, and F. Xu. Denseattentionseg: Segment hands from interacted objects using depth input. *arXiv preprint arXiv:1903.12368*, 2019.

[6] I. M. Bullock, T. Feix, and A. M. Dollar. The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments. *The International Journal of Robotics Research*, 34(3):251–255, 2015.

[7] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[8] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[9] C. Choi, S. Ho Yoon, C.-N. Chen, and K. Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proc. ICCV*, pages 3123–3132, 2017.

[10] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[12] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proc. CVPR*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[15] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, A. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[16] A. U. Khan and A. Borji. Analysis of hand segmentation in the wild. *CoRR*, abs/1803.03317, 2018.

[17] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Proc. CVPR*, 2015.

[18] Y. Li, M. Liu, and J. M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proc. ECCV*, 2018.

[19] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[20] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015.

[21] S. R. Malireddi, F. Mueller, M. Oberweger, A. K. Bojja, V. Lepetit, C. Theobalt, and A. Tagliasacchi. Handseg: A dataset for hand segmentation from depth images. *CoRR*, abs/1711.05944, 2017.

[22] A. Mittal, A. Zisserman, and P. H. S. Torr. Hand detection using multiple proposals. In *Proc. BMVC.*, 2011.

[23] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *Proc. ICCV*, 2017.

[24] B. Myanganbayar, C. Mata, G. Dekel, B. Katz, G. Ben-Yosef, and A. Barbu. Partially occluded hands: A challenging new dataset for single-image hand pose estimation. In *Proc. ACCV*, 2018.

[25] S. Narasimhaswamy, Z. Wei, Y. Wang, J. Zhang, and M. Hoai. Contextual attention for hand detection in the wild. *arXiv preprint arXiv: 1904.04882*, 2019.

[26] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *CoRR*, abs/1611.08323, 2016.

[27] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from RGB-D images. In *Proc. ICCV*, pages 3889–3897, 2015.

[28] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1):157–173, 2008.

[29] G. Seif. Semantic Segmentation Suite. https://github.com/GeorgeSeif/Semantic-Segmentation-Suite, 2018.

[30] J. Shotton, A. 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 *Proc. CVPR*, 2011.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[32] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *Proc. ACM SIGGRAPH*, 33(5):169:1–169:10, Sep 2014.

[33] A. Valada, J. Vertens, A. Dhall, and W. Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *Proc. Intl. Conf. on Robotics and Automation*, 2017.

[34] A. Wetzler, R. Slossberg, and R. Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. *arXiv preprint arXiv:1507.05726*, 2015.

[35] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010.

[36] D. Yang, B. Moon, H. Kim, and Y. Choi. Synthetic hands generator for rgb hand tracking. In *TENCON 2018-2018 IEEE Region 10 Conference*, 2018.

[37] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proc. CVPR*, 2017.

[38] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *Proc. ICCV*, 2017.