TRANSFER LEARNING FROM SYNTHETIC TO REAL IMAGES USING VARIATIONAL AUTOENCODERS FOR PRECISE POSITION DETECTION

Tadanobu Inoue[1], Subhajit Chaudhury[1], Giovanni De Magistris[1] and Sakyasingha Dasgupta[2†]

[1]IBM Research, Japan. {inouet, subhajit, giovadem}@jp.ibm.com
[2]Ascent Robotics Inc., Japan, sakya@ascent.ai

ABSTRACT

Capturing and labeling camera images in the real world is an expensive task, whereas synthesizing labeled images in a simulation environment is easy for collecting large-scale image data. However, learning from only synthetic images may not achieve the desired performance in the real world due to a gap between synthetic and real images. We propose a method that transfers learned detection of an object position from a simulation environment to the real world. This method uses only a significantly limited dataset of real images while leveraging a large dataset of synthetic images using variational autoencoders. Additionally, the proposed method consistently performed well in different lighting conditions, in the presence of other distractor objects, and on different backgrounds. Experimental results showed that it achieved accuracy of $1.5\,\mathrm{mm}$ to $3.5\,\mathrm{mm}$ on average. Furthermore, we showed how the method can be used in a real-world scenario like a "pick-and-place" robotic task.

Index Terms— deep learning, position detection, transfer learning, variational autoencoder, computer simulation

1. INTRODUCTION

Supervised deep learning tasks require a large collection of labeled data for producing generalizable performances of unseen test data, and in estimating the location of objects [1, 2, 3]. For image-based learning, it is time-consuming to capture and label camera images in the real world. In contrast, it is easy to synthesize and collect large-scale labeled images in a simulation environment. Since there is a "reality gap" between simulation and real environments, it is difficult to match performances in the real world by learning only from these synthesized images. Thus, there is a need to bridge the gap between real and simulated images to learn useful features in a cross-domain manner.

To overcome this gap, Shrivastava *et al.* [4] proposed a method to generate realistic images by refining synthesized ones using adversarial deep learning. Santana and Hotz [5] combined variational autoencoders (VAE) [6] and generative
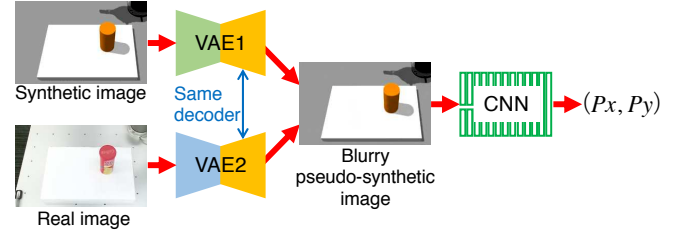
† This work was carried out during his position with IBM Research.

Fig. 1. Proposed concept for detecting object position

adversarial networks (GAN) [7] to generate realistic road images. To generate these realistic images, the approaches needed numerous unlabeled real images for adversarial training during refinement. However, collecting a large number of unlabeled real images can be extremely time-consuming and difficult in some cases. For example, it might be cumbersome to capture all possible combinations of object types and locations in a given scene configuration.

Domain randomization approaches [8, 9, 10, 11] provide promising methods to overcome the reality-gap by training multiple random configurations in simulations without many real images. It was argued that if the network was trained on multiple random configurations, a real image could also be treated as yet another random configuration. The literature also shows examples, where domain adaptation methods have been used for wider robotic applications [12, 13, 14, 15, 16] beyond vision-based tasks [17, 18].
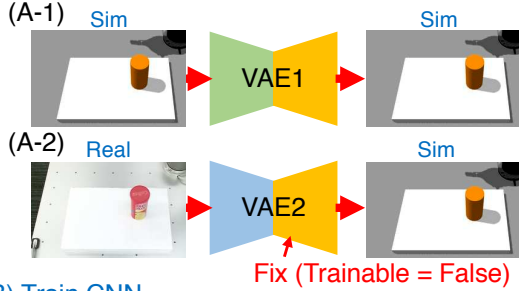
We propose a transfer learning method for detecting real object positions from RGB-D image data using two generative models that generate common pseudo-synthetic images from synthetic and real images. This method uses a significantly limited dataset of real images, which are typically costly to collect while leveraging a large dataset of synthetic images that can be easily generated in a simulation environment. Furthermore, the proposed model remains invariant to changes in lighting conditions, the presence of other distractor objects, or backgrounds. The obtained precision in detecting the position of the desired object ensures real-world application potential. We demonstrate its application in a typical robotic "pick-and-place" task as shown in the video (https://youtu.be/30vji7nJibA).
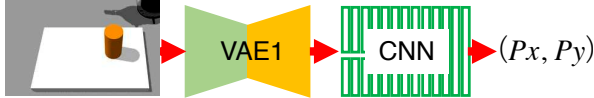
2. DETECTING OBJECT POSITIONS USING VAES

Figure 1 depicts the concept of our proposed method. The core idea of this work is that the distribution of image features may vary between simulated and real environments, but the output label of the object position should remain invariant for the same scene.

Our method consists of three broad steps as shown in Fig. 2. We use two VAEs for generating similar common images from synthetic and real image data and use this common image data to train a convolutional neural network (CNN) for predicting the object position with improved accuracy. Here, although two VAEs have distinct encoder layers as generative models for images, they have the same decoder, which is used to train the CNN. Thus, even if the VAE generates blurry images, the CNN will learn to predict object positions from this skewed but common image space. Furthermore, since the CNN can be trained with many generated images from the synthetic domain, we can achieve improved object position estimation from a significantly limited set of labeled real images.

Fig. 2. Three steps of our proposed method: (A) Train two VAEs sequentially; (A-1) Train VAE1 to generate synthetic images; (A-2) Train VAE2 to generate synthetic images from real images; (B) Train CNN to detect object position using VAE1 outputs with synthetic images; (C) Detect real object positions using VAE2 outputs with real images and CNN

2.1. Variational real to synthetic mapping

Two VAEs are prepared to generate common pseudo-synthetic images from synthetic and real images. A simulation environment is set up and large-scale synthetic images are captured along with corresponding ground-truth object position labels. We train VAE1, which encodes and decodes from a synthetic image to the same synthetic image as shown in Fig. 2 (A-1).

The weights from VAE1 is used to initialize another VAE network with the same structure (VAE2). This VAE learns a conditional distribution that encodes and decodes from a real image to the corresponding synthetic image as shown in Fig. 2 (A-2). During the training, we fix the decoder layers and adapt only the parameters for the encoder, which receives the real images as input. This is equivalent to forcing the latent space obtained from the synthetic and real images to be identical. The learned encoder and decoder can be combined to generate pseudo-synthetic images as output from the corresponding real image as input.

2.2. Object detection on common image space

A CNN is trained to detect object positions as shown in Fig. 2 (B). To close the gap between synthetic and real images, we use the outputs of the trained VAE1 in the previous step, instead of using synthetic images directly. Since both VAE1 and VAE2 use the same decoder (generator), the image space of both the outputs is the same which enables cross-domain transfer of learned tasks. This forms the primary idea that is presented in this paper. Due to the availability of a large training dataset synthesized in a simulation environment, we can train the CNN adequately to obtain an accurate object detector.

Finally, in the test phase, object positions are detected in the real world as shown in Fig. 2 (C). In this case, VAE2 outputs blurry pseudo-synthetic common images, and the CNN trained with the similar common images outputs the object position.
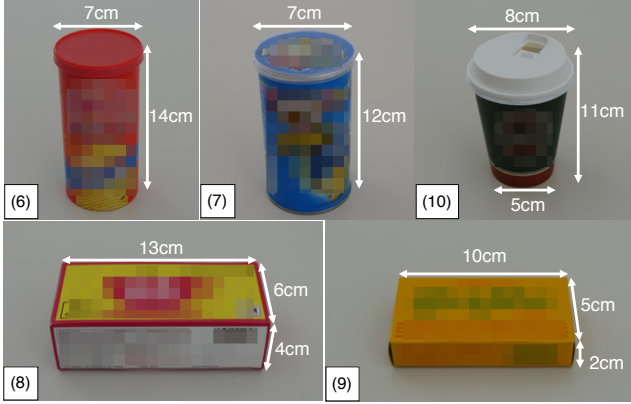
3. EXPERIMENTS

3.1. Experimental setup

For all experiments, we used a Gazebo® [19] simulation environment and Kinect® [20] camera. In Gazebo, object models were located on a white styrofoam (45×30 cm) at specific positions. A corresponding Kinect model was also loaded in Gazebo to capture RGB-D images of the workspace scene. At the same time, objects were manually located on a same-sized white styrofoam at the specific positions in the real world. Furthermore, the images captured in both Gazebo and the real world were cropped to a smaller region.

Our method was evaluated in 14 experiments (a)-(n). We used five simple real objects created by a 3D printer as shown in Table 1, for experiments (a)-(g) and five complex textured household objects as shown in Fig. 3, for experiments (h)-(n).

Table 1. Simple real objects

No.	Experiments	Color	Shape	Size (cm)
(1)	(a), (b)	red	cube	$5 \times 5 \times 5$
(2)	(c), (f), (g)	green	cube	$4 \times 4 \times 4$
(3)	(d), (g)	black	cylinder	radius 3.5 height 1
(4)	(e), (g)	blue	triangular prism	radius 4.5 height 1
(5)	(g)	red	cube	$4 \times 4 \times 4$

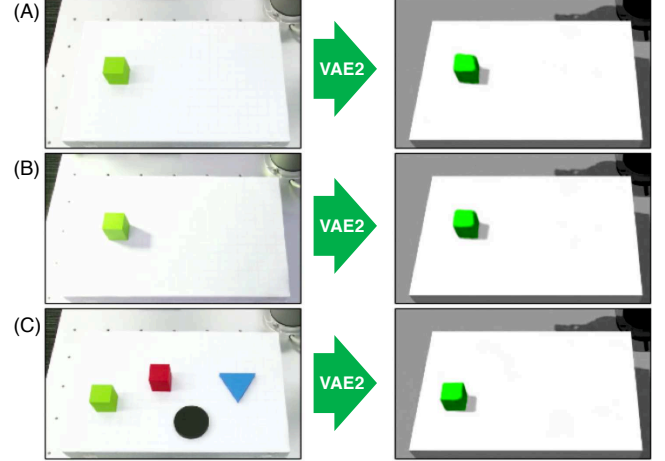**Fig. 3**. Complex textured household objects

3.2. Evaluation on simple real objects

First, as a baseline experiment (a), we evaluated the position detection of a red cube (1) in a naive manner, using the CNN trained with 54 real images at 5 cm grid positions directly. Then, our method was applied to the red cube (1), green cube (2), black cylinder (3), and blue triangular prism (4) (experiments (b)-(e)). We used 4131 synthetic images in which each synthetic object was located at 5 mm grid positions for training VAE1. Only 54 real and 54 synthetic images in which each object was located at 5 cm grid positions for training the VAE2 were used. This real image data was not augmented [21].

In experiment (b), the mean squared error (MSE) of synthetic and real input images was 9.9×10^3 on average and the MSE of two VAE output images was 2.7×10^{-3} on average. Our method using VAEs improved the similarity between images in real and synthetic domains for the inputs of later CNNs, which then overcame the "reality gap."

Subsequently, the strength of the method was assessed in different lighting conditions. We usually kept the experimental space light turned on during the two VAEs training. In the test phase, the room light was turned off and a table light was turned on instead, for creating a different lighting condition. The images in Figs. 4 (A) and 4 (B) on the left-hand side are raw images captured by the physical Kinect. The brightness levels of the captured images were controlled by Kinect's

auto-brightness functionality, but we saw that a shadow from the green cube was changed between different conditions. As observed from the right side of Figs. 4 (A) and 4 (B), in both cases the VAE2 learned to generate very similar pseudo-synthetic images regardless of the lighting differences.

Fig. 4. Images generated by VAE2 under different lighting conditions and with the presence of distractor objects: (A) Room light turned on and table light turned off; (B) Room light turned off and table light turned on; (C) Scene containing green cube (2), red cube (5), black cylinder (3) and blue triangular prism (4)

In the second set of experiments, we evaluated the validity of our method against the presence of multiple distractor objects. The VAE2 was trained with only a single green cube object and it was subjected to the newly captured images with multiple objects without any further re-training. As shown in the right side of Fig. 4 (C), the VAE2 continued to successfully generate pseudo-synthetic images with only the green cube while completely ignoring the other objects in the same scene. Therefore, this selectivity is quite useful for detecting the position of target objects even in the presence of numerous distractor objects of varying colors and shapes.

Upon successfully learning to generate common images with the VAEs, the CNN was trained to detect object positions using 4131 VAE1 outputs from the synthetic images generated in Gazebo. Figure 5 shows the experimental results of prediction errors for the above cases. Compared to the baseline results shown in Fig. 5 (a), our method (Fig. 5 (b)) showed a considerable reduction in prediction errors. Our method was successfully applied to differently shaped objects (Figs. 5 (b)-(e)), and performed well in different lighting conditions and with other objects present (Figs. 5 (f)-(g)).

3.3. Evaluation on complex textured household objects

In addition, our method was applied to more complex textured objects (6)-(10) shown in Fig. 3. We tried to detect the posi-
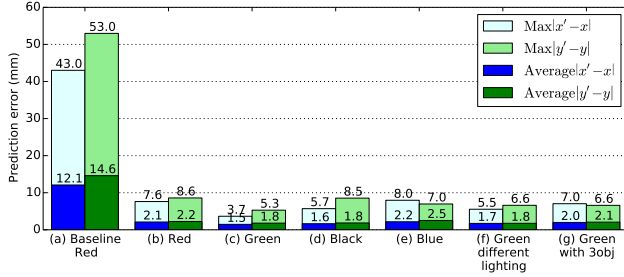
Fig. 5. Experimental results for prediction errors: (a) Baseline for red cube; (b) red cube; (c) green cube; (d) black cylinder; (e) blue triangular prism; (f) green cube under different lighting condition; and (g) green cube with three other objects

tion of object (6) from the scenes where three objects (6)-(8) were located on a white styrofoam (experiments (h) and (i)). As a baseline, we evaluated the outputs of the CNN trained directly by being given 54 real images (experiment (h)). In the real images, object (6) was located at 5 cm grid positions with objects (7) and (8) which were located randomly on a white background. We used 4131 synthetic images in which a simplified orange cylinder was located at 5 mm grid positions in Gazebo for training the VAE1 and the 54 real images of three objects ((6)-(8)). Then, a corresponding 54 synthetic orange cylinder images were used for training the VAE2 (experiment (i)).
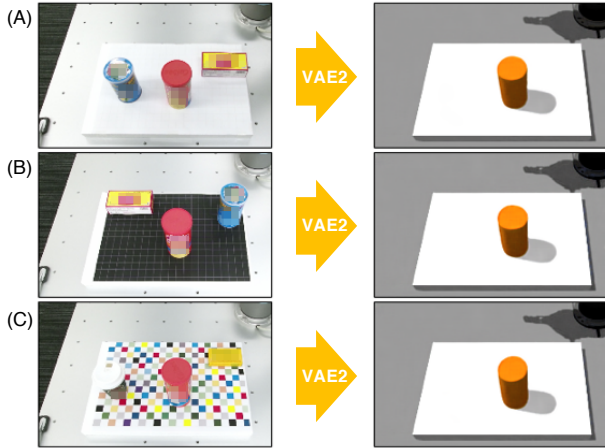

Fig. 6. Images generated by VAE2 in more complex textured object cases: (A) objects (6), (7), and (8) on a white background; (B) objects (6), (7), and (8) on a black background (C) object (6), (9), and (10) on a colorful checkered background

Subsequently, we assessed its performance against unseen backgrounds and different distractor objects without any retraining in the test phase. Black paper and colorful checkered paper were located on the white styrofoam as unseen back-

grounds (experiments (j) and (k)). We tried unseen object combinations with (6), (9), and (10) on the unseen black and checkered backgrounds (experiments (l) and (m)). We also tested when there was only a single object (6) on the unseen checkered background (experiment (n)).

Figure 6 shows outputs of the VAE2 trained with real images of (6), (7), and (8) on a white background. The VAE2 could extract the object (6) in real images and reconstruct corresponding simplified orange cylinder shapes, even if the there were changes in backgrounds and other distractors.

Figure 7 shows the experimental results of prediction errors for experiments (h)-(n). Since the object colors were complex, the baseline result (Fig. 7 (h)) was considerably degraded compared to the simple color case in Fig. 5 (a). On the other hand, our method could suppress the degradation on prediction errors shown in Fig. 7 (i). We had an accuracy of 1.5 mm to 3.5 mm on average on different backgrounds and with different object combinations.
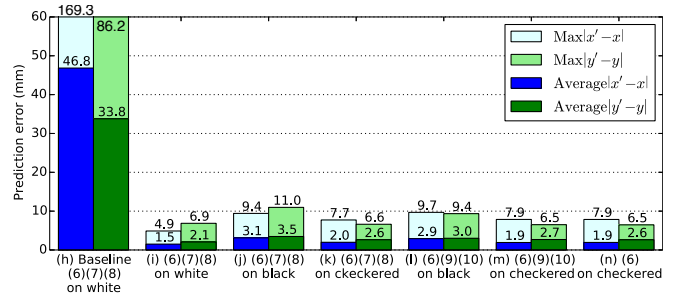

Fig. 7. Experimental results for prediction errors: (h) Baseline with objects (6), (7), and (8) on white background; (i) objects (6), (7), and (8) on white background; (j) objects (6), (7), and (8) on black background; (k) objects (6), (7), and (8) on checkered background; (l) objects (6), (9), and (10) on black background; (m) objects (6), (9), and (10) on checkered background; and (n) Single object (6) on checkered background

Finally, we tested for precise position detection in a robotic "pick-and-place" task, as shown in the video (https://youtu.be/30vji7nJibA).

4. CONCLUSION

We presented a transfer learning method using two VAEs to detect object positions precisely using only a significantly limited dataset of real images while leveraging a large dataset of synthetically generated images. Our method performed solidly in different lighting conditions, with other objects present, and on different backgrounds. It achieved accuracy of 1.5 mm to 3.5 mm on average. We also demonstrated its efficiency in a real-world robotic application, like "pick-and-place" task.

5. REFERENCES

[1] J. Leitner, S. Harding, M. Frank, A. Forster, and J. Schmidhuber, "Artificial neural networks for spatial perception: Towards visual object localisation in humanoid robots," in *International Joint Conference on Neural Networks (IJCNN)*, 2013.

[2] A. Collet and S. S. Srinivasa, "Efficient multi-view object recognition and full pose estimation," in *International Conference on Robotics and Automation (ICRA)*, 2010.

[3] J. Tang, S. Miller, A. Singh, and P. Abbeel, "A textured object recognition pipeline for color and depth image data," in *International Conference on Robotics and Automation (ICRA)*, 2012.

[4] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] E. Santana and G. Hotz, "Learning a driving simulator," in *arXiv:1608.01230*, 2016.

[6] D.P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. OzairA. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems (NIPS)*, 2014.

[8] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," in *Robotics: Sicence and Systems (RSS)*, 2017.

[9] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep nerual networks from simulation to the real world," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[10] M. Yan, I. Frosio, S. Tyree, and J. Kautz, "Sim-to-real transfer of accurate grasping with eye-in-hand observations and continuous control," in *Neural Information Processing Systems (NIPS) Workshop on Acting and Interacting in the Real World: Challenges in Robot Learning*, 2017.

[11] F. Zhang, J. Leitner, M. Milford, and P. Corke, "Sim-to-real transfer of visuo-motor policies for reaching in clutter: Domain randomization and adaptation with modular networks," in *arXiv:1709.05746*, 2017.

[12] A. A. Rusu, M. Vecerik, T. Rothorl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real roboto learning from pixels with progressive nets," in *arXiv 1610.0428*, 2016.

[13] I. Higgins, A. Pal, A. A. Rusu, L. Matthey, C. P. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchne, "Darla: Improving zero-shot transfer in reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2017.

[14] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[15] S. James, A. J. Davison, and E. Johns, "Transferring end-to-end visiomotor control from simulation to real world for a multi-stage task," in *1st Annual Conference on Robot Learning (CoRL)*, 2017.

[16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *arXiv:1710.06537*, 2017.

[17] X. Peng, B. Sun, K. Ali, and Kate Saenko, "Learning deep object detectors from 3d models," in *International Conference on Computer Vision (ICCV)*, 2015.

[18] B. Kulis, K. Saenko, and Trevor Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[19] Open Source Robotics Foundation, "Gazebo simulator," http://gazebosim.org.

[20] Microsoft, "Kinect for xbox one," http://www.xbox.com/en-US/xbox-one/accessories/kinect.

[21] Q. Bateux, E. Marchand J. Leitner, F. Chaumette, and P. Corke, "Visual servoing from deep neural networks," in *Robotics: Science and Systems (RSS), Workshop New Frontiers for Deep Learning in Robotics*, 2017.