Transfer Learning by Finetuning Pretrained CNNs Entirely with Synthetic Images

Param Rajpura[1], Alakh Aggarwal[2], Manik Goyal[2], Sanchit Gupta[3], Jonti Talukdar[4], Hristo Bojinov[5] and Ravi Hegde[1]

[1] Indian Institute of Technology Gandhinagar, India 382355
[2] Indian Institute of Technology (BHU) Varanasi, India 221005
[3] BITS Hyderabad, India 500078
[4] Nirma Institute of Technology, India 382 481
[5] Innit Inc., USA 94063

Abstract. We show that finetuning pretrained CNNs entirely on synthetic images is an effective strategy to achieve transfer learning. We apply this strategy for detecting packaged food products clustered in refrigerator scenes. A CNN pretrained on the COCO dataset and fine-tuned with our 4000 synthetic images achieves mean average precision (mAP @ 0.5-IOU) of 52.59 on a test set of real images (150 distinct products as objects of interest and 25 distractor objects) in comparison to a value of 24.15 achieved without such finetuning. The synthetic images were rendered with freely available 3D models with variations in parameters like color, texture and viewpoint without a high emphasis on photorealism. We analyze factors like training data set size, cue variances, 3D model dictionary size and network architecture for their influence on the transfer learning performance. Additionally, training strategies like fine-tuning with selected layers and early stopping which affect transfer learning from synthetic scenes to real scenes were explored. This approach is promising in scenarios where limited training data is available.

1 Introduction

Deep Convolutional Neural Networks (CNNs) have fulfilled the demand for a robust feature extractor and have achieved state-of-the-art performance on image classification, object detection, and segmentation [1, 2] tasks. The availability of large sets of training images has been a prerequisite for successfully training CNNs [1]. Manual annotation of images for object detection is a time-consuming and mechanical task; what is more, in some applications the cost of capturing images with sufficient variety is prohibitive. In fact the largest image datasets are built upon only a few categories for which images can be feasibly curated (20 categories in PASCAL VOC [3], 80 in COCO [4], and 200 in ImageNet [5]).

There have been solutions proposed to reduce annotation efforts by employing transfer learning or simulating scenes to generate large image sets. The research community has proposed multiple approaches for the problem of adapting vision-based models trained in one domain to a different domain [6–10]. Examples include: re-training a model in the target domain [11]; adapting the weights of a

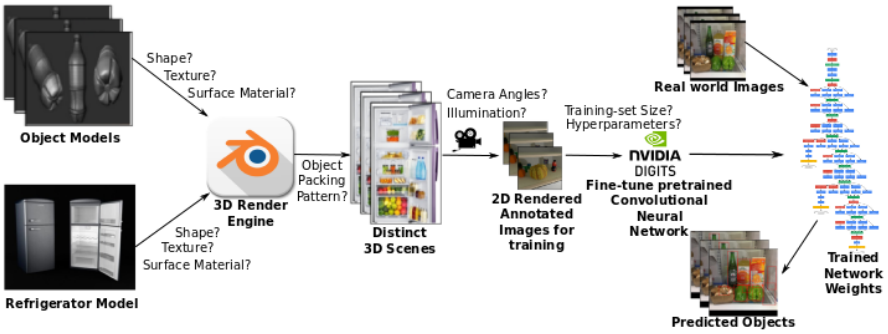

Fig. 1. Overview of our approach to train object detectors for real images based on rendered images.

pre-trained model with adaptive batch normalization [12]; overlaying synthetic text on real background images [13]; and, learning common features by maximizing domain confusion between source and target domains [14]. Donahue et al. [15] introduced the concept of learning linear classifiers for target dataset over DeCAF (features) extracted from CNN trained on large public datasets. Though Tommasi et al. [16] report that DeCAF not only does not solve the dataset bias problem in general, but in some cases (both class- and dataset-dependent) they capture specific information that induce worse performance than what can be obtained with less powerful features. Oquab et al. [17] used mid-level representation and finetuned them for target datasets which has been considered standard practice given annotated dataset is available for target domain.

Attempts to use synthetic data for training CNNs to adapt in real scenarios have been made in the past. Peng et. al. used available 3D CAD models to render images (after varying the projections and orientations of the objects) for evaluation on 20 categories in the PASCAL VOC 2007 data set [18]. They aimed to study the effects of texture and background by training the classifier. Su and coworkers [19] used the rendered 2D images from 3D on varying backgrounds for pose estimation. Their work also uses an object proposal stage and limits the objects of interest to a few specific categories from the PASCAL VOC data set. Georgakis and coworkers [20] propose to learn object detection with synthetic data generated by object instances being superimposed into real scenes at different positions, scales, and illumination. They propose the use of existing object recognition data sets such as BigBird [21] rather than using 3D CAD models. They limit their synthesized scenes to low-occlusion scenarios with 11 products in GMU-Kitchens data set. Gupta et. al. generate a synthetic training set by taking advantage of scene segmentation to create synthetic training examples, however the goal is text localization instead of object detection [13]. Tobin et. al. perform domain randomization with low-fidelity rendered images from 3D meshes, however their objective is to locate simpler polygon-shaped objects re-

stricted to a table top in world coordinates [22]. In [23, 24], the Unity game engine is used to generate RGB-D rendered images and semantic labels for outdoor and indoor scenes. They show that by using photo-realistic rendered images the effort for annotation can be significantly reduced. They combine synthetic and real data to train models for semantic segmentation, however the network requires depth map for semantic segmentation.

In this paper, we report performance of transfer learning by finetuning a pretrained network (with standard public datasets) to identify objects in refrigerator with reasonably small sized dataset of synthetic images rendered from available 3D models. We automate the process of rendering 3D models with variations in viewpoints, cues etc. and annotating the 2D images to use it for object detection in real scenes. We report the study of cues or hyper-parameters involved to efficiently achieve transfer learning from synthetic to real images. Our experiments explore the effects of data set size, 3D model repository sizes. While lesser attention has been given to understanding selective layer fine-tuning for transfer learning, we explore it and report significant improvements. Furthermore, training strategy like early stopping [25] is also used for transfer learning from simulation to reality. Hence, this approach is particularly relevant in the detection of object candidates in scenes with large intra-class variance as opposed to one with only a few specific categories using synthetic datasets which do not require extensive effort towards achieving photorealism. The rest of this paper is organized as follows: our methodology is described in section 2, followed by the results we obtain reported in section 3, finally concluding the paper in section 4.

2 Method

To study transfer learning from synthetic to real images we choose the object detection task where given an RGB image captured inside a refrigerator, our goal is to predict a bound-box for each object of interest. In addition, there are few objects in the scene that need to be neglected. Our approach is to train a deep CNN with synthetic rendered images from available 3D models. Overview of the approach is shown in Figure 1. Our work can be divided into two major parts namely synthetic image rendering from 3D models and transfer learning by fine-tuning the deep neural network with synthetic images.

2.1 Synthetic Generation of Images from 3D Models

We use Blender and its Python APIs to load 3D models and automate the scene rendering. We use Cycles Render Engine with Blender since it supports ray-tracing to render synthetic images. Since all the required annotation data is available, we use the KITTI [26] format with bound-box co-ordinates, truncation state and occlusion state for each object in the image.

Considering the information embedded about the environment, illumination, surface materials, shapes etc. in real world scenes, we include following aspects

like a) Number of objects, b) Shape, Texture, and Materials of the objects, c) Texture and Materials of the refrigerator, d) Packing pattern of the objects, e) Position, Orientation of camera and f) Illumination via light sources while rendering each scene.

To simulate the real world scenario, we need 3D models, their texture information and metadata. Thousands of 3D CAD models are available online. We choose ShapeNet [27] database and Archive3D [28]. Among various categories from ShapeNet like bottles, tins, cans and food items, we selectively add 616 various object models including objects of interest and distractor objects to the repository (R_0) for generating scenes. The variety helps randomize the aspect of shape, texture and materials of the objects. For the refrigerator, we choose a model from Archive3D [28] suitable for the application. The design of refrigerator remains same for all the scenarios though the textures and material properties are dynamically chosen.

For generating training set, the refrigerator model with 5-25 randomly selected objects from R_0 are imported in each scene. To simulate object packing in refrigerator, we use three patterns namely grid, random and bin packing for 3D models. The grid places the objects at predefined distances on the refrigerator tray. Random placements drop the objects at random locations while bin packing tries to optimize the usage of tray top area placing objects very close and clustered in the scene to replicate real world scenarios in refrigerator. To replicate the packing in 3D, we also stack few objects vertically. The light sources are placed such that illumination is varied in every scene and the images are not biased to a well lit environment since refrigerators generally tend to have dim lighting. Multiple cameras are placed at random location and orientation to render images from each scene. The refrigerator texture and material properties are dynamically chosen for every rendered image.

2.2 Transfer Learning by Fine-tuning CNN and Evaluation

For neural network training we use NVIDIA-DIGITS^{TM}-DetectNet [29] along with GoogleNet weights pretrained on ImageNet using Caffe [30] library in backend. For a comparison among various architectures we also use Faster-RCNN [31] (with ResNet-101 [32] as feature mapping network) and SSD [33] using Tensorflow and weights pretrained on COCO [4] dataset. During training, the labelled RGB images with resolution (in pixels) 512 x 512 are used. We neglect objects truncated or highly occluded in the images in the ground truth label.

We evaluate our object detector using manually annotated crowd-sourced refrigerator images. Figure 3 illustrates the variety in object textures, shapes, scene illumination and environment cues present in the test set. The real scenarios also include other objects like vegetables, fruits, etc. which need to be neglected by the detector. We address them as distractor objects. For performance evaluation, we compute Intersection over Union (IoU) score. With a threshold parameter, predicted bound boxes are classified as True Positives (TP), False Positives (FP) and False Negatives (FN). Precision (PR) and Recall (RE) are calculated using

these metrics and a simplified mAP score is defined by the product of PR and RE [34].

All the experiments were carried on workstation with Intel^R Core^{TM} i7-5960X processor accelerated by NVIDIA^R GEFORCE^{TM} GTX 1070. Hyper-parameters search on learning rate, learning rate policy, training epochs, batch-size were performed for training all neural network models.

3 Results and Discussion

The purpose of our experiments was to evaluate the efficacy of transfer learning from rendered 3D models on real refrigerator scenarios. Hence we divide this section into two parts:

- Factors affecting Transfer Learning: We study following factors using the DetectNet architecture:
  a) Training Dataset Size
  b) Selected Layer Fine-tuning: Features learned at each layer in CNNs have been distinct and found to be general across domains and modalities. Fine-tuning of the final fully-connected linear classification layers has been used in practice for transfer learning across applications. Hence, we fine-tune selected layers of the network and report the performance.
  c) Object Dictionary Size: Variance in objects used for rendering has been observed to increase detection performance significantly [19]. Hence, we report performance for various object repository size used while finetuning.
- Detection Accuracy: Here, we represent the analysis of the performance on real dataset achieved with three different architectures and three training datasets with varying cues and complexity[6].

3.1 Factors affecting transfer learning

Considering other parameters like object dictionary size and fine-tuned network layers, we vary the training data size from 500-6000. We observe an increase in mAP up to 4000 images followed by a slight decline in performance as shown in Figure 2 a). Note that the smaller dataset is a subset of the larger dataset size. After an extent, we observe decline in accuracy as we increase the dataset size.

We use GoogleNet FCN architecture with 13 hierarchical levels including inception modules as single level. mAP vs. number of epochs chart is presented in Figure 2 b) for models with different layers selected for fine-tuning. Starting from training just the final coverage and bounding-box regressor layers we sequentially open deeper layers for fine-tuning. We observe that fine-tuning all the inception modules helps transfer learning from synthetic images to real images in our application. The results show that selection of the layers to fine-tune proves to be important for detection performance.

[6] Trained network weights and synthetic dataset are available at `https://github.com/paramrajpura/Syn2Real`

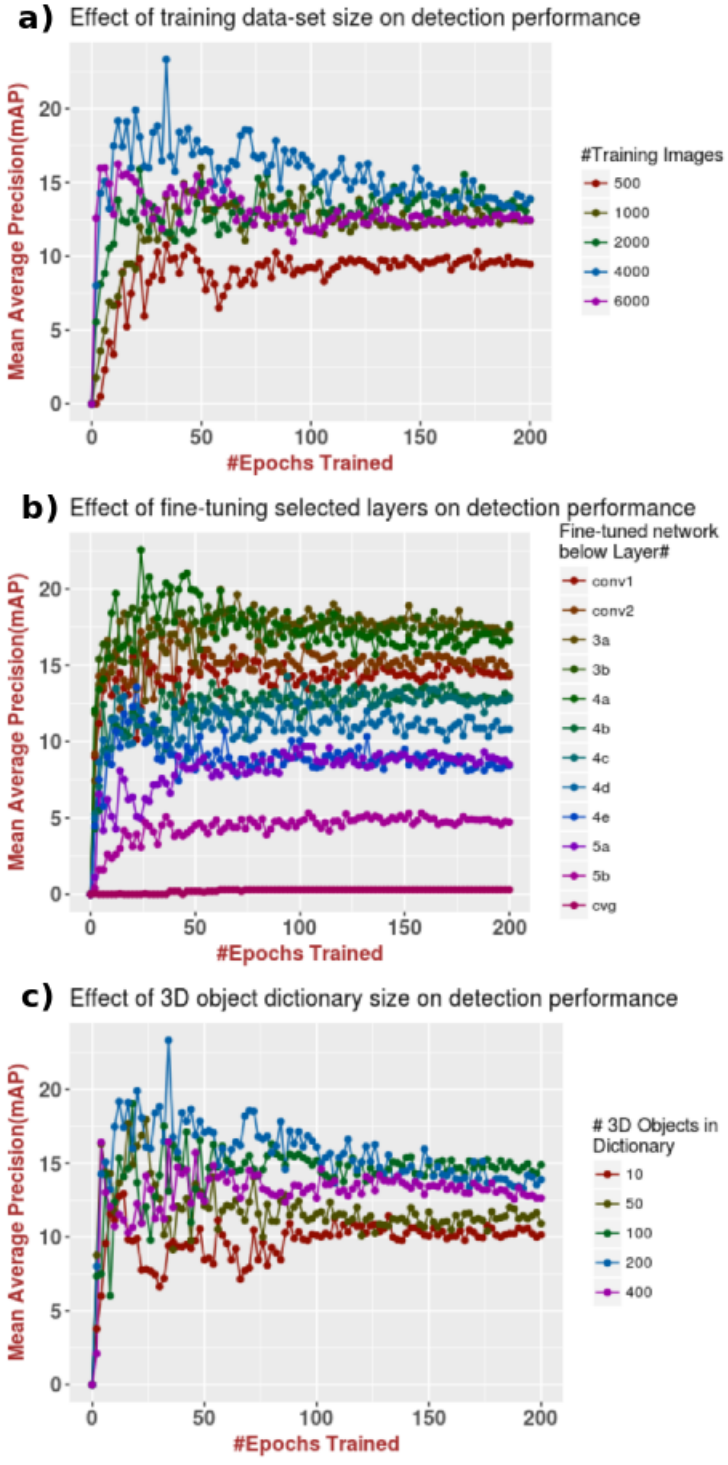

Fig. 2. a) Detection results on the validation image-set varying train dataset size. b) Neurons in layers of CNN learn distinct features. Network weights were fine-tuned by freezing layers sequentially. The figure represents the performance with weights fine-tuned till mentioned layers. c) Variety in training data affects the capability of generalizing object detection in real scenarios.

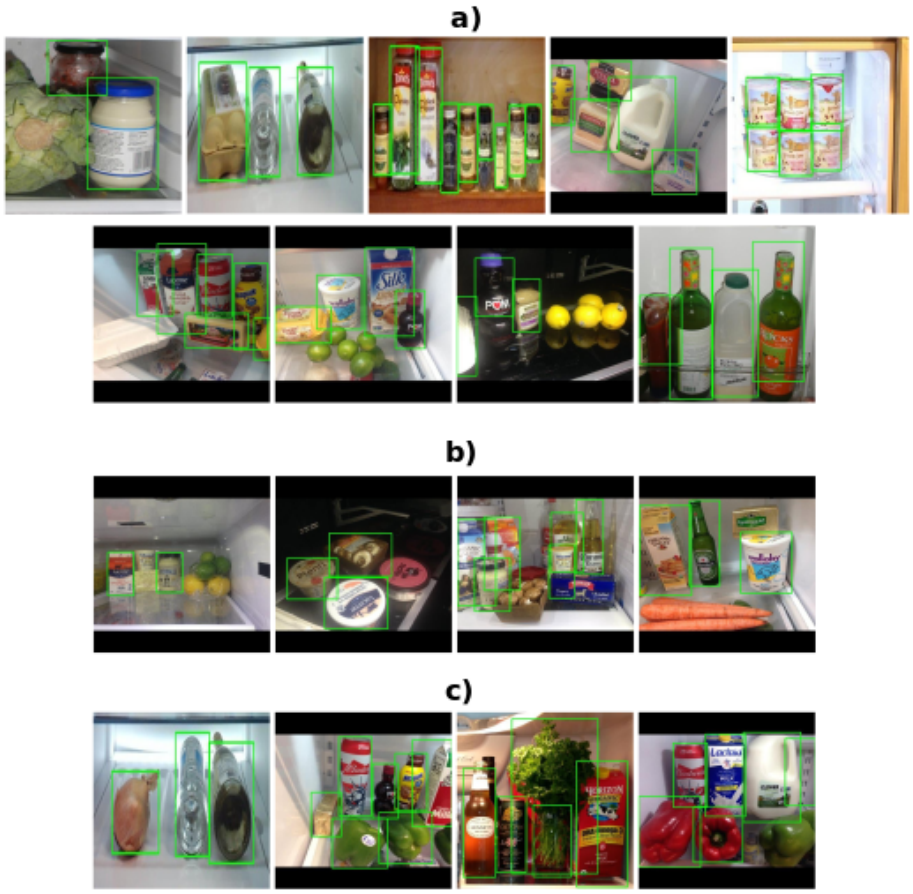

Fig. 3. Scenes representing variance in scale, background, textures, illumination, packing patterns and material properties wherein a) Object detector correctly predicts the bound boxes for all objects of interest. b) Object detector misses objects of interest. c) Object detector falsely predicts the presence of an object.

To study the relationship of variance in 3D models with performance, we incrementally add distinct 3D models to the dictionary starting from 10 to 400. We observe an increase in mAP up to 200 models and slight decline later on as represented in Figure 2 c).

We report in Figure 2 mAP vs. epochs trained plots over mAP vs. variance in factor to also represent the relevance of early stopping [25]. The networks trained by varying factors, show their peak performances for 25-50 epochs of training while the performance declines contrary to saturating which suggests over-fitting to synthetic images.

Fig. 4. Figure illustrates the improvement in object detection by finetuning CNN with synthetic rendered dataset. First row highlights the prediction from baseline model trained on COCO dataset with white bound-boxes. Second row marks predictions (with green bound-boxes) from fine-tuned CNN with synthetic dataset.

3.2 Detection Accuracy

We build 3 training datasets with increasing variance and cues. Dataset A consists only the objects of interest (positives) packed inside the refrigerator with grid, random and bin packing. The refrigerator wall is assigned a random solid color for every scene. Dataset B consists of a subset of Dataset A and images with distractor objects and refrigerator wall with glossy or glass material properties. While, Dataset C with few images from Dataset A and B also contains scenes where objects were vertically stacked on each other bringing the scenario more closer to the real world in terms of object arrangement. We evaluate three architectures namely DetectNet, Faster-RCNN and SSD on a set of 170 crowd-sourced

refrigerator scenes with variation in cues, scale, illumination and blur, covering 150 distinct objects of interest considered as positives and 25 distractor objects as negatives. Figure 3 shows the variety in test set. The best model achieves mAP @ 0.5 IOU of 52.59 on this dataset which is a promising result considering that no real images were used while fine-tuning. In order to compare our performance with benchmark results, we consider Faster-RCNN (with ResNet-101 as feature mapping network) pretrained on COCO dataset as baseline model. Considering objects classes like bottle, cup and bowl from COCO object detection dataset as objects of interest, 24.15% mAP @ 0.5 IoU and 20.27% mAP @ 0.7 IoU is achieved on our test set. Figure 4 shows few images where our model outperforms the predictions from baseline model. It is observed that objects missed by baseline model were detected by model trained with synthetic images.

We observe that detector handles scale, shape and texture variance. Though packing patterns like vertical stacking or highly oblique camera angles lead to false predictions. Few vegetables among the distractor objects are falsely predicted as objects of interest suggesting the influence of pre-training on real dataset and lack of variety in distractor objects models available in object repository.

Dataset A *Size* : 3576		
Architecture	*mAP@0.5IOU*	*mAP@0.7IOU*
FRCNN	49.24	39.61
SSD	23.97	10.92
DetectNet	32.53	11.83
Dataset B *Size* : 2750		
Architecture	*mAP@0.5IOU*	*mAP@0.7IOU*
FRCNN	49.35	24.13
SSD	17.83	5.17
DetectNet	35.31	9.38
Dataset C *Size* : 3696		
Architecture	*mAP@0.5IOU*	*mAP@0.7IOU*
FRCNN	52.59	37.28
SSD	30.86	17.79
DetectNet	30.32	7.83

Table 1. Performance on datasets with various architectures.

4 Conclusion

The study with selected layer fine-tuning and freezing gives an insight of the important cues for efficient transfer learning from synthetic to real while comparison in architectures shows the significance of region proposal networks in detecting class-agnostic objects of interest.

We further observe scope in understanding the factors affecting transfer learning from synthetic to real images. The reported experiments hinted that few deeper layers were required to be fine-tuned with domain specific dataset (images rendered from 3D models) to achieve better performance compared to using pre-trained CNN as feature extractor. Study related to the visualization of features transferred from synthetic dataset by controlling the visual cues like texture, background, color, shape, illumination shall be useful. Further improvement in the performance can be achieved by training CNNs for semantic segmentation using synthetic images and adding depth information to the training sets to help in cases with high degree of occlusion. The results also inspire to use similar approach to build object detector using rendered images for larger set of object classes within varying environments.

Acknowledgment

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

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, Curran Associates Inc. (2012) 1097–1105
2. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 07-12-June., IEEE (jun 2015) 1–9
3. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision **111**(1) (2014) 98–136
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **8693 LNCS**(PART 5) (2014) 740–755
5. Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (jun 2009) 248–255
6. Li, W., Duan, L., Xu, D., Tsang, I.W.: Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 36. (jun 2014) 1134–1148
7. Hoffman, J., Rodner, E., Donahue, J., Darrell, T., Saenko, K.: Efficient Learning of Domain-invariant Image Representations. In: ICLR. (jan 2013) 1–9
8. Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: LSDA: Large Scale Detection Through Adaptation. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, MIT Press (2014) 3536–3544

9. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE (jun 2011) 1785–1792
10. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning Transferable Features with Deep Adaptation Networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. (2015) 97–105
11. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Proceedings of the 27th International Conference on Neural Information Processing Systems, MIT Press (2014) 3320–3328
12. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting Batch Normalization For Practical Domain Adaptation. Arxiv Preprint **1603.04779**(10.1016/B0-7216-0423-4/50051-2) (mar 2016)
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic Data for Text Localisation in Natural Images. Arxiv Preprint **1604.06646**(10.1109/CVPR.2016.254) (apr 2016)
14. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep Domain Confusion: Maximizing for Domain Invariance. Arxiv Preprint **1412.3474** (dec 2014)
15. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. ICML'14, JMLR.org (2014) I–647–I–655
16. Tommasi, T., Patricia, N., Caputo, B., Tuytelaars, T. In: A Deeper Look at Dataset Bias. Springer International Publishing, Cham (2015) 504–516
17. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '14, Washington, DC, USA, IEEE Computer Society (2014) 1717–1724
18. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3D models. Proceedings of the IEEE International Conference on Computer Vision **2015 Inter** (dec 2015) 1278–1286
19. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. Proceedings of the IEEE International Conference on Computer Vision **2015 Inter** (may 2015) 2686–2694
20. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing Training Data for Object Detection in Indoor Scenes. Arxiv Preprint **1702.07836** (feb 2017)
21. Singh, A., Sha, J., Narayan, K.S., Achim, T., Abbeel, P.: BigBIRD: A large-scale 3D database of object instances. In: Proceedings - IEEE International Conference on Robotics and Automation, IEEE (may 2014) 509–516
22. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. Arxiv Preprint **1703.06907** (mar 2017)
23. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (jun 2016) 3234–3243
24. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. Arxiv Preprint **1511.07041**(10.1109/CVPR.2016.442) (nov 2015)
25. Yao, Y., Rosasco, L., Caponnetto, A.: On early stopping in gradient descent learning. Constructive Approximation **26**(2) (aug 2007) 289–315

26. Sharp, Toby: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on : date, 16-21 June 2012. IEEE (2012)
27. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Arxiv Preprint **1512.03012**(10.1145/3005274.3005291) (dec 2015)
28. 3D, A.: Archive 3D (2015)
29. Barker, J., Sarathy, S., July, A.T.: DetectNet : Deep Neural Network for Object Detection in DIGITS (2016)
30. Vlastelica, M.P., Hayrapetyan, S., Tapaswi, M., Stiefelhagen, R.: Kit at MediaEval 2015 - Evaluating visual cues for affective impact of movies task. In: CEUR Workshop Proceedings. Volume 1436., New York, New York, USA, ACM Press (2015) 675–678
31. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6) (jun 2017) 1137–1149
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
33. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 9905 LNCS. (2016) 21–37
34. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Volume 7574 LNCS. Springer, Berlin, Heidelberg (2012) 340–353