Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence

2021 International Conference on Computer, Control and Robotics

Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence

Riccardo Zanella
DEI - University of Bologna
riccardo.zanella2@unibo.it

Alessio Caporali
DEI - University of Bologna
alessio.caporali2@unibo.it

Kalyan Tadaka
DEI - University of Bologna
kalyan.tadaka@studio.unibo.it

Daniele De Gregorio
EYECAN.ai
daniele.degregorio@eyecan.ai

Gianluca Palli
DEI - University of Bologna
gianluca.palli@unibo.it

***Abstract*—In this work, we present a procedure to automatically generate an high-quality training dataset of cable-like objects for semantic segmentation. The proposed method is explained in detail using the recognition of electric wires as a use case. These particular objects are commonly used in an extremely wide set of industrial applications, since they are of information and communication infrastructures, they are used in construction, industrial manufacturing and power distribution. The proposed approach uses an image of the target object placed in front of a monochromatic background. By employing the chroma-key technique, we can easily obtain the training masks of the target object and replace the background to produce a domain-independent dataset. How to reduce the reality gap is also investigated in this work by correctly choosing the backgrounds, augmenting the foreground images exploiting masks. The produced dataset is experimentally validated by training two algorithms and testing them on a real image set. Moreover, they are compared to a baseline algorithm specifically designed to recognise deformable linear objects.**

***Keywords*—Image Segmentation, Dataset Labeling, Deformable Objects, Chroma-key, Domain Randomization.**

I. Introduction

The availability of big public datasets [1]–[3] has promoted advances of deep learning algorithms in computer vision applications, such as image classification, object detection and semantic segmentation. Thus, the key issue in modern computer vision deals more and more with gathering and labeling big amounts of data. Usually, the process of segmenting and annotating the training images is performed manually, and it is notoriously tedious, inaccurate and time consuming. Moreover, the more complex the visual perception task is, the slower becomes the required annotation procedure. For instance, labeling a single image for 2D semantic segmentation can take several hours per image. Innovative companies, like Scale.ai, Superannotate.ai, Segments.ai and many others, are basing their business on advanced image labeling pipeline that can speed-up and lighten the burden. These solutions often exploit a superpixel algorithm which helps the user to quickly select large portions of the image instead of individual pixels. Other new approaches rely on weakly supervised learning [4] as Segments.ai that iterates between image labeling and model training in order to provide the user with initial – coarse – labels for each new image instead of having it labeled from scratch.

The aforementioned big public datasets [1]–[3] usually concern general classes (e.g. person, car, tree, cat, dog, etc.) that may not suit the needs of a specific task. Robotic applications, especially in industrial settings, typically require the detection or segmentation with very high success rate of small but very specific set of object instances captured from different viewpoints in highly-cluttered scenes. Electric wires, more than other objects, have some peculiarities that bring to some interesting challenges on segmentation tasks: 1) they are deformable objects, which means that they are not characterized by a specific shape; 2) they are very lacking in features; 3) they aren't characterized by any particular color. Since a cable can feature a wide variety of shapes and colors, to train a segmentation model, the generation of a large scale dataset to cover such great variability is necessary.

This article is motivated by the lack of simple and effective solutions to generate big image dataset for training, specifically in the field of cable-like objects. We present here a method, developed within the Horizon 2020 REMODEL project, which relies on the chroma-key technique and enables to easily label a given object on an entire video sequence. As the REMODEL project aims at automatizing the wiring procedure, image segmentation is a key point to address sub-tasks like cable grasp, terminal insertion and wire routing. Therefore, in this paper we focus mainly on electric wires, even thought we show also the applicability of the proposed method to other object typologies. To generate large datasets, a novel labeling pipeline demanding a minimal human intervention despite the volume of produced labeled data is implemented in this work. First, a video sequence of the target object is taken with a proper background which should be homogeneous and easy to be distinguished from the target. Then, the user does not have to manually label the acquired images, but, instead: a) he/she has simply to tune 1 (possibly 3) parameter once per video

This work is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 870133 as part of the RIA project REMODEL (Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects).

sequence; b) the target object will be automatically segmented in the entire sequence, by producing a superimposed pixels mask for each frame, by exploiting chroma key (a well known technique used to compose two images); c) the original video sequence backgrounds can, therefore, be replaced to increase the domain randomization. The main contributions of this work can be summarized as follows:

- The first chroma key approach for data labeling;
- An easy and reliable procedure to automatically generate large training datasets of specific items for semantic segmentation;
- A high-quality public dataset of electric wires for semantic segmentation in general purpose applications (available online https://www.kaggle.com/zanellar/electric-wires-image-segmentation);
- Tests and comparisons of different state-of-the-art algorithms on this dataset.

II. Related Works

The annotation processes for semantic segmentation is labor-intensive using traditional methods [3], [5]. A lot of research effort have been spent on investigating alternative strategies to help the human operator in this task [6]. Advanced solutions like weakly or semi-supervised segmentation have been proposed.

Weakly supervised learning studies attempt to construct predictive models by learning with incomplete, inexact or inaccurate supervision [4]. Weakly supervised learning for semantic segmentation employs different levels of supervision, like labeling only few pixels (e.g. interactive methods [7]), grouping images containing common objects (e.g. co-segmentation [8]) or providing only image-level labels [9]. In interactive segmentation frameworks [7] small portions of target objects are roughly highlighted by human operators through markers, called seeds. These seeds are used for a training stage that will produce some rough labels for all other images. The user can then produce more seeds and repeat the procedure until the desired quality level is reached.

Object co-segmentation [8] aims to detect and segment the semantically similar objects from a set of images. It gives very weak prior that the images contain the same objects for automatic object segmentation. Although there is a certain gap between models trained by weak/semi-supervision and models trained by full supervision, many researchers are making efforts to reduce the gap.

Several approaches for creating datasets have been developed also within the robotics research community. A semi-automatic method to create labeled datasets for object detection is presented in [10]. The system leverages on moving a 2D camera by means of a robot and an augmented reality pen to define initial object bounding box. Zeng et al. [11] present a 6D pose estimation system for Amazon Picking Challenge, where they segment and label the set of target objects placed on the shelf from depth and multi-view information. This work, not only requires depth information, but is also strongly tailor-made on the task's domain. Besides the robotic community, another popular approach to speed-up creation of training datasets consists in the use of synthetic rendered images [12]–[16]. However, obtaining a dataset of realistic images requires hours of highly specialized human work to design suitable synthetic scenes along with high-performance graphical hardware. In these cases, a non-photorealistic scene (*i.e.* a simple CAD model rendered on random background) can cause a well-known problem called *domain shift* [17]. In order to reduce this shift, and avoid spending time on photorealism, several *domain adaptation* techniques are applied [18]–[20]. Recent works [18], [21], [22] focus on developing ad-hoc adaptation techniques to close the performance gap between training and test distribution. Unfortunately, the performance achievable is still quite far from those obtainable training on real data or fine-tuning on few annotated samples.

In this paper we propose a method to automatically create a training dataset for semantic segmentation from real images and we validate it on electric wires by training different segmentation algorithms. To the best of our knowledge this is the first public dataset for semantic wire segmentation, moreover we are the first presenting this method for generating high-quality datasets from real images with minimal human intervention.

Visual perception of Deformable Linear Objects (DLOs), e.g. wires, cables, ropes, etc., has been typically addressed in fairly simple settings. In [23] Augmented Reality markers are deployed to track end-points, while in other works, like [24], detection relies on background removal. Yan at al. [25] developed a more sophisticated method that relies on Gaussian Mixture Models, but it requires the assumption of having a good color contrast between object and background (which has to be homogeneous). The state-of-the-art solution for DLOs detection is presented in [26]. This algorithm, called Ariadne, is based on biased random walks over the Region Adjacency Graph built on a super-pixel over-segmentation of the source image. Unfortunately, this approach has some weakness: it requires an external detector to localize cable terminals; the prediction is intrinsically quite slow due to the exploration process; it can easily fail due to perspective effects or when cables are adjacent.

III. Automatic Dataset Generation

In section I we underlined the importance of a smart solution to collect training data for data-driven models that requires less human intervention possible. In this section, we detail our method and we present a dataset generated for semantic segmentation of electric wires. The proposed strategy employs chroma key to firstly label a set of images and then replacing the background to randomize the domain and enlarge the dataset.

A. Auto-labeling with Chroma Key

The Chroma Key (CK) is a technique widely used in film and motion picture industries to combine two images together (usually foreground and background). It requires a

foreground image I_{fg} containing a target object that we want to overlap to a background image I_{bg}. The target must be placed in front of a monochromatic panel, called screen (usually green or blue). The technique consists of a *chroma-separation* phase, where we isolate the target object (foreground) from the monochromatic panel (original background) and then an *image-overlay* phase, where we compose the foreground and a new background. In the *chroma-separation* phase, we choose a specific hue range which contains solely the color of the screen (e.g. green) and exclude any other color belonging to the foreground. Then, by finding the pixels within that range, we obtain a mask for the target I_{mt} and a complementary mask for the monochromatic background I_{ms}. Thus, creating a dataset with this technique is really straightforward and it can be done in 2 steps:

1) record an high quality video of the target object on a green screen, from which we gather the input images;
2) find the chroma range of the pixels belonging to the monochromatic background and create the corespondent mask with *chroma separation*.

In our dataset, while gathering the images, we hold the electric wire by its extremities and we move it within the frame composing different shapes. To generalize more we also change the light setups, the wire color and the number of wire in the scene. From a random video frame we easily find the hue levels for the specific screen color we are using (green or blue). These levels, once found for one image, remain valid for any other image taken with the same light temperature setting and white balance. Hence, known the chroma range of the screen we immediately obtain the mask for the wire from each frame in the video.

B. Domain Randomization

The labeling procedure with *chroma separation* automatically generates labeled data ready to be used for training, but with a low variability. In fact, in the gathering phase we need to randomize the scene featuring target object in the following aspects: number of instances, color, size, position and shape. Nevertheless, the background is always uniform and monochromatic. The performance of a segmentation algorithm trained with images in homogeneous backgrounds would be significantly degraded when working in a complex and chaotic environment. Clutter background in fact easily confuses the algorithm, due to possible similarities between the target and the background, especially if it has never seen them in training. This weakness can be readily overcome by replacing the background in the input images (*image-overlay* phase). In fact, by using the masks, we can combine the foreground with a random background that replaces the green screen. This process, known as domain randomization [13], [20], aims to provide enough synthetic variability in training data such that at test time the model is able to generalize to real-world data. Hence, the choice of the background images is a key point for generalizing well to multiple real-world target domains without the need of accessing any target scenario data in training.

The backgrounds that we propose for a domain-independent dataset can be divided in 3 categories: (1) lowly textured images with shadows and lights; (2) highly textured images with color gradients and regular or geometric shapes; (3) highly textured images with chaotic and irregular shapes. These backgrounds introduce high variance in the environment properties that should be ignored in the learning task. For instance, in our task the segmentation algorithm will ignore shadows and cubic or spherical objects, while it should focus more in cylindrical shapes, hence we chose the set of backgrounds in Figure 2 according to these considerations.

The presented method introduces two main difficulties that must be faced. The first evident issues of CK concerns the color of the target object. In fact, the color histogram of the object should be well far enough to the range reserved for the screen, or in other words we can not have green wires on a green screen. This implies that the segmentation algorithm never sees green wires in training, thus if it encounters a green wire in a real scene, it would likely produce some false negative. The solution to this issue are two: we can use a different background for the green objects (e.g. a blue screen) or, as we actually do in our dataset, we can randomize the hue of the wire trying to cover also the missing color range (i.e. green). Another issues is caused by the background replacement, which introduces a discontinuity in the synthetic image generated. This may be problematic for the learning, especially in our case with the wires, since the algorithm will probably focus on that sharp feature to segment the object, compromising the prediction in a real image, devoid of the learnt discontinuity. To overcome this issue, the output image I_{out} is obtained according to the following formula

$$I_{\text{out}} = I^{\mathcal{G}}_{\text{mt}} I_{\text{fg}} + (1_{h\times w} - I^{\mathcal{G}}_{\text{mt}}) I_{\text{bg}}. \tag{1}$$

i.e. as a linear combination of the foreground I_{fg} and background I_{bg} images weighted respectively by the target mask processed by a Gaussian filter $I^{\mathcal{G}}_{\text{mt}} = \mathcal{G}(I_{\text{mt}})$ and its complement $(1_{h\times w} - I^{\mathcal{G}}_{\text{mt}})$, where $1_{h\times w}$ is a unit matrix with the same size of the mask.

C. Electric Wire Dataset

The strategy presented in this section has been employed to generate a dataset of 28584 RGB images 720×1280 for semantic segmentation of electric wires. The raw dataset has 3176 images and it includes blue, red, yellow, white and black wires, with different light setups and shapes. To improve the screen and wire separation, besides the hue, we also use the saturation and value channels. For each raw image, a background image (4000×2248) is randomly picked among the 15 shown in Figure 2 and 8 new synthetic images are created, as visible in Figure 1. In each new image, foreground and background are separately augmented (by using the mask) before the merging. In particular, the background is randomly flipped, shifted, scaled and rotated (all with probability $p = 0.5$). Then, it is processed with motion blur and elastic transformation ($p = 0.2$), and in the end it is randomly cropped at 1280×720 ($p = 1$). The foreground, instead, is transformed

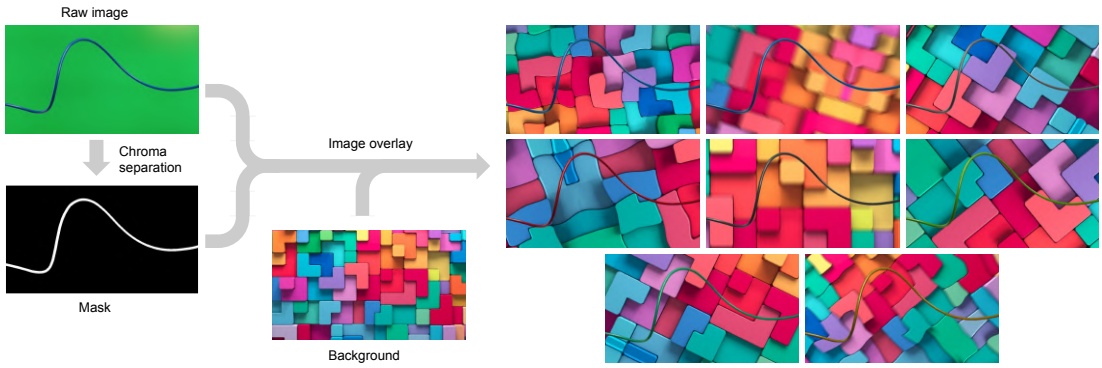

Fig. 1: Schematic process to generate the 8 synthetic images by background-foreground separated augmentation and *image-overlay*.

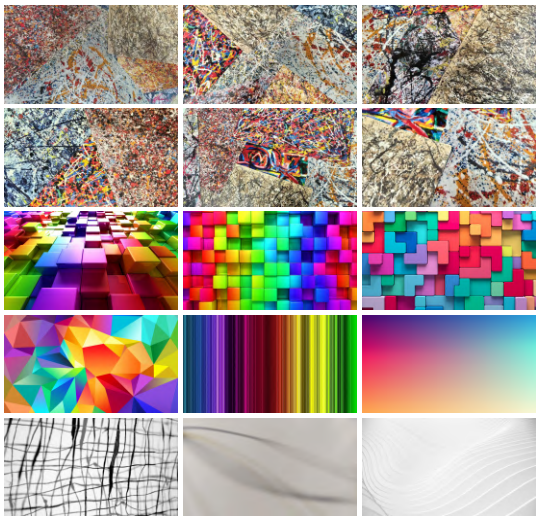

Fig. 2: Images used to replace the background in the output dataset.

only by shuffling the channels ($p = 0.5$), converting to grey ($p = 0.1$) and randomizing the hue in the range of $[-100, 100]$ ($p = 0.5$).

D. Final Considerations on the Output Dataset

The dataset produced by our method contains mainly synthetic images produced by chroma-key overlay. However, the reality gap in the resulting dataset is considerably small compared to those that might be obtained from rendering or simulation. In fact, the main visual discrepancy between real and output images is the object's contour, which has already been smoothed with the combination in eq. (1). To further reduce the gap, we add to the dataset also the input images with the original background, and to avoid over-fitting on green background we randomize the hue and shuffle the channels in training.

The types of background that we suggest to use are intended to make the dataset general purpose and domain-independent. In fact, in the next section we are going to experimentally validate our wires dataset on several scenarios, empirically proving that a set of abstract backgrounds is sufficient to obtain highly satisfactory predictions also in real environments never seen in training. Clearly, to improve the results on a specific real domain, fine-tuning can be also performed. We point out that there are actually two ways to do so. The first is the traditional fine-tuning on a small set of manually labeled images. The second consists in creating a dataset using photos of the specific task environment as backgrounds.

IV. Semantic Segmentation

Two deep learning networks are exploited to perform the training and testing needed to validate our work, namely DeeplabV3+ [27] and HRNet [28].

DeepLabV3+ [27] is an encoder-decoder network which is at the state of the art in deep learning semantic segmentation. It is the last iteration of the famous DeepLab family models. It employes the encoder-decoder structure combined with atrous spacial pyramid pooling (ASPP). As encoder module, DeepLabV3 is used. It is able to encode multi-scale contextual information. The presence of atrous convolutions, instead of the common convolutions, allows the explicitly control the resolution of features computed (via the output stride parameter). For the semantic segmentation task, an output stride of 16 (or 8) is used for denser features. Concerning the decoder, it consists of a simple yet effective module which refines the segmentation results along object boundaries. Here, the low-level features are concatenated to the bi-linearly upsampled (4x) high-level features coming from the decoder. Several convolutions are performed to refine the features and a final upsampling (4x) is performed. Compared to a straightforward one bi-linearly 16x upsampling, the presented decoder module performs much better.

High-Resolution Network (HRNet) [28] is the state of the art in diverse fields such as human pose estimation, semantic segmentation and object detection. It maintains high-resolution representations through the whole network layers by connecting the high-to-low resolution convolution streams in parallel and by repeatedly exchange the information across

TABLE I: The average Dice Coefficient computed for each algorithm, across the images of each test set (*C1*, *C3*, *C4*, *C2*) and the union (*Tot*). In all the tests the predictions are thresholded at 0.5.

Algorithm	*C1*	*C2*	*C3*	*C4*	Tot.
DeepLabV3+	0.928	0.934	0.943	0.935	**0.935**
HRNet	0.923	0.939	0.911	0.926	**0.925**
Ariadne	0.655	0.512	0.632	0.595	0.598

resolutions. The benefit of such structure consists in having a representation semantically richer and spatially more precise.

A. Training and Test

We train DeepLabV3+ with a ResNet-101 backbone for 200 epochs, with batch size 10, output stride 16, separable convolutions, using Adam for the optimization and employing a polynomial learning rate adjustment policy starting from 10^{-6} to a minimum of 10^{-9}, with power 0.95. HRNet is instead initialized with a pretrained model on ImageNet. The network is then trained for 270 epochs, with batch size 6, using SGD for the optimization, weight decay 5×10^{-4}, momentum 0.9, initial learning rate of 1×10^{-5}. The learning rate adjustment policy is polynomial with a power of 0.9. The early stopping in both the training is configured to end the process when the validation loss does not decrease for 5 epochs in a row. Both the models are implemented in PyTorch 1.4.0 and trained with an NVIDIA GeForce GTX 2080 Ti on an Intel Core i9-9900K CPU clocked at 3.60GHz. The data augmentation scheme include hue randomization, channel shuffling, flipping and finally resizing (360×640).

The training dataset is obtained from 90% of the original dataset auto-generated as in section III, while the validation is done on the remaining 10%. To test the algorithms, we use another dataset of 60 manually labeled images collected in different real scenarios. The test dataset is composed by 4 categories of 15 images each:

C1: scenes with only the target wires laying on a surface and no other disturbing objects. The difficulties in this scenes are the high contrast shadows of the wires, possible chroma similarities between the wires and the background, the dense crosses of wires, the light settings and the perspective distortions.

C2: scenes with the target wires only on a highly-featured and complex background and no other disturbing objects. Here the challenge for the algorithms is to extract the correct features belonging to the wires in a cluttered scene.

C3: scenes with the target wires in a realistic industrial setting like an electric panel. These can be considered as an example of an application setting, where the difficulties may be given by metallic surface reflecting the wires and other disturbing objects like commercial electromechanical components characteristic of these panels.

C4: scenes with the target wires in other generic realistic settings among other objects of different nature. The difficulties in these scenes are several and a combination of those found before.

Each algorithm produces a mask M_p which corresponds to the the predicted semantic segmentation of the wire. We evaluate and compare the outputs by means of the Dice coefficient Dice $= 2\frac{|M_p \cap M_{gt}|}{|M_p|+|M_{gt}|}$, where M_{gt} is the ground truth. Table I resumes the average Dice obtained in the test dataset by DeepLabV3+, HRNet and Ariadne [26], state-of-the-art algorithm for DLO segmentation. Ariadne yields a b-spline model for each wire which is here used as predicted mask M_p. In order to make the the comparison with Ariadne more meaningful, we tuned the parameters specifically for the given test dataset and we manually found for each wire the b-spline thickness best fitting with the target. In Figure 3 are visible few example of test images for each category and the outputs of both DeepLabV3+ and HRNet, where true positive ($M_p \cap M_{gt}$), false positive ($M_p - M_{gt}$) and false negative ($M_{gt} - M_p$) are shown in yellow, red and green respectively.

From these tests we can conclude that the auto-generated dataset reaches an high level of reliability ($Dice > 0.9$) for both HRNet and DeepLabV3+ in any scenario without any fine-tuning. More in detail, with reference to Figure 3, we can observe that prospective distortions (*C1*-Sample5, *C2*-Sample 4), color similarities with the background (*C1*-Sample3, *C1*-Sample4), multiple-wire dense intersections (*C1*-Sample2, *C2*-Sample5), wire reflections in metallic surfaces (*C3*-Sample3) and strong shadows (*C1*-Sample2) are all correctly solved using both the algorithms. Moreover, the hue randomization trick used in the foreground images enables the algorithms to correctly recognize also green wires (*C3*-Sample4); whereas the selection of background images allows to effectively segment electric wires in settings never seen in training, very confusing and also with other objects that might look similar to them, such as the table border in *C4*-Sample5 or the handle of the pliers in *C4*-Sample1.

The quantitative results of Table I show that DeepLabV3+ performs on average slightly better than HRNet. In fact, from the qualitative comparison of Figure 3, we observe that the predictions of HRNet are on average a little less confident and sharp at the edges (*C1*-Sample2, *C1*-Sample3). This might be due to an higher sensitivity of HRNet to the reality gap, already discussed in subsection III-B, that we tried to reduce by introducing the Gaussian blur on the mask in eq. (1). However, even thought the predictions of DeepLabv3+ are more precise, it produces evident false negative (like those in *C2*-Sample3) more frequently than HRNet. Table I revels also that both DeepLabV3+ and HRNet trained on our dataset obtain significantly higher performance than the baseline Ariadne.

V. Conclusions

In this paper, we address the problem of recognising and segmenting electric wires from images, which are deformable objects very common in many applications but also lacking

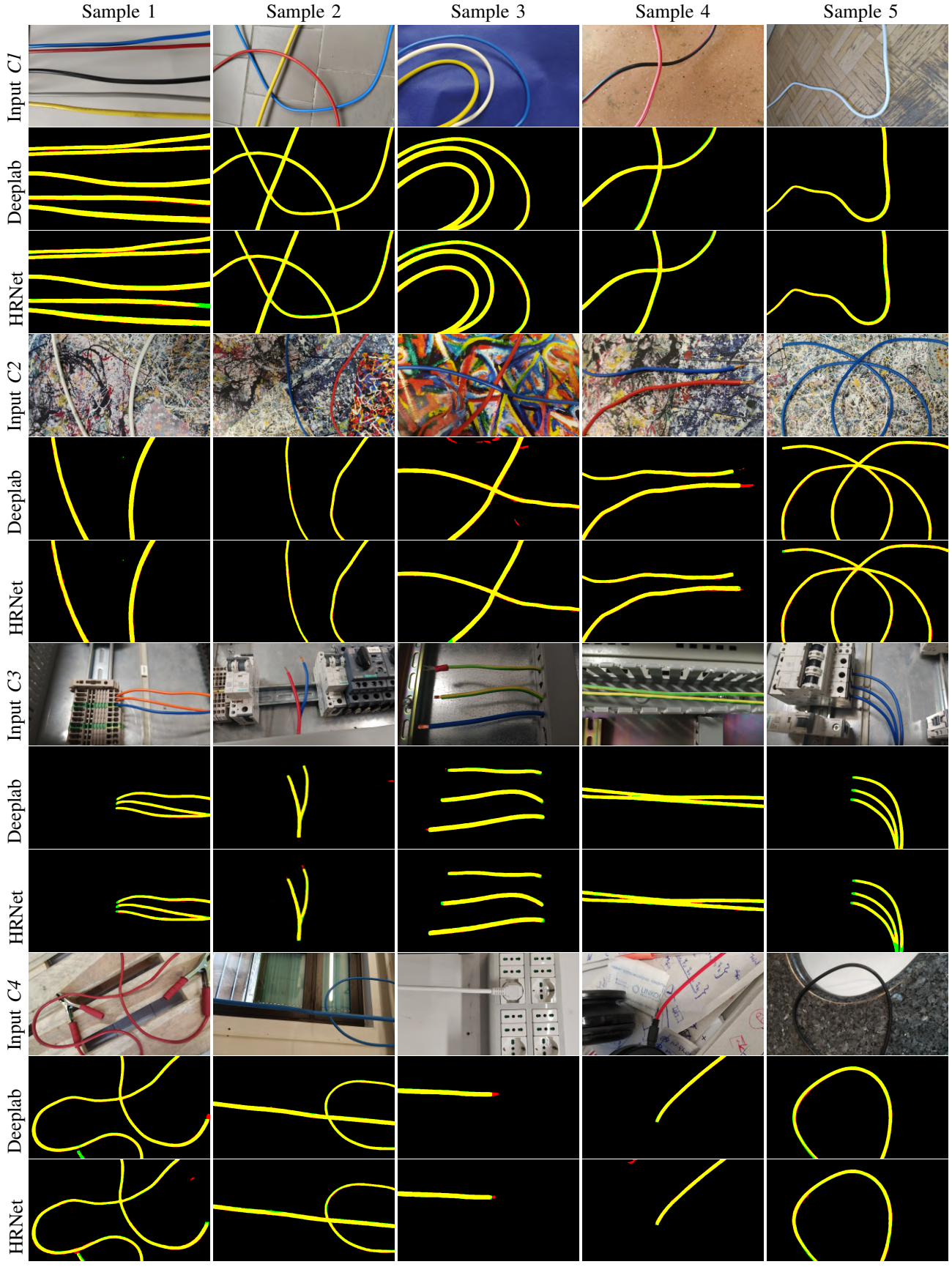

Fig. 3: Qualitative evaluation of DeepLabV3+ and HRNet using 20 sample images from the test set (5 images from each category). The yellow areas are the true positives, the red areas the false positive and green areas the false negative.

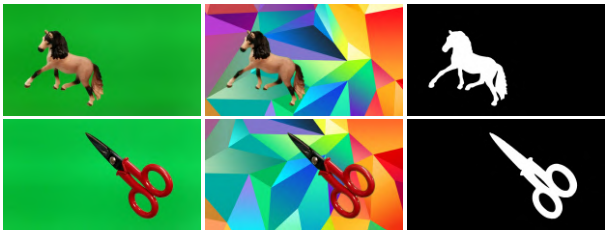

Fig. 4: Sample images from other hypothetical dataset auto-generated with the proposed CK-based technique.

of visual features. A novel strategy to automatically generate a domain-independent dataset has been presented and experimentally validated by training and testing two algorithms, namely HRNet and DeepLabv3+. The experimental results show the effectiveness of the dataset, that enables the segmentation algorithms to correctly recognize the wires in different settings never seen in training with an Average Dice index greater than 0.92. We underline that the presented approach to create the electric-wire dataset can be applied to any other object small enough to be placed and moved in front of a monochromatic panel, like those in Figure 4. In future works, we will formally extend this method to generic objects, then we will also further reduce the human intervention in the *chroma-separation* phase by employing a learning-based methods and improve the *image-composition* phase by reducing the reality gap in the edges.

REFERENCES

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[2] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

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

[4] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.

[5] C. Rother, V. Kolmogorov, and A. Blake, "" grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004.

[6] H. Zhu, F. Meng, J. Cai, and S. Lu, "Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 12–27, 2016.

[7] H. Ramadan, C. Lachqar, and H. Tairi, "A survey of recent interactive image segmentation methods," *Computational Visual Media*, pp. 1–30, 2020.

[8] Z. Lu, H. Xu, and G. Liu, "A survey of object co-segmentation," *IEEE Access*, vol. 7, pp. 62 875–62 893, 2019.

[9] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023.

[10] D. De Gregorio, A. Tonioni, G. Palli, and L. Di Stefano, "Semiautomatic labeling for deep learning in robotics," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 611–620, 2019.

[11] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1386–1383.

[12] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Understanding real world indoor scenes with synthetic data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4077–4085.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[14] J. C. Balloch, V. Agrawal, I. Essa, and S. Chernova, "Unbiasing semantic segmentation for robot perception using synthetic data feature transfer," *arXiv preprint arXiv:1809.03676*, 2018.

[15] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2038–2041.

[16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[17] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," *arXiv preprint arXiv:1511.05547*, 2015.

[18] S. Sankaranarayanan, Y. Balaji, A. Jain, S. Nam Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3752–3761.

[19] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*. Springer, 2016, pp. 102–118.

[20] J. Borrego, A. Dehban, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Applying domain randomization to synthetic data for object category detection," *arXiv preprint arXiv:1807.09834*, 2018.

[21] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3d human pose estimation: motion to the rescue," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 949–12 961.

[22] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6810–6818.

[23] X. Jiang, K.-m. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," *Advanced Robotics*, vol. 25, no. 3-4, pp. 473–489, 2011.

[24] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier, and A. Cherubini, "Dual-arm robotic manipulation of flexible cables," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 479–484.

[25] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[26] D. De Gregorio, G. Palli, and L. Di Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 662–677.

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

[28] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.