Dataset Augmentation with Synthetic Images Improves Semantic Segmentation

P. S. Rajpura
IIT Gandhinagar
param.rajpura@iitgn.ac.in

M. Goyal
IIT Varanasi
manik.goyal.cse15@iitbhu.ac.in

R.S. Hegde
IIT Gandhinagar
hegder@iitgn.ac.in

H. Bojinov
Innit Inc.
hristo.bojinov@innit.com

***Abstract*—Although Deep Convolutional Neural Networks trained with strong pixel-level annotations have significantly pushed the performance in semantic segmentation, annotation efforts required for the creation of training data remains a roadblock for further improvements. We show that augmentation of the weakly annotated training dataset with synthetic images minimizes both the annotation efforts and also the cost of capturing images with sufficient variety. Evaluation on the PASCAL 2012 validation dataset shows an increase in mean IOU from 52.80% to 55.47% by adding just 100 synthetic images per object class. Our approach is thus a promising solution to the problems of annotation and dataset collection.**

***Index Terms*—Convolutional Neural Networks (CNN); Deep learning; Semantic Segmentation; Synthetic datasets; 3D Rendering**

I. Introduction

Deep Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance on several image processing and computer vision tasks like image classification, object detection, and segmentation [1], [2]. Numerous applications depend on the ability to infer knowledge about the environment through image acquisition and processing. Hence, scene understanding as a core computer vision problem has received a lot of attention. Semantic segmentation, the task of labelling pixels by their semantics (like 'person', 'dog', 'horse'), paves the road for complete scene understanding. Current state-of-the-art methods for semantic segmentation are dominated by deep convolutional neural networks (DCNNs) [3]–[5]. However, training end-to-end CNNs requires large scale annotated datasets. Even with a large enough dataset, training segmentation models with only image level annotations is quite challenging [6]–[8] as the architecture needs to learn from higher level image labels and then predict low-level pixel labels. The significant problem here is the need for pixel-wise annotated labels for training, which becomes a time-consuming and expensive annotation effort. Even standard datasets like PASCAL VOC [9] provide only 1464 (training) and 1449 (validation) pixel-wise labelled images. Some researchers have extended training dataset [10] (consisting of the same 21 classes as PASCAL VOC) to counter this problem. In practical applications, the challenge still stands since various classes of objects need to be detected, and annotated dataset for such training is always required.

Most of the reports using deep CNN models for semantic segmentation use weakly-annotated datasets. Typically, such weak annotations take the form of bound-boxes, because forming bound-boxes around every instance of a class is around 15 times faster than doing pixel-level labelling [11]. These approaches rely on either defining some constraints [12] or on using multiple instance learning [13] techniques. [14] uses GraphCut for approximating bound-boxes to semantic labels. Although deep CNNs (such as the one proposed by [15] using DeepLab model [5]) significantly improved the segmentation performance using such weakly-annotated datasets, they failed to provide good visualization on test images.

There have been solutions proposed to reduce annotation efforts by employing transfer learning or simulating scenes. The research community has proposed multiple approaches for the problem of adapting vision-based models trained in one domain to a different domain [16]–[20]. Examples include: re-training a model in the target domain [21]; adapting the weights of a pre-trained model [22]; using pre-trained weights for feature extraction [23]; and, learning common features between domains [24]. Augmentation of datasets with synthetically rendered images or using datasets composed entirely of synthetic images is one of the techniques that is being explored to address the dearth of annotated data for training all kinds of CNNs. Significant research in transfer learning from synthetically rendered images to real images has been published [25], [26]. Most researchers have used gaming or physics rendering engine to produce synthetic images [25] especially in the automotive domain. Peng et al. [27] have done progressive work in the Object Detection context, understanding various cues affecting transfer learning from synthetic images. But they train individual classifiers for each class after extracting features from pre-trained CNN. They show that adding different cues like background, object textures, shape to the synthetic image increases the performance [26], [28] for object detection. There has not been an attempt yet to benchmark performances on the standard PASCAL VOC [9] semantic segmentation benchmark using synthetic images.

To the best of our knowledge, our report is the first attempt at combining weak annotations (generating semantic labels from bound box labels) and synthetically rendered images from freely available 3D models for semantic segmentation. We demonstrate a significant increase in segmentation performance (as measured by the mean of pixel-wise intersection-over-union (IoU)) by using semantic labels from weak annotations and synthetic images. We used a Fully Convolutional

Network (FCN-8s) architecture [3] and evaluate it on the standard PASCAL VOC [9] semantic segmentation dataset. The rest of this paper is organized as follows: our methodology is described in section II, followed by the results we obtain reported in section III, finally concluding the paper in section IV.

II. Method

Given an RGB image capturing one or many of the 20 objects included in PASCAL VOC 2012 semantic segmentation challenge, our goal is to predict a label image with pixel-wise segmentation for each object of interest. Our approach, represented in Figure 1, is to train a deep CNN with synthetic rendered images from available 3D models. We divide the training of FCN into two stages: fine-tuning FCN with the `Weak(10k)` dataset (real images with bound box labels) generated from bound box labelled images, and fine-tuning with our own `Syn(2k)` dataset (synthetic images rendered from 3D models). Our methodology can be divided into two major parts: dataset generation and fine-tuning of the FCN. These are explained in the following subsections.

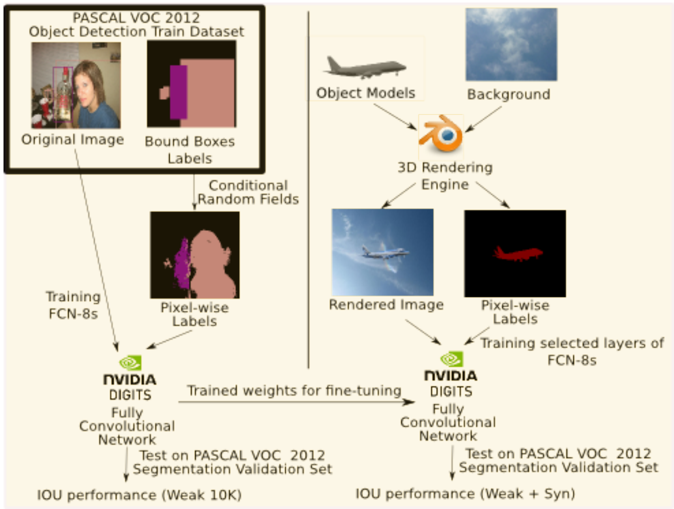
PASCAL VOC 2012
Object Detection Train Dataset
Original Image
Bound Boxes Labels
Conditional Random Fields
Pixel-wise Labels
Training FCN-8s
NVIDIA DIGITS
Fully Convolutional Network
Trained weights for fine-tuning
Test on PASCAL VOC 2012 Segmentation Validation Set
IOU performance (Weak 10K)
Object Models
Background
3D Rendering Engine
Rendered Image
Pixel-wise Labels
Training selected layers of FCN-8s
NVIDIA DIGITS
Fully Convolutional Network
Test on PASCAL VOC 2012 Segmentation Validation Set
IOU performance (Weak + Syn)

Figure 1. Overview representing the approach for learning semantic segmentation from weak bound-box labels and synthetic images rendered using 3D models.

A. Dataset generation

Weakly supervised semantic annotations: To train the CNN for semantic segmentation, we use the available bound-box annotations in PASCAL-VOC object detection challenge training set (10k images with 20 classes). Since the bound-boxes fully surround the object including pixels from background, we filter those pixels into foreground and background. Later the foreground pixels are given their corresponding object label in cases where multiple objects are present in an image. Two methods were chosen for converting bound-boxes to semantic segmentation namely Grab-Cut [14] and Conditional Random Fields (CRF) as deployed by [5], [15]. Based upon the performance on a few selected images, we use the labels from CRF for training the CNN. Figure 2 shows the comparison of results from both methods. Grab-Cut tends to miss labelling smaller objects but is precise in labelling larger objects. CRF labels objects of interest accurately with a small amount of noise around the edges.

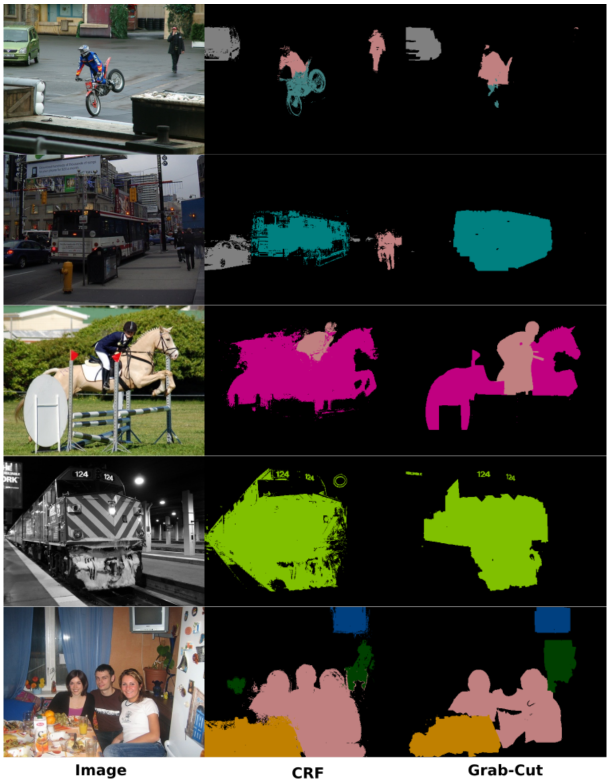

Figure 2. Comparision of CRF and Grab-Cut segmentation from bound box labels

Synthetic images rendered from 3D models: We use the open source 3D graphics software Blender for this purpose. Blender-Python APIs facilitates the loading of 3D models and automation of scene rendering. We use Cycles Render Engine available with Blender since it supports ray-tracing to render synthetic images. Since all the required information for annotation is available, we use the PASCAL Segmentation label format with labelled pixels for 20 classes.

Real world images have lot of information embedded about the environment, illumination, surface materials, shapes etc. Since the trained model, at test time must be able to generalize to the real world images, we take into consideration the following aspects during generation of each scenario:

- Number of objects
- Shape, Texture, and Materials of the objects
- Background of the object
- Position, Orientation of camera
- Illumination via light sources

In order to simulate the scenario, we need 3D models, their texture information and metadata. Thousands of 3D CAD models are available online. We choose ShapeNet [29] database since it provides a large variety of models in the 20 categories for PASCAL segmentation challenge. Figure 3a shows few of the models used for rendering images. The variety helps randomize the aspect of shape, texture and materials of the objects. We use images from SUN database [30] as background images. From the large categories of images, we select few categories relevant as background to the classes of objects to be recognized.

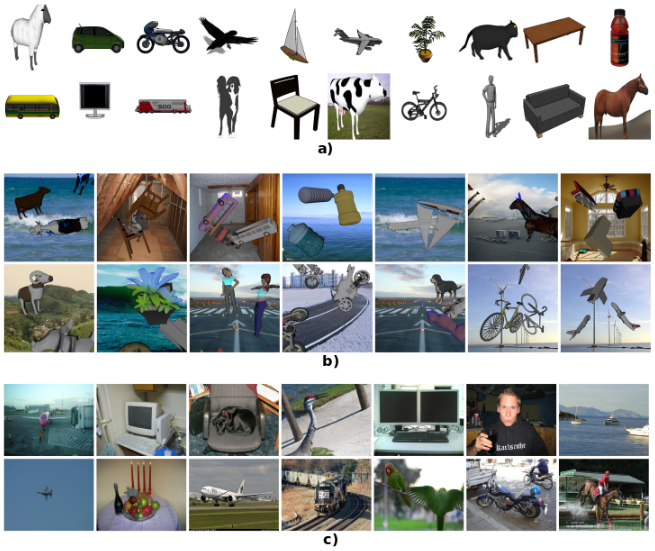

Figure 3. (a) 3D models used for synthetic dataset (b) Synthetic images with multiple objects (c) Images from the training set of PASCAL VOC 2012 Object Detection Dataset

For generating training set with rendered images, the 3D scenes need to be distinct. For every class of object, multiple models are randomly chosen from the model repository. Each object is scaled, rotated and placed at random location within the field of view of the camera which is placed at a predefined location. The scene is illuminated via directional light source. Later, a background image is chosen from the database and the image is rendered with Cycles Render Engine, finally generating RGB image and pixel-wise labelled image. Figure 3b shows few rendered images used as training set while Figure 3c shows the subset of real images from PASCAL Object Detection dataset (`Weak(10k)`) used in training.

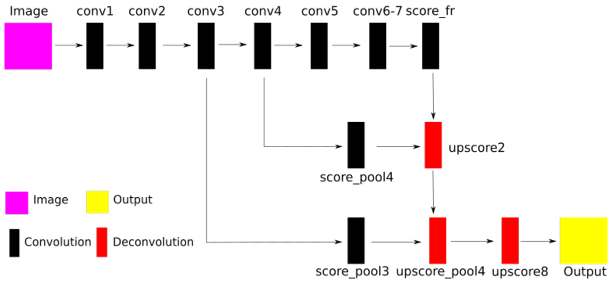

Figure 4. FCN-8s Architecture [3]

B. Fine-tuning the Deep CNN

We fine-tune FCN-8s [3] pretrained on ImageNet [31] initially with 10k real images along with semantic labels generated from bound-boxes using CRF. All layers in the network are fined-tuned with base learning rate of $1e^{-5}$. We further reference this model as baseline model. In next stage, we fine tune the baseline model with synthetic images generated from Blender. Selected layers (score_pool3, score_pool4, upscore2, upscore_pool4 and upscore8 shown in Figure 4) consisting of 2 convolutional and 3 deconvolutional layers are fine-tuned with base learning rate of $1e^{-6}$. The network is trained with Adam optimizer for pixel-wise softmax loss function. Since the rendered images from 3D models are not rich in terms cues like textures, shadows and hence are not photo-realistic, we choose to fine-tune only few layers to capture majorly the higher hierarchical features like shape of the object.

III. Results and Discussion

The experiments were carried out using the workstation with Intel Core i7-5960X processor accelerated by NVIDIA GEFORCE GTX 1070. NVIDIA-DIGITS^{TM} (v5.0) [32] was used with Caffe library to train and manage the datasets. The proposed CNN was evaluated on the PASCAL-VOC 2012 segmentation dataset consisting of 21 classes (20 foreground and 1 background class). The PASCAL 2012 segmentation dataset consists of 1464 (train) and 1449 (val) images for training and validation respectively.

Table I shows the comparison of various CNN models trained on datasets listed in the first column. The performances reported are calculated according to the standard metric, mean of pixel-wise intersection-over-union (IoU). The first row lists the 21 classes and the mean IOU over all 21 classes.

The first row displays the performance when FCN is fine-tuned for real images with strong pixel-wise annotations from PASCAL VOC 2012 segmentation training dataset addressed as `Real(1.5k)`. Using bound-box annotated images as weak annotations was an alternative proposed earlier by [15] which performed better than just using standard training dataset (with size of approximately 1.5k images). The second row showcases that the model trained with 10k weak bound-box annotated data (converted to pixel-wise labels using CRF) improved the mean-IoU performance from 47.68% to 52.80%. The predictions from CNN trained on `Weak(10k)` are represented in Figure 5 third column. On comparing them with ground truth, it is observed that predictions miss the shape and sharp boundaries of the object.

Considering the CNN fine-tuned for `Weak(10k)` dataset as the baseline, we further fine-tuned it with rendered images of single class. Table I highlights the effect of using synthetic dataset on few classes namely car, bottle and aeroplane. `Syn_Car(100)` denotes the dataset of 100 synthetic images with car as the object of interest. We observed that by using few synthetic images from single class, the performance of segmenting car as well as 7 other classes improved. The improvement in other classes can be explained by common features learned from the car images. This trend can be

Table I. IoU Values for different classes evaluated on the PASCAL Semantic Segmentation Validation Set [9]. IoU Values higher than baseline model have been highlighted

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dinning Table	Dog	Horse	Motor Bike	Person	Potted Plant	Sheep	Sofa	Train	Tv Monitor	Mean IoU
Real(1.5k)	**87.88**	58.07	**46.47**	54.61	39.73	41.56	61.67	48.47	64.93	**16.92**	30.43	39.47	53.43	44.40	59.03	65.47	14.06	56.05	24.75	54.05	**38.62**	47.62
Weak(10k)	87.12	68.81	27.39	55.46	42.76	47.25	73.96	51.15	71.37	13.17	57.88	42.95	**60.74**	54.92	54.99	**66.01**	41.38	65.71	30.37	67.48	**27.99**	52.80
Weak(10k)+Syn_Car(100)	**87.38**	**69.75**	**27.48**	55.35	**46.25**	44.38	**77.70**	**69.31**	69.32	11.35	57.28	42.93	59.65	54.34	**56.47**	65.70	36.12	**66.08**	29.46	66.47	**28.26**	**53.38**
Weak(10k)+Syn_Bot(100)	86.78	67.98	24.29	**57.65**	**45.62**	**57.55**	73.15	50.42	68.80	10.38	**58.98**	**42.98**	58.71	54.70	50.58	63.92	**41.46**	**66.27**	28.46	64.04	27.97	52.41
Weak(10k)+Syn_Aero(100)	85.39	67.89	23.13	**63.90**	**48.89**	27.83	68.51	47.04	65.39	5.29	56.24	27.21	54.36	51.41	44.46	61.50	23.72	61.90	17.60	56.34	16.73	46.42
Weak(10k)+Syn(2k)	**87.88**	**69.39**	**27.65**	**57.23**	**47.76**	**55.52**	**76.19**	**61.83**	**72.58**	**14.91**	**59.23**	**48.50**	60.56	**56.64**	**62.49**	63.43	**49.54**	**65.87**	**36.95**	**68.69**	22.06	**55.47**

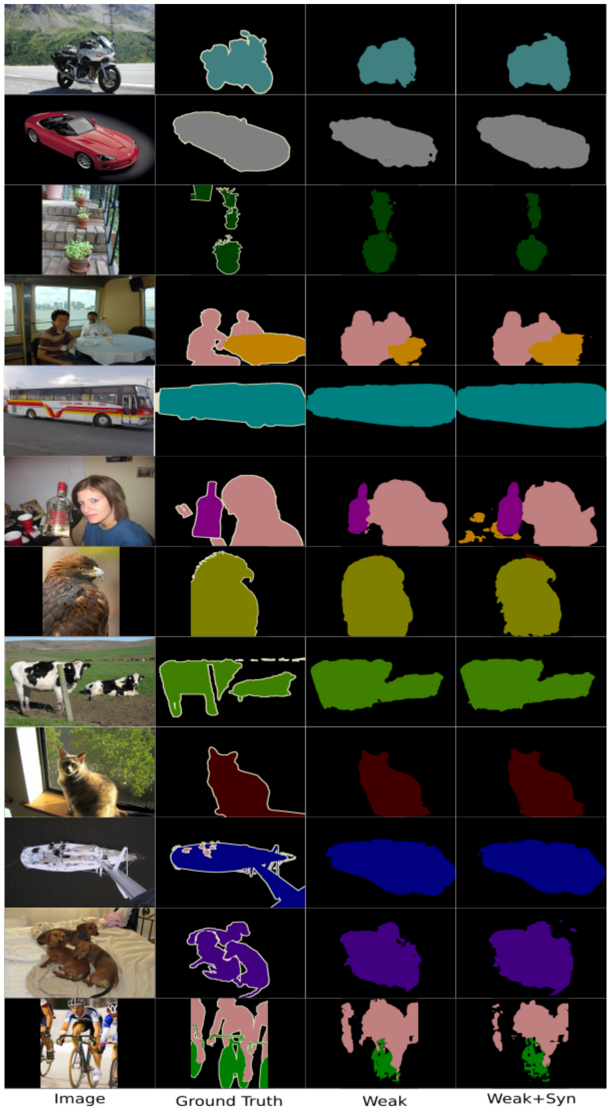

Figure 5. Comparision of the segmented labels from Ground Truth, predictions from CNN trained with `Weak(10k)` and `Weak(10k)+Syn(2k)`

observed for other classes like bottle (`Syn_Bot(100)`) and aeroplane (`Syn_Aero(100)`).

Finally, we fine-tuned the baseline model with complete set of synthetic images (100 images per class; 20 classes) addressed as `Syn(2k)`. The mean IoU of this model increased from 52.80% to 55.47% as shown in Table I which clearly proves our hypothesis of supplementing synthetic images with weak annotated dataset. Some classes (car, bottle) showcased a significant improvement (10% for car, 8% for bottle) indicating the synthetic images in such cases to be more informative than others. While the classes like bicycle, dog, person and TV-monitor had lower IoU values since we had fewer 3D models available for those object types. Since objects like cow, cat, person etc. have highly variable appearance compared to other object classes, we observe lesser improvement in performance.

Figure 5 shows the comparison of the semantic labels generated from network trained on `Weak(10k)` dataset and `Weak(10k)+Syn(2k)` dataset. The latter predictions are better since they produce sharper edges and shapes. The results prove that shape information from the synthetic models help eliminate the noise generated from CRF in labels. It is worth noting that even though synthetic images are non photo-realistic, and lack visual information from relevant backgrounds for objects, multiple object class in a single image or rich textures but they represent higher hierarchical feature like shape and thus can be used alongside weakly annotated images to achieve better performance on semantic segmentation tasks.

IV. Conclusion

Our report demonstrates a promising approach to minimize the annotation and dataset collection efforts by using rendered images from freely available 3D models. The comparison shows that using 10k weakly annotated images (which approximately equals the annotation efforts for 1.5k strong labels) with just 2k synthetic rendered images gives a significant rise in segmentation performance.

This work can be extended by training CNN with larger synthetic dataset, with richer 3D models. Adding other features like relative scaling and occlusions can further strengthen the synthetic dataset. The effect of using synthetic dataset with improved architectures for semantic segmentation are being explored further.

Acknowledgment

We acknowledge funding support from Innit Inc. consultancy grant CNS/INNIT/EE/P0210/1617/0007.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” 2012.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper with Convolutions,” 2014.

[3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *Cvpr 2015*, p. 5, 2015.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs,” *Iclr*, pp. 1–14, 2014.

[6] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, “Weakly supervised structured output learning for semantic segmentation,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 845–852.

[7] J. Verbeek and B. Triggs, “Region classification with markov field aspect models,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[8] J. Xu, A. G. Schwing, and R. Urtasun, “Tell me what you see and i will show you where it is,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3190–3197.

[9] 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.

[10] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors,” in *2011 International Conference on Computer Vision*, Nov 2011, pp. 991–998.

[11] 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*. Cham: Springer International Publishing, 2014, pp. 740–755.

[12] D. Pathak, P. Krahenbuhl, and T. Darrell, “Constrained convolutional neural networks for weakly supervised segmentation,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 11-18-December-2015, pp. 1796–1804, 2016.

[13] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, “F ULLY C ONVOLUTIONAL M ULTI -C LASS M ULTIPLE I NSTANCE L EARNING,” no. 1, pp. 1–4, 2015.

[14] C. Rother, V. Kolmogorov, and A. Blake, “"GrabCut": interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, p. 309, 2004.

[15] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, “Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation,” *arXiv preprint*, p. 10, 2015.

[16] L. Duan and I. W. Tsang, “Learning with Augmented Features for Heterogeneous Domain,” 2012.

[17] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, “Efficient Learning of Domain-invariant Image Representations,” 2013.

[18] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, “LSDA: Large Scale Detection Through Adaptation,” 2014.

[19] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” 2011.

[20] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning Transferable Features with Deep Adaptation Networks,” 2015.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” 2014.

[22] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, “Revisiting Batch Normalization For Practical Domain Adaptation,” 2016.

[23] A. Gupta, A. Vedaldi, and A. Zisserman, “Synthetic Data for Text Localisation in Natural Images,” 2016.

[24] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep Domain Confusion: Maximizing for Domain Invariance,” 2014.

[25] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez, “The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *(CVPR)*, 2016.

[26] X. Peng and K. Saenko, “Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks,” p. 13, 2017.

[27] X. Peng, B. Sun, K. Ali, and K. Saenko, “Learning Deep Object Detectors from 3D Models,” 2014.

[28] X. Peng and K. Saenko, “Combining Texture and Shape Cues for Object Recognition With Minimal Supervision,” 2016.

[29] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” dec 2015.

[30] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3485–3492.

[31] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2009, pp. 248–255.

[32] “Image Segmentation Using DIGITS 5.” [Online]. Available: https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5/