Sketch-a-Segmenter: Sketch-Based Photo Segmenter Generation

Conghui Hu, Da Li, Yongxin Yang, Timothy M. Hospedales, *Member, IEEE*, and Yi-Zhe Song, *Senior Member, IEEE*

***Abstract*— Given pixel-level annotated data, traditional photo segmentation techniques have achieved promising results. However, these photo segmentation models can only identify objects in categories for which data annotation and training have been carried out. This limitation has inspired recent work on few-shot and zero-shot learning for image segmentation. In this article, we show the value of sketch for photo segmentation, in particular as a transferable representation to describe a concept to be segmented. We show, for the first time, that it is possible to generate a photo-segmentation model of a novel category using just a single sketch and furthermore exploit the unique fine-grained characteristics of sketch to produce more detailed segmentation. More specifically, we propose a sketch-based photo segmentation method that takes sketch as input and synthesizes the weights required for a neural network to segment the corresponding region of a given photo. Our framework can be applied at both the category-level and the instance-level, and fine-grained input sketches provide more accurate segmentation in the latter. This framework generalizes across categories via sketch and thus provides an alternative to zero-shot learning when segmenting a photo from a category without annotated training data. To investigate the instance-level relationship across sketch and photo, we create the SketchySeg dataset which contains segmentation annotations for photos corresponding to paired sketches in the Sketchy Dataset.**

***Index Terms*— Sketch-based photo segmentation, category-level segmentation, fine-grained segmentation, dataset.**

I. Introduction

PHOTO segmentation is a long-standing computer vision research problem. With new deep learning architectures and training datasets, segmentation results have improved dramatically. Training recent methods [1]–[5] on pixel-level labeled datasets like PASCAL VOC 2012 [6] or MS-COCO [7] has produced excellent results. However, models trained in this way can only be applied to photos depicting objects belonging to the same categories as those annotated in the training set. That is, only photos of categories seen by the network in advance can be segmented. Also, pixel-level labeling for photos is extremely time-consuming. It is extraordinarily difficult to collect segmentation annotation for large-scale datasets such as ImageNet [8]; thus, contemporary segmentation capabilities are limited by the cost of data annotation to a relatively small number of categories.

To address the data limitation issue, a one-shot semantic photo segmentation method was proposed to segment based on only one annotated photo [9]. Zero-shot learning for segmentation [10]–[12] was further considered, in which word-vectors are used to embed training and testing categories and to enable cross-category knowledge transfer for the segmentation of novel categories without annotated training data. While appealing in principle, these approaches depend on extracting word-vector embeddings from text corpora, which means they are applicable only if the category name is both in the dictionary and unambiguous. For categories that are hard to define with a single word or that have ambiguous names, these techniques may not be appropriate.

Recently, sketch, as a complementary modality to text (word-vectors), has been explored in-depth, largely driven by the ubiquitous nature of touchscreens. Details conveyed in sketch have been exploited for fine-grained image retrieval [13]–[16] and image synthesis [17]–[19]. The general consensus is that sketch can be applied to those cases where word-vectors cannot represent the target concept. The fine-grained details in sketch make it a particularly good representation for conveying instance-level details such as posture in addition to category-level information. The advantage of sketch is more obvious when it comes to part-level representation, where parts become increasingly cumbersome or impossible to encode using word-vectors, such as the parts of a machine or the parts of IKEA furniture to be assembled. Our user study shows that rendering sketch is indeed a time-effective way for users to train a segmentation model when compared with the conventional annotation approach – average time required for drawing one sketch (54.84s) vs. annotating one photo segmentation mask (109.01s). (see later in Section IV-D ("User study on sketch rendering time")).

In this article, we condition a photo segmentation model on a sketch that parameterizes the segmentation task. Given an input sketch describing the concept to segment, we synthesize the weights of a photo segmenter for that concept. Related sketch-photo tasks include Sketch-a-Classifier [20], where sketch is used to synthesize a shallow photo classification model. Compared to Sketch-a-Classifier, our sketch-based photo segmentation problem is more challenging in that we

Manuscript received February 7, 2020; revised July 24, 2020 and August 27, 2020; accepted September 27, 2020. Date of publication October 7, 2020; date of current version October 14, 2020. This work was supported by the China Scholarship Council (CSC). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jiaying Liu. *(Corresponding author: Conghui Hu.)*

Conghui Hu, Da Li, Yongxin Yang, and Yi-Zhe Song are with the SketchX Laboratory, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: conghui.hu@surrey.ac.uk).

Timothy M. Hospedales is with the Institute of Perception, Action and Behaviour (IPAB), The University of Edinburgh, Edinburgh EH8 9YL, U.K.

Digital Object Identifier 10.1109/TIP.2020.3028292

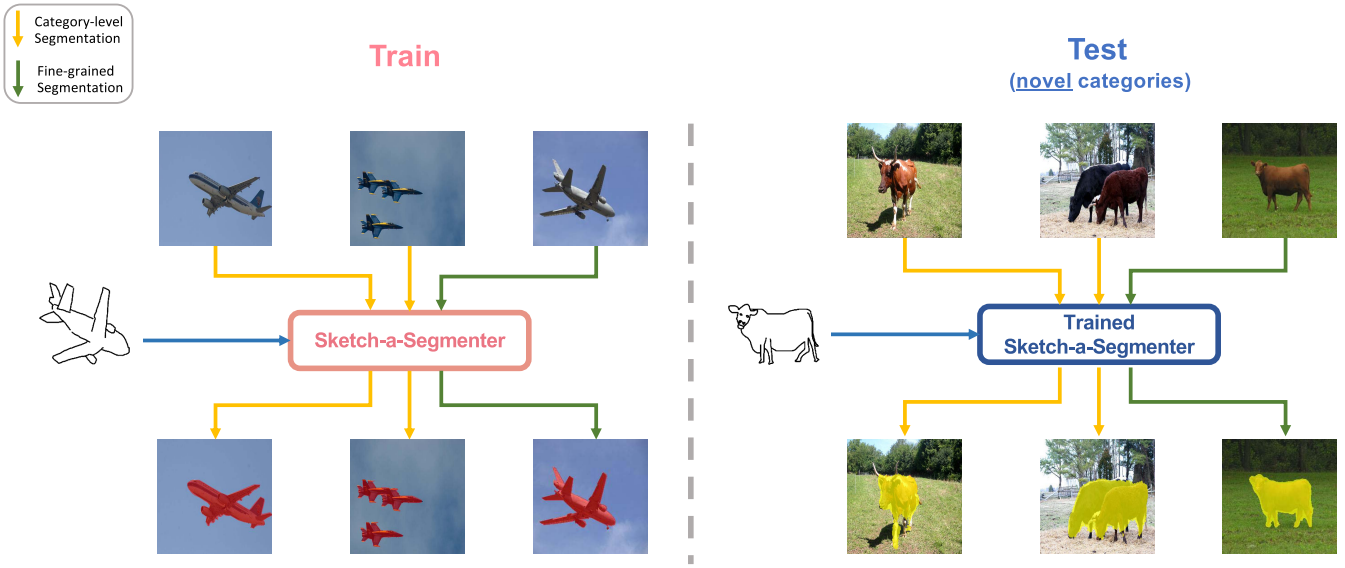


Fig. 1. Sketch-a-Segmenter: Whole object. Given a sketch of an object category (first two columns) or specific instance (third column), we predict the weights for the pixel-wise classification layer in photo segmentation. The trained Sketch-a-Segmenter is tested on novel categories that have not been used for training.

produce a pixel-level classifier as output. In addition, our solution is more sophisticated in that we synthesize the weights for a deep segmentation model rather than a shallow classifier, and our weight generator and photo segmentation model are trained jointly end-to-end rather than with staged training. However, just as Sketch-a-Classifier can produce a novel photo classifier using one sketch, with the proposed Sketch-a-Segmenter, generating a photo segmenter for a novel category is only a sketch away. Fig. 1 provides an illustrative overview of our framework: a *cow* sketch is all it takes to generate a *cow* photo segmenter from a Sketch-a-Segmenter trained on *airplanes*.

We furthermore aim to demonstrate that fine-grained sketch details like the posture and position of the sketch are of benefit to photo segmentation. However, for this purpose, we found no directly suitable existing dataset. Therefore, we annotated photos from the Sketchy dataset [21] with pixel-level ground-truth segmentation. In the original Sketchy dataset, sketches are drawn based on their photo counterparts. Thus, our new SketchySeg dataset provides instance-level sketch-photo-segmentation pairs for training. Compared with category-level segmentation, this instance-level correspondence enables better segmentation models to be generated. Finally, we illustrate the potential of our framework to segment individual object parts, as illustrated in Fig. 2.

Our main contributions are as follows: (i) We make the first attempt to tackle sketch-driven photo segmentation. (ii) We propose a weight synthesis model that generates photo segmentation weights conditioned on one single sketch illustrating the concept to segment. (iii) We contribute the SketchySeg dataset containing pixel-level segmentation annotation for photos with instance-level sketch correspondence. (iv) We analyze the effectiveness of sketch in three different segmentation specificity levels (category-level, fine-grained level, and part-level) and demonstrate promising results.

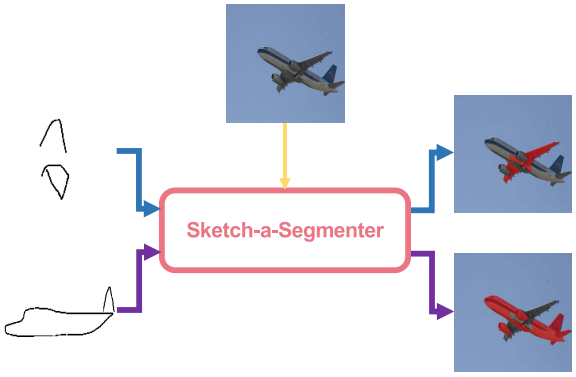


Fig. 2. Sketch-a-Segmenter: Object parts. The segmentation results depend on the part sketch input. For instance, if the sketch contains the front wing of the airplane, only the front wing part in the photo is expected to be segmented out.

II. Related Work

A. Semantic Photo Segmentation

Following the pioneering work [1], deep networks are commonly designed to contain only convolution layers [2]–[5], [22] to avoid the spatial information loss by down-sampling and pooling operations. To obtain sufficient contextual information, atrous convolution, spatial pyramid pooling [23], and multi-scale features [24] are built into networks. Many popular semantic segmentation methods leverage encoder-decoder architecture. Encoder networks are designed similarly to those used for photo classification, without global pooling and fully-connected layers that remove the spatial feature map. The decoder then gradually recovers the feature map to the original image size. In order to keep precise localization information, feature maps from the encoder are combined with their corresponding part in the decoder for prediction [25], [26]. Pooling indices computed in max-pooling layers of the encoder are also

used for up-sampling [27], [28] and to help reduce the number of trainable parameters.

The above photo segmentation methods can only be applied to data depicting objects in the same categories as those used for training. This limitation combined with the time-consuming nature of pixel-level labeling severely restricts the applicability of the conventional approach. To relieve this annotation bottleneck in photo segmentation, one line of work on weakly-supervised photo segmentation methods propose using cheaper annotations at the bounding box [29] or image category [30] level. Another line of work aims to enable segmentation to scale to novel categories more easily via one-shot learning [9].

B. Zero-Shot Learning

Zero-shot learning has been well studied in image recognition context, where annotated images are available for some auxiliary categories but not for testing categories. Knowledge transfer is achieved via category-level representations such as word-vectors [31]–[33] or attribute-vectors [34]. Given such category-embedding, novel categories can be recognized via establishing a cross-domain mapping between photo and category models, or training a verification model to check whether a category embedding is paired with a corresponding photo [31]. Zero-shot learning has also been extended to the task of photo segmentation [10]–[12]. Testing categories are embedded using either word-vector or ImageNet hierarchy as a semantic representation, and the segmentation map is then generated based on the semantic distances between the target category and all the training categories [10], whereas word embeddings are used in [11] to project each image pixel to class probabilities and [12] introduce semantic information in the feature space. These methods are thus restricted to those categories that are clearly nameable or in the ImageNet dataset. In contrast, our category-agnostic sketch-based segmentation model can be used to segment arbitrary visual concepts in images once they can be visualized and depicted – without needing to be nameable or present in existing ontologies such as ImageNet.

C. Model Generation Methods

In our framework, photo semantic segmentation weights are predicted based on the input sketch. This is related to several model synthesis approaches in other areas of machine learning and pattern recognition. In model regression [35], [36], a few-shot input model is regressed to a paired pre-trained many-shot model; thus, the regression network can be used to improve few-shot recognition. More generally, in HyperNetworks [37], parameters for deep network layers are synthesized based on layer index via a jointly-trained hypernetwork; this provides a mechanism for knowledge sharing. This approach has also been applied to few-shot learning [38], [39], where few-shot photos provide a condition for the synthesis of the corresponding category recognition weights. In terms of application, our approach is most related to Sketch-a-Classifier (SAC) [20] that takes sketch as a category representation and synthesizes a photo recognition model. However, SAC solves a category-level rather than a pixel-level problem and operates on a model-regression principle. In terms of methodology, our approach is related to the more sophisticated Hypernetworks in that the (sketch-conditional) weight generator and (photo-segmentation) base model are trained jointly end-to-end.

III. Methodology

A. Problem Definition

Setup: For concepts lacking pixel-level annotated data for training a supervised photo segmenter, sketch can be used to describe the visual concept to be segmented such that it can be used to generate the parameters of the corresponding category-level, fine-grained or part-level segmentation models. To train any of such models, we use N data tuples $\{(\mathbf{s}, \mathbf{p}, \mathbf{m})\}_{i=1}^{N}$ of sketch $\mathbf{s}$, photo $\mathbf{p}$ and segmentation mask $\mathbf{m}$. These tuples are used to train a segmentation model $\mathcal{S}_W(\mathcal{F}_\Theta(.))$, where $\mathcal{F}_\Theta$ is a photo feature extractor with parameters Θ and $\mathcal{S}_W$ is the pixel-level classifier with parameters W. Unlike parameters Θ, segmentation weights W are synthesized via a hypernetwork $W = \mathcal{H}_V(.)$ that inputs a sketch feature and is parameterized by V. Thus, the whole segmentation model inputs a photo and segments it according to the supervisory sketch to produce a segmentation mask as:

$$\mathbf{m} = \mathcal{S}_W(\mathcal{F}_\Theta(\mathbf{p}))$$
$$W = \mathcal{H}_V(G_\Phi(\mathbf{s}))$$

where G_Φ is a sketch feature extractor, and the sketch defines the region of photo to segment. The weights W are synthesized for each segmentation task, while other parameters V, Θ, Φ are task-agnostic.

Sketch-Based Category-Level Segmentation: In category-level segmentation, the photo $\mathbf{p}$ and segmentation mask $\mathbf{m}$ are in exact correspondence, while the sketch $\mathbf{s}$ only needs to have a category-level correspondence to the segmentation mask of the photo. This is analogous to the granularity of zero-shot learning, where the task is specified at category-level. We require the weight synthesis hypernetwork to generalize across categories so that at testing time, photos of novel categories can be segmented by providing a category-level corresponding sketch to synthesize segmentation weights.

Sketch-Based Fine-Grained Segmentation: Within a single object category, shape, posture and position of an object can vary. This makes the required procedure of mapping from sketch to segmentation hard to learn if training tuples only have a category-level correspondence. If fine-grained training data is available that matches photos with exactly corresponding sketches, a better sketch-conditional segmentation model can be learned. Thus, in this case, training tuples $(\mathbf{s}, \mathbf{p}, \mathbf{m})$ are in instance-level correspondence. It is expected that sketch-based photo segmentation will produce more accurate results in this case.

Sketch-Based Part-Level Segmentation: In this problem setting, a specific part of the object is required to be segmented (e.g., body parts or parts of a vehicle). In this case, sketch can indicate the required part. This is potentially convenient because there are many parts and subgroups of parts compared to full object categories, and the ability to specify a part to

segment on the fly without going through the loop of collecting annotated segmentation masks and re-training models is potentially valuable.

B. HyperNet-DeepLabv3+ Network

We extend the DeepLabv3+ [5] segmentation network to include a Hypernetwork [37] that reads the desired input from the sketch domain and synthesizes weights in DeepLabv3+ accordingly at three different levels: (i) category-level: to explore the effect of sketch as a transferable representation in generating a photo segmenter for photos of the same category; (ii) fine-grained level: to utilize the appearance and spatial alignment between sketch and photo for fine-grained segmentation model parameter synthesis; and (iii) part-level: to employ object part sketches as a pre-condition for part-level photo segmentation.

Sketch-Based HyperNet Module: To generate weights for photo segmentation, our HyperNet takes sketch features as a description of the desired concept to segment, and specific weights and biases for the corresponding segmentation are synthesized accordingly. While traditional semantic segmentation models use a globally shared set of convolutional layers and pixel-wise classifiers, we remove the pixel-wise classifier weights and plug in the output of the HyperNet in their place. Thus, the HyperNet dynamically synthesizes the classifier weights for the segmentation network. If the sketch-based HyperNet module is parameterized with V, then the weight generation model $\mathcal{H}_V(.)$ is designed to generate the weights and biases for the pixel-wise classification layer in the photo segmentation model.

$$W_w, W_b = \mathcal{H}_V(G_\Phi(\mathbf{s})) \tag{1}$$

where $G_\Phi(.)$ is the sketch feature extractor. The size of the generated weights W_w and biases W_b are $k \times k \times c \times 2$ and 2 accordingly. Here k is the kernel size for the pixel-wise classification layer for segmentation, and c corresponds to the number of channels in the input photo feature map at the pixel-wise classification layer. The final dimension is 2, as each pixel in the output segmentation map is classified as either foreground or background.

Photo Feature Extractor Module: The photo feature extractor module takes the photo for segmentation as input. The state of the art semantic segmentation model – DeepLabv3+ [5] is exploited, up to its penultimate layer. Input photos are first encoded to a low-resolution feature map with the network backbone used for photo classification. The following decoder part then refines the feature map to a higher resolution. The function of the photo feature extractor module is to extract the feature map from the input photo for later segmentation with the weights and biases predicted from the HyperNet module. Θ represents the parameters in the photo feature extractor module $\mathcal{F}$. The model learns to extract the feature map

$$\mathbf{f} = \mathcal{F}_\Theta(\mathbf{p}) \tag{2}$$

where $\mathbf{p}$ and $\mathbf{f}$ are the input photo and extracted feature map for $\mathbf{p}$ respectively. $\mathbf{f}$ is of size $h \times w \times c$, corresponding to height, width, and depth of the photo feature respectively.

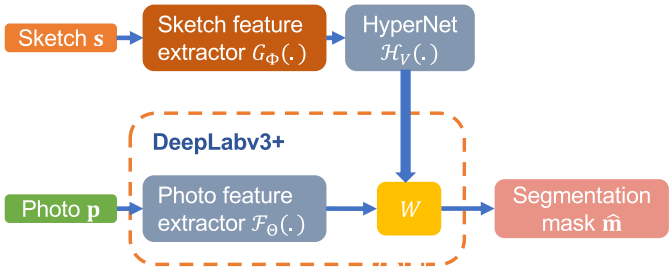


Fig. 3. Network architecture for Sketch-a-Segmenter. The gray modules (HyperNet and Photo feature extractor) are trained together for segmentation.

It will then be fed into the pixel-wise classification layer whose weights and biases are generated by the sketch-based HyperNet module.

Sketch-Based Segmentation: The pixel-wise classification layer of our model exploits the weights and biases (W_w and W_b) generated by the HyperNet module $\mathcal{H}$ from the input sketch feature, and uses these to classify each pixel in the input photo $\mathbf{p}$, as encoded into $\mathbf{f}$ by the photo feature extractor module $\mathcal{F}$. The final prediction $\hat{\mathbf{m}}$ is generated by

$$\hat{\mathbf{m}} = \sigma(\mathcal{U}(W_w \otimes \mathbf{f} + W_b)) \tag{3}$$

where, $\otimes$ indicates convolution, $\mathcal{U}$ is the bilinear up-sampling operation that ensures the height and width of the predicted results are the same as the input photo for pixel-level segmentation. $\sigma(.)$ is the softmax function.

C. Architecture Details

Our Sketch-a-Segmenter architecture is illustrated in Fig. 3. The HyperNet module architecture is the same for all three category-level, fine-grained and part-level segmentation tasks: One fully-connected layer to encode the sketch feature vector and generate weights and biases for photo segmentation.

D. Objective Function

We train the network to generate binary (foreground vs. background) segmentation maps. A weighted two-class pixel-level cross-entropy loss is calculated and used to train both the $\mathcal{H}$ and $\mathcal{F}$ modules together,

$$\mathcal{L} = -\sum w_{p_{(i,j)}} \mathbf{m}_{p_{(i,j)}} \log(\hat{\mathbf{m}}_{p_{(i,j)}}) \tag{4}$$

Here, $p_{(i,j)}$ represents the pixel in row i and column j, and $\mathbf{m}_{p_{(i,j)}}$ is the ground-truth label for that pixel. In order to compensate for imbalanced class frequency, we define a pixel-wise weight $w_{p_{(i,j)}}$ which will vary according to background/target-object pixel.

IV. Experiments

A. Datasets and Settings

Dataset: Sketch-Based Category-Level Segmentation: For this experiment, we combine photos from 15 categories in the PASCAL VOC 2012 dataset that have corresponding sketch categories in the Sketchy dataset. The mapping between the 15 categories in the PASCAL VOC 2012 and Sketchy dataset

TABLE I
CATEGORY CORRESPONDENCE BETWEEN THE PASCAL VOC AND SKETCHY DATASET

Photo category in PASCAL VOC	Sketch category in Sketchy
bird	songbird
dog	dog
horse	horse
sheep	sheep
aeroplane	airplane
bicycle	bicycle
boat	sailboat
bottle	wine_bottle
dining table	table
sofa	couch
cat	cat
cow	cow
car	car_(sedan)
motorbike	motorcycle
chair	chair

[21] is listed in Table I. The training photo set is enlarged with photos from the same category in the Semantic Boundaries Dataset (SBD) [40]. The test photos are all from the PASCAL VOC 2012 validation set. In order to test cross-category generalization, we split the 15 categories into 10 training and 5 testing categories.

Dataset: Sketch-Based Fine-Grained Segmentation: Unlike sketch-based category-level segmentation, where sketches and photos are paired at the category level during training, we can exploit the instance-level correspondence for sketch-based fine-grained segmentation. In existing segmentation datasets like PASCAL VOC 2012 or MS-COCO, there is no corresponding sketch for each photo. To create a sketch-photo instance-level paired segmentation dataset, we select photos from the Sketchy Dataset that match those categories used in the category-level experiments and then annotate their segmentation ground truth with the LabelMe toolbox [41]. Since sketches in the Sketchy dataset are drawn according to a specific photo, after annotating these photos with segmentation masks, we now have $(\mathbf{s}, \mathbf{p}, \mathbf{m})$ tuples paired at the instance level. In total, our new SketchySeg dataset contains 15 annotated categories as listed in Table I and 100 photos with pixel-level annotations for each category. Some examples are shown in Fig. 4.

Dataset: Sketch-Based Part-Level Segmentation: Among these datasets, this is the hardest one to collect, as part-level annotation is rare in both sketch and photo domain. We use the overlapped category (airplane) of the existing sketch perceptual grouping (SPG) dataset [42] and PASCAL-Part dataset [43]. For airplanes in the SPG dataset, the sketch strokes are annotated with 4 part labels: front wing, body, window and tail wing. The frequency of occurrences of the window and tail wing is very low compared with the front wing and body which exist in nearly every sketch. So we then combined the body, window and tail wing together as body. Airplane photo parts in the PASCAL-Part dataset are also grouped accordingly.

Feature Extraction: For sketch feature extraction, MobilenetV2 [44] pre-trained using image data for the 1000-category ILSVRC is fine-tuned with the Sketchy dataset (excluding the 5 test categories). The same fine-tuned MobilenetV2 model is used for feature extraction for all sketches. Photo features used for weight prediction are extracted by exploiting the model fine-tuned with photos (excluding the 5 test categories) in the Sketchy dataset. The 1280-d feature after global pooling is used in all category-level, fine-grained and part-level segmentation experiments.

Training Settings: The same learning policy as [5] is applied for training with an initial learning rate of 0.007. Mini-batch size in all experiments is 32. For PASCAL and PASCAL-Part photos used in the category-level and part-level experiments, the image crop size is 513×513. While in fine-grained segmentation, the image crop size for photos in the Sketchy dataset is 256×256. We assign a higher weight in Eq. 4 for the target-object than for background pixels (1.0 vs. 0.1) due to their imbalanced proportions.

B. Models for Comparison

Random segmentation: A lower bound. Generates a segmentation mask by randomly labeling each pixel.

Average mask segmentation: A more reasonable lower bound. The average segmentation mask is calculated based on all the ground-truth marks in the training set. It is applied to all test photos and used to calculate segmentation accuracy. We try thresholds [0.1,0.2,..0.9] to generate a binary segmentation map and then select the best result from all 9 cases.

DeepLabv3+: We feed the same photos into DeepLabv3+ [5] for segmentation but use a single pixel-wise classifier for all categories; thus, DeepLabv3+ is trained for generic binary foreground/background segmentation using the training categories.

SPNet [11]: For fair comparison, we implement SPNet zero-shot segmentation with our photo feature extractor architecture. Word-vectors are used as category representation as per Hyper-DeepLabv3+Wordvec.

ZSVM [12]: The same encoder-decoder architecture as DeepLabv3+ is employed for photo segmentation. However, before being concatenated with the encoded image feature map, word-vectors are projected to the feature space via a variational mapping mechanism.

OSLSM [9]: Photo segmentation is accomplished by conditional weight prediction on the pixel-wise classification layer in DeepLabv3+. Outputs from the MobilenetV2 logits layer for one masked photo in the Sketchy dataset are then treated as a pre-condition for parameter generation.

DeepLabv3+Sketch feature: Instead of being regarded as an input for weight synthesis, the sketch feature is directly concatenated with the photo feature map before pixel-wise prediction. An additional fully-connected layer is used to reduce the 1280-d sketch feature to 256-d (the same as the number of photo feature map channels).

DeepLabv3+Sketch: Unlike the original DeepLabv3+, both a photo and its category-level, fine-grained level or part-level corresponding sketch are depth concatenated and fed into the network for photo segmentation map prediction. The sketch is resized according to the image crop size in each experiment.

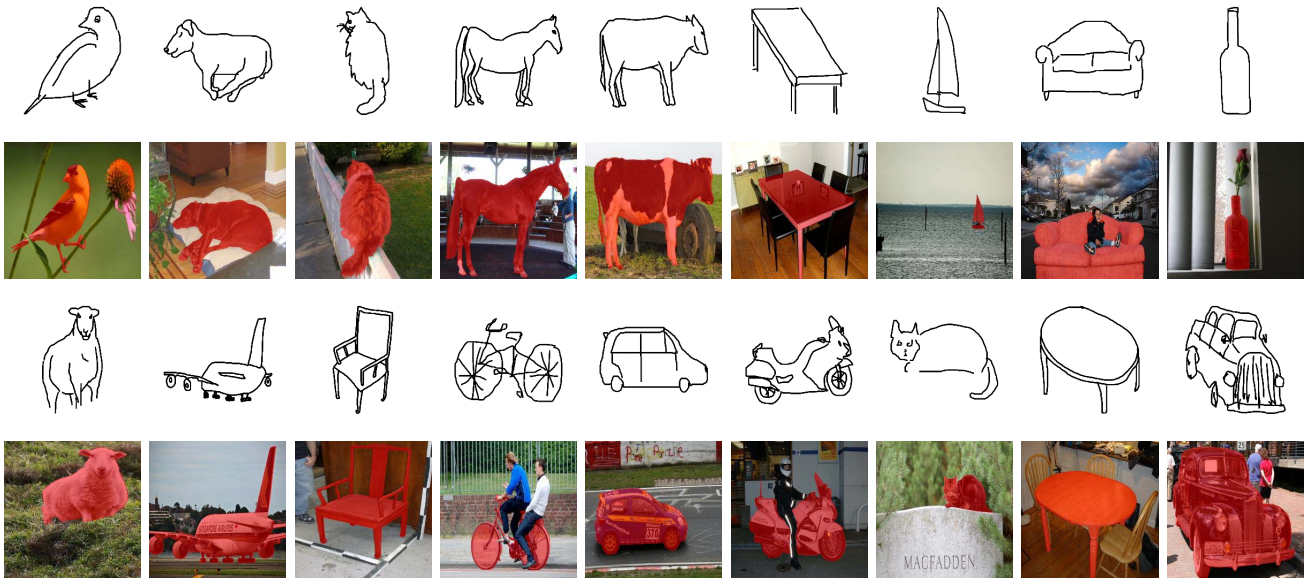


Fig. 4. Examples from SketchySeg dataset. Row1, 3: Sketch in the Sketchy dataset. Row 2, 4: Annotated segmentation map for photo.

Hyper-DeepLabv3+Wordvec: A language-based zero-shot learning architecture. The DeepLabv3+ pixel-wise classification layer parameters are synthesized by the HyperNet module taking a word-vector as input. There is one word-vector per category, so the category-specific weights are generated according to the input word-vector. One 300-d vector is extracted for each category by using the word2vec model pre-trained on the Google News corpus [45].

Hyper-DeepLabv3+Photo: In this baseline, one photo feature is provided to the HyperNet module and used to predict the weights and biases. For category-level segmentation, Sketchy dataset photos are used to generate weights for photo segmentation in the PASCAL VOC 2012 dataset. For fine-grained segmentation, there is no other instance-level paired photo for photos in the Sketchy dataset, so the same input is used for feature extraction before HyperNet and photo segmentation.

Hyper-DeepLabv3+Sketch: Our full model, which takes sketch as input for category-level, fine-grained and part-level segmentation and generates the corresponding weights for segmentation. The DeepLabv3+ architecture is used for all the segmentation problem variants.

C. Results

Sketch-Based Category-Level Photo Segmentation:

Settings: Of the 15 overlapped categories in Table I, we use 10 (bird, dog, horse, sheep, aeroplane, bicycle, boat, bottle, dining table, sofa) for training and the rest for testing. During training, within each mini-batch, sketches are randomly selected as the input for the HyperNet module. For each sketch, one photo of the same category is fed into the DeepLabv3+ branch. The final loss is calculated based on the predicted segmentation map for the input photos. The same sketch-photo pairing strategy is used in testing. To increase the reliability of test segmentation accuracy, all photos in the set of testing categories are used for testing, and 5 different inputs from the corresponding categories are randomly selected for the HyperNet module. Specifically, each photo is segmented 5 times based on the weights and biases generated from the 5 different sets of sketches. We explore whether it is beneficial to use more than one sketch to describe the segmentation task and exploit 1, 3 and 5 sketches per task. For multiple sketches, the element-wise average of all sketches is calculated to obtain the final feature vector for classifier weight synthesis. Each photo in the test set is tested 5 times against the 5 category-level paired inputs to the HyperNet module apart from the experiment using word-vectors for weight prediction, as there is only one word-vector per category.

Evaluation Metrics: We use the standard mean Intersection-Over-Union (mIOU) and pixel accuracy for evaluation. All pixels belonging to the same category as the input of the HyperNet are regarded as foreground and the rest as background. Experiments for category-level segmentation are repeatedly retrained 3 times, and the final mIOU and pixel accuracy reported are the averaged values (except the Average mask segmentation because there is only one average mask for training categories).

Results: The results are shown in Table II, from which we can draw the following conclusions: (i) The Average mask baseline performs better than the Random segmentation baseline as expected. (ii) The binary foreground segmentation model trained on training categories can to some extent predict foreground vs. background even when applied to novel categories. Its mIOU is clearly superior to the average segmentation mask. (iii) Compared with the DeepLabv3+ binary segmentation results, performance is improved with any auxiliary information about the target concept to segment, either photo, word-vector or sketch. (iv) Provided with the same word-vectors as category representation,

Fig. 5. Category-level segmentation results. Row 1: Input sketch for Hyper-DeepLabv3+Sketch (1). Row 2: Input photo for Hyper-DeepLabv3+Photo. Row 3: Segmentation results of DeepLabv3+. Row 4: Segmentation results of Hyper-DeepLabv3+Photo. Row 5: Segmentation results of Hyper-DeepLabv3+Wordvec. Row 6: Segmentation results of Hyper-DeepLabv3+Sketch (1). Row 7: Ground-truth segmentation map.

TABLE II
CATEGORY-LEVEL SEGMENTATION RESULTS: SKETCHY AND PASCAL VOC DATASETS. SKETCH/SKETCH FEATURE (#): NO. OF INPUT SKETCHES USED

Method	mIOU (%)	Pixel Accuracy (%)
Random segmentation	30.46 ± 0.00	50.00 ± 0.00
Average mask segmentation	47.72 ± 0.00	76.84 ± 0.00
DeepLabv3+ [5]	63.83 ± 0.73	83.52 ± 0.56
SPNet [11]	65.31 ± 1.10	84.78 ± 0.62
ZSVM [12]	66.17 ± 0.45	85.83 ± 0.57
OSLSM [9]	65.40 ± 2.36	84.36 ± 1.94
DeepLabv3+Sketch feature (1)	64.38 ± 0.13	84.22 ± 0.25
DeepLabv3+Sketch (1)	67.54 ± 0.49	86.25 ± 0.29
Hyper-DeepLabv3+Photo	66.93 ± 0.44	86.72 ± 0.28
Hyper-DeepLabv3+Wordvec	70.84 ± 0.16	89.30 ± 0.17
Hyper-DeepLabv3+Sketch (1)	71.71 ± 0.45	89.00 ± 0.62
Hyper-DeepLabv3+Sketch (3)	72.16 ± 0.16	89.78 ± 0.08
Hyper-DeepLabv3+Sketch (5)	**73.35 ± 0.24**	**90.24 ± 0.27**

Hyper-DeepLabv3+Wordvec achieves better segmentation results than SPNet and ZSVM. (v) Photo-based Hyper-DeepLabv3+ is worse than sketch- or word-vector-based segmentation. (vi) With HyperNet, one sketch already provides a superior task description compared to word-vector. (vii) When provided with more (3 or 5) sketches to describe the task in HyperNet-DeepLabv3+, the results continue to improve. (viii) Hyper-DeepLabv3+ is a better way to drive photo segmentation with sketch compared with DeepLabv3+Sketch feature and DeepLabv3+Sketch. Qualitative comparisons between DeepLabv3+, Hyper-DeepLabv3+Wordvec, Hyper-DeepLabv3+Photo and HyperDeepLabv3+Sketch (1) are shown in Fig. 5, and Fig. 6 shows examples where two different categories occur in the same photo. Providing different sketches as input drives the segmentation to attend to different parts of the photo.

Sketch-Based Fine-Grained Photo Segmentation:

Settings: The train/test category split is the same as that for category-level segmentation. However, training data can be *instance-level* pairs of sketches and photos/masks . In the Sketchy dataset, there are multiple sketches drawn for the same photo, so each photo can be paired with more than one sketch even when training or testing with instance-level corresponding pairs. But in the photo baseline, only the input

Fig. 6. Category-level segmentation results of Hyper-DeepLabv3+Sketch (1) in multi-object scenes. Row 1: Input sketch; Row 2: Segmentation results; Row 3: Ground-truth segmentation map. Different input sketches lead to different segmented regions.

TABLE III

FINE-GRAINED SEGMENTATION RESULTS: SKETCHYSEG DATASET. SKETCH-BASED METHODS ARE TESTED WITH (C)ATEGORY-LEVEL, (I)NSTANCE-LEVEL AND (P)OSITION-LEVEL MATCHED SKETCHES

Method	mIOU (%)	Pixel Accuracy (%)
Random segmentation	31.91 ± 0.00	50.00 ± 0.00
Average mask segmentation	57.79 ± 0.00	76.92 ± 0.00
DeepLabv3+ [5]	74.35 ± 0.46	88.17 ± 0.26
Hyper-DeepLabv3+Photo	72.56 ± 0.13	87.22 ± 0.11
DeepLabv3+Sketch feature (C)	72.39 ± 0.18	87.31 ± 0.10
DeepLabv3+Sketch feature (I)	72.91 ± 0.21	87.58 ± 0.10
DeepLabv3+Sketch feature (P)	74.00 ± 0.16	88.11 ± 0.09
DeepLabv3+Sketch (C)	74.97 ± 0.46	87.97 ± 0.31
DeepLabv3+Sketch (I)	76.78 ± 0.56	88.77 ± 0.36
DeepLabv3+Sketch (P)	77.32 ± 0.73	88.95 ± 0.46
Hyper-DeepLabv3+Sketch (C)	79.13 ± 0.20	90.32 ± 0.11
Hyper-DeepLabv3+Sketch (I)	80.48 ± 0.43	90.92 ± 0.23
Hyper-DeepLabv3+Sketch (P)	**80.96 ± 0.39**	**91.14 ± 0.21**

photo itself can be regarded as the task descriptor for weight generation.

Three different types of sketches are used for evaluating the impact of the degree of fine-grained sketch-photo alignment. **C**ategory: Coarse-grained category-level pairing. **I**nstance: A sketch of the corresponding photo instance, but without position and scale alignment. **P**osition: The corresponding instance sketch, aligned to the correct position and scale. To predict segmentation results based on one type of sketch, we randomly select 5 sketches of the corresponding type per photo to formulate the testing set. Similar to category-level segmentation, each photo in the test set of SketchySeg is also segmented 5 times with 5 different auxiliary sketch sets.

Evaluation Metrics: The same mIOU and pixel accuracy metrics are used but ground-truth is defined differently, as only the pixels of the corresponding instance are set as foreground, while the others are defined as background. The reported results are also averaged over 3 repetitions except the Average mask segmentation.

Results: From the results in Table III we can see: (i) Similar to category-level segmentation, binary DeepLabv3+ achieves clearly better results than the Random segmentation and Average mask segmentation baselines. (ii) Sketch-based description of the object to segment provides improved performance compared to a generic binary foreground segmenter in all DeepLabv3+Sketch and sketch-based Hyper-DeepLabv3+ experiments. (iii) Weight-synthesis is a superior approach to sketch-driven segmentation (Hyper-DeepLabv3+Sketch vs. DeepLabv3+Sketch feature and DeepLabv3+Sketch) for all three types of sketches. (iv) Given the same Hyper-DeepLabv3+Sketch model trained with instance-level paired data and identical test photos from Sketchy dataset, using position+instance aligned (P) sketches at run-time provides improved performance compared to randomly selected coarse-grained (C) and instance-level paired (I) inputs in all sketch-aided experiments. Note that the distinction between the coarse and fine-grained settings corresponds to different applications where the user knows (I/P) or does not know (C) the expected appearance details and pose of the object to be segmented. Qualitative results are shown in Fig. 7.

Sketch-Based Part-Level Photo Segmentation:

Settings: The goal here is to sketch a certain object part and then generate a segmenter for that part in photos. Existing part-segmentation databases are insufficient to do cross-part-category evaluation as described in the previous sections. Therefore, we present this as an illustrative proof of concept only.

For airplane sketches in the SPG dataset, we use 600 for training and the remaining 200 for testing. For the photos, 100 airplane photos in the PASCAL-Part dataset are exploited for testing, and the remaining photos are used for training. All photos have three different kinds of annotations: front wing annotated only, body (including all parts of the airplane except the front wing) only, and whole airplane. Sketches with a given part are paired with the corresponding photo annotation. We use the same MobileNetV2 model fine-tuned with the Sketchy dataset for feature extraction for sketches in the SPG dataset; thus, none of the test sketches are used during training.

Evaluation Metrics: Standard mIOU and pixel accuracy are also used as evaluation metrics. The photo annotation ground-truth is adjusted based on the part indicated in the input sketch. For instance, when using the front wing part sketch as input, the foreground pixels in the ground-truth photo segmentation map contain only the front wing part.

Results: As shown in Fig. 8 and Table IV, the object part segmented can be controlled by inputting different part sketches. (i) Since the training and test category in this experiment are both "airplane", the Average mask segmentation results are relatively better than the previous category-level or fine-grained segmentation. (ii) In both DeepLabv3+Sketch feature and DeepLabv3+Sketch, the auxiliary sketch information helps improve the photo segmentation accuracy when compared with the lower bounds. (iii) Similar to the category-level and fine-grained segmentation results, our proposed sketch-based Hyper-DeepLabv3+ is a better approach for sketch-driven photo segmentation. With the help of a part sketch alone, our model can segment the corresponding photo

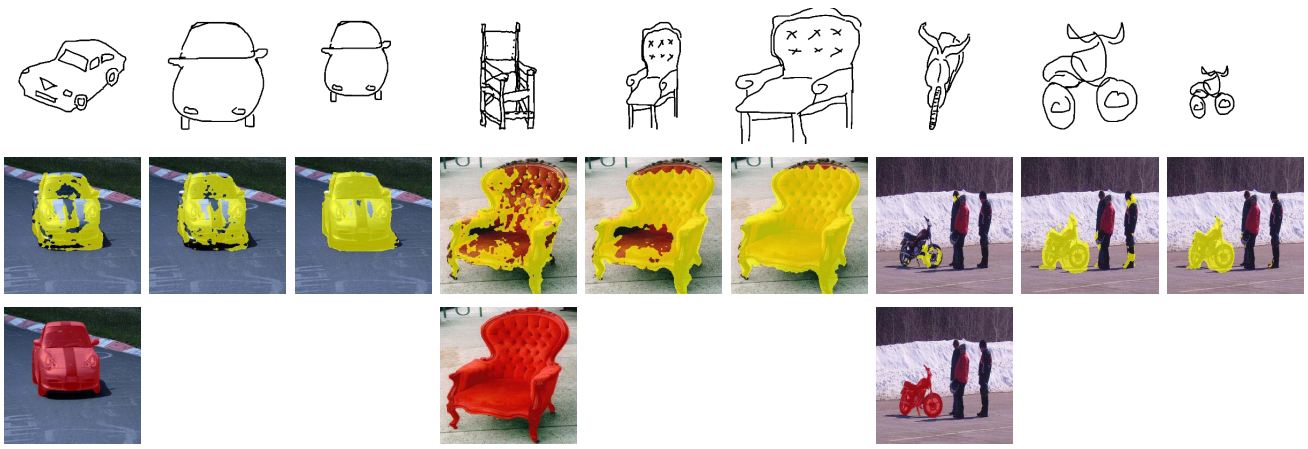


Fig. 7. Fine-grained segmentation results of Hyper-DeepLabv3+Sketch. Row 1: Different input sketch types with pairings at Category-level (Col. 1, 4, 7), Instance-level (Col. 2, 5, 8), and Position-level (Col. 3, 6, 9). Row 2: Segmentation results. Row 3: Ground-truth segmentation map.

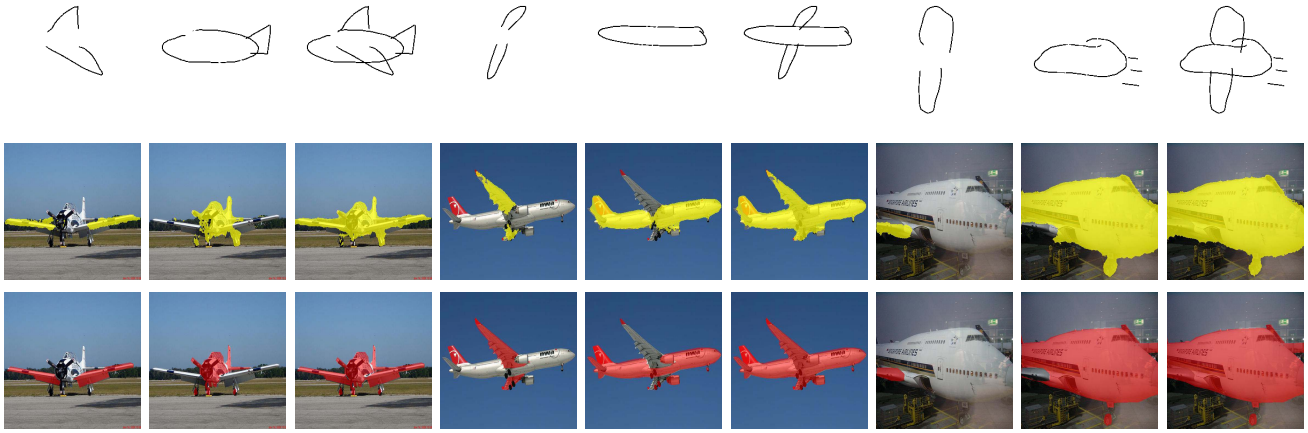


Fig. 8. Part-level segmentation results of Hyper-DeepLabv3+Sketch. Row 1: Part of sketch indicated; Row 2: Segmentation results; Row 3: Ground-truth segmentation map.

TABLE IV
PART-LEVEL SEGMENTATION RESULTS: SPG AND PASCAL-PART DATASETS

Method	mIOU (%)	Pixel Accuracy (%)
Random segmentation	27.97 ± 0.00	50.00 ± 0.00
Average mask segmentation	58.19 ± 0.00	89.41 ± 0.00
DeepLabv3+Sketch feature	70.08 ± 0.03	92.44 ± 0.02
DeepLabv3+Sketch	74.28 ± 0.48	93.92 ± 0.29
Hyper-DeepLabv3+Sketch	**79.71 ± 0.08**	**95.63 ± 0.07**

part with 79.71% and 95.63% mIOU and pixel accuracy, respectively.

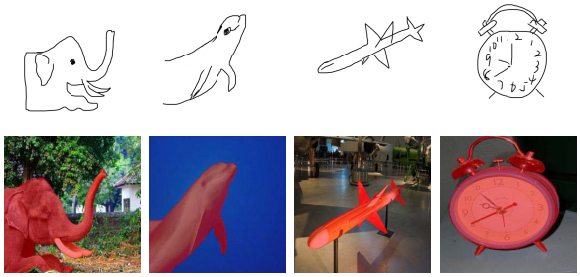


Fig. 9. User study examples. Sketches (above) and segmentation maps (below) annotated by participants.

D. Further Analysis

1) User Study on Sketch Rendering Time: To verify the feasibility of sketch-aided photo segmentation, we conduct a user study to compare the time required for drawing one sketch against labeling one photo segmentation map. First, 50 photos are randomly selected from the Sketchy dataset [21] for each of the participants. We then ask 5 participants to annotate the segmentation map for specified objects in those photos using the LabelMe toolbox [41]. This results in 250 annotated segmentation maps, with an average annotation time of 109.01s. The same participants are then asked to draw a sketch for the same objects from each of the 50 photos using an iPad. The 250 total sketches require an average time of 54.84s. Example segmentation maps and sketches rendered in the user study are shown in Fig 9. The results suggest that sketches certainly represent a time-saving approach for users to train segmenters when compared with manual annotation.

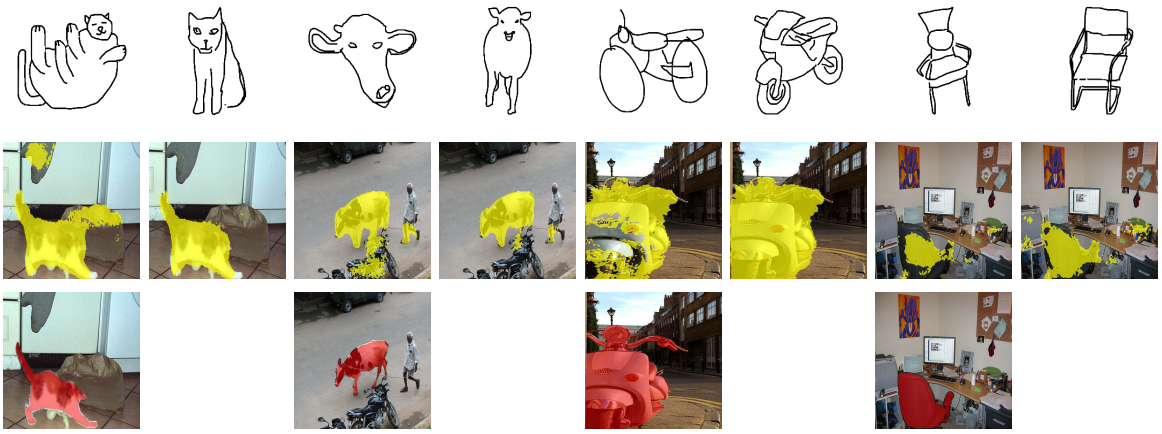


Fig. 10. Influence of sketch quality on category-level segmentation. Row 1: Different input sketches ranking among the Bottom-10 (Col. 1, 3, 5, 7) and Top-10 (Col. 2, 4, 6, 8) quality. Row 2: Segmentation results. Row 3: Ground-truth segmentation map.

Fig. 11. Category-level segmentation results of Hyper-DeepLabv3+Sketch (1) when tested with sketches from TU-Berlin dataset. Row 1: Input sketch. Row 2: Segmentation results. Row 3: Ground-truth segmentation map.

TABLE V
INFLUENCE OF SKETCH QUALITY ON CATEGORY-LEVEL SEGMENTATION

Sketch Quality	mIOU (%)	Pixel Accuracy (%)
Bottom-10	70.43 ± 0.34	88.50 ± 0.58
Top-10	**71.92 ± 0.69**	**89.21 ± 0.57**

2) Influence of Sketch Quality: Here, we evaluate whether the quality of sketch has an effect on weight prediction. MobilenetV2 pre-trained with 1000-category ILSVRC is fine-tuned with all 125 categories in the Sketchy dataset. The logits layer score of the corresponding category is then utilized for sketch quality measurement (higher score → better quality). We then group sketches based on such scores and produce a Top-10 group that has the 10 best quality sketches, and a Bottom-10 group consisting of the 10 worst sketches. It can be seen from Table V and Fig. 10 that the high-quality sketches (Top-10) tend to perform better in segmentation model generation and produce more accurate segmentation maps than the Bottom-10 ones.

3) Applicability of Model: The objective of Sketch-a-Segmenter is to perform segmentation of *unseen* photo categories with the aid of any free-hand sketches. To further demonstrate this, we select four new categories (roller blades, suitcase, toothbrush, fire hydrant) that are not included in Sketchy and construct a new test set by sampling sketches from TU-Berlin [46] (80 per category). We then source and label category-paired photos from the Internet (50 per category). The results show that our category-level Hyper-DeepLabv3+Sketch (1), trained on Sketchy and PASCAL, could segment photos from this new test set with 68.98% and 88.74% mIOU and pixel accuracy respectively, compared with 64.52% and 84.57% when tested with the DeepLabV3+ model. Some qualitative segmentation results are shown in Fig. 11.

V. CONCLUSION

In this article, we introduce the novel task of sketch-based photo segmentation model generation, as well as the associated SketchySeg dataset. This paradigm provides a novel alternative approach to complement the few existing methods for dealing with the annotation bottleneck posed by the conventional supervised semantic segmentation problem setting. We show that sketch-based photo segmentation can generalize across categories, thus potentially enabling users to dynamically synthesize segmentation models for novel categories on the fly.

Preliminary results on part-based segmentation also show the potential for specifying desired subregions to extract on the fly. In future work, we intend to build a unified segmentation system for novel categories where the function of segmenters can be defined by user input, e.g., whole object → object-level segmentation, part sketch → part-level segmentation.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: https://arxiv.org/abs/1412.7062

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[5] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 801–818.

[6] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[7] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[8] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[9] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.

[10] S. Naha and Y. Wang, "Object figure-ground segmentation using zero-shot learning," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2842–2847.

[11] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation," in *Proc. CVPR*, Jun. 2019, pp. 8256–8265.

[12] N. Kato, T. Yamasaki, and K. Aizawa, "Zero-shot semantic segmentation via variational mapping," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–8.

[13] Q. Yu, F. Liu, Y.-Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy, "Sketch me that shoe," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 799–807.

[14] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Deep spatial-semantic attention for fine-grained sketch-based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5551–5560.

[15] K. Pang *et al.*, "Generalising fine-grained sketch-based image retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 677–686.

[16] S. Dey, P. Riba, A. Dutta, J. L. Llados, and Y.-Z. Song, "Doodle to search: Practical zero-shot sketch-based image retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2179–2188.

[17] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "A comprehensive survey to face hallucination," *Int. J. Comput. Vis.*, vol. 106, no. 1, pp. 9–30, Jan. 2014.

[18] M. Zhu, J. Li, N. Wang, and X. Gao, "A deep collaborative framework for face photo–sketch synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3096–3108, Oct. 2019.

[19] M. Zhu, N. Wang, X. Gao, J. Li, and Z. Li, "Face photo-sketch synthesis via knowledge transfer," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–7.

[20] C. Hu, D. Li, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-classifier: Sketch-based photo classifier generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9136–9144.

[21] P. Sangkloy, N. Burnell, C. Ham, and J. Hays, "The sketchy database: Learning to retrieve badly drawn bunnies," in *Proc. SIGGRAPH*, 2016, p. 119.

[22] L. Li, H. Fu, and C.-L. Tai, "Fast sketch segmentation and labeling with deep learning," *IEEE Comput. Graph. Appl.*, vol. 39, no. 2, pp. 38–51, Mar. 2019.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2169–2178.

[24] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[25] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[26] S. Kohl *et al.*, "A probabilistic u-net for segmentation of ambiguous images," in *Proc. NeurIPS*, 2018, pp. 6965–6975.

[27] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[28] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[29] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 876–885.

[30] Z. Shi, Y. Yang, T. M. Hospedales, and T. Xiang, "Weakly-supervised image annotation and segmentation with objects and attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2525–2538, Dec. 2017.

[31] A. Frome *et al.*, "Devise: A deep visual-semantic embedding model," in *Proc. NeurIPS*, 2013, pp. 2121–2129.

[32] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. NeurIPS*, 2013, pp. 935–943.

[33] Z. Jia, Z. Zhang, L. Wang, C. Shan, and T. Tan, "Deep unbiased embedding transfer for zero-shot learning," *IEEE Trans. Image Process.*, vol. 29, pp. 1958–1971, 2020.

[34] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 951–958.

[35] Y.-X. Wang and M. Hebert, "Learning to learn: Model regression networks for easy small sample learning," in *Proc. ECCV*, 2016, pp. 616–634.

[36] Y.-X. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Proc. NeurIPS*, 2017, pp. 7029–7039.

[37] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," 2016, *arXiv:1609.09106*. [Online]. Available: https://arxiv.org/abs/1609.09106

[38] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5822–5830.

[39] S. Qiao, C. Liu, W. Shen, and A. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7229–7238.

[40] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 991–998.

[41] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, May 2008.

[42] K. Li *et al.*, "Universal sketch perceptual grouping," in *Proc. ECCV*, 2018, pp. 582–597.

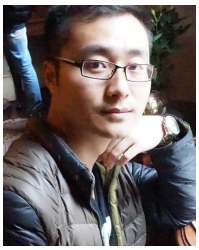
[43] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1971–1978.

[44] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

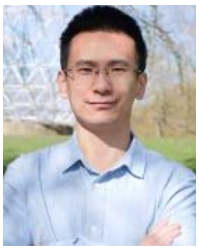
[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NeurIPS*, 2013, pp. 3111–3119.

[46] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, Aug. 2012.

Conghui Hu is currently pursuing the Ph.D. degree with Research Laboratory, Centre for Vision Speech and Signal Processing (CVSSP), University of Surrey. Her research interest includes computer vision, particularly focused on free-hand sketch and its applications.



Da Li received the Ph.D. degree from SketchX Research Laboratory, U.K., under the Supervision of Dr. Yi-Zhe Song and Dr. Timothy M. Hospedales. He is currently a Researcher within Machine Learning Group, Samsung AI Centre, Cambridge, U.K. His research interests include transfer learning, meta-learning and semi-supervised learning, spanning applications in computer vision, and speech recognition with deep learning approaches.



Yongxin Yang received the Ph.D. degree from the Queen Mary University of London. He is currently a Lecturer with the University of Surrey. His research interests include multi-task learning, transfer learning, and meta-learning. He has broad interests in applications of machine learning, e.g., computer vision, medical informatics, and finance.

Timothy M. Hospedales (Member, IEEE) is currently a Professor with the School of Informatics, The University of Edinburgh. He is also a Principal Researcher with Samsung AI Centre Cambridge, where he leads the Machine Learning Group. His research interests include machine learning and computer vision. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI). He serves as an Area Chair of several major events including ICCV, CVPR, ECCV, AAAI, and ACL and a Program Chair of BMVC 2018.

Yi-Zhe Song (Senior Member, IEEE) received the degree (Hons.) from the University of Bath in 2003, the M.Sc. degree (Hons.) from the University of Cambridge in 2004, and the Ph.D. degree in computer vision and machine learning from the University of Bath, in 2008. He is currently a Reader of computer vision and machine learning with the Centre for Vision Speech and Signal Processing (CVSSP), U.K.'s largest academic research center for Artificial Intelligence with approximately 200 researchers. Previously, he was a Senior Lecturer with the Queen Mary University of London and a Research and Teaching Fellow with the University of Bath. He is a fellow of the Higher Education Academy. He is a Full Member of the Review College of the Engineering and Physical Sciences Research Council (EPSRC), the U.K.'s main agency for funding research in engineering and the physical sciences, and serves as an Expert Reviewer for the Czech National Science Foundation.