Synthetic-Real Domain Adaptation for Probabilistic Pose Estimation

Omar Del-Tejo-Catala
Instituto Tecnologico de Informatica
Camino de Vera, S/N
Spain 46022, Valencia
odeltejo@iti.es

Javier Perez
Instituto Tecnologico de Informatica
Camino de Vera, S/N
Spain 46022, Valencia
javierperez@iti.es

Jose-Luis Guardiola
Universidad Politecnica de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
joguagar@iti.es

Alberto J. Perez
Universidad Politecnica de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
aperez@disca.upv.es

Juan-Carlos Perez-Cortes
Universidad Politecnica de Valencia
Camino de Vera, S/N
Spain 46022, Valencia
jcperez@iti.es

ABSTRACT

Real samples are costly to acquire in many real-world problems. Thus, employing synthetic samples is usually the primary solution to train models that require large amounts of data. However, the difference between synthetically generated and real images, called domain gap, is the most significant hindrance to this solution, as it affects the model's generalization capacity. Domain adaptation techniques are crucial to train models using synthetic samples. Thus, this article explores different domain adaptation techniques to perform pose estimation from a probabilistic multiview perspective. Probabilistic multiview pose estimation solves the problem of object symmetries, where a single view of an object might not be able to determine the 6D pose of an object, and it must consider its prediction as a distribution of possible candidates. GANs are currently state-of-the-art in domain adaptation. In particular, this paper explores CUT and CycleGAN, which have unique training losses that address the problem of domain adaptation from different perspectives. This work evaluates a patch-wise variation of the CycleGAN to keep local information in the same place. The datasets explored are a cylinder and a sphere extracted from a Kaggle challenge with perspective-wise symmetries, although they holistically have unique 6D poses. One of the main findings is that probabilistic pose estimation, trained with synthetic samples, cannot be solved without addressing domain gap between synthetic and real samples. CUT outperforms CycleGAN in feature adaptation, although it is less robust than CycleGAN in keeping keypoints intact after translation, leading to pose prediction errors for some objects. Moreover, this paper found that training the models using synthetic-to-real images and evaluating them with real images improves the model's accuracy for datasets without complex features. This approach is more suitable for industrial applications to reduce inference overhead.

Keywords

Pose Estimation, CycleGAN, Image-to-image, Graph Neural Networks, UNet, Domain adaptation, CUT, Symmetry Robust Pose Estimation

1 INTRODUCTION

Obtaining a large number of real samples to train a model is often expensive, making it infeasible for many industrial applications. To address this issue, researchers have proposed training models with synthetic samples, which are cheaper and easier to produce on a large scale [Sve21a, Che21a, Sha22a]. However, synthetic samples cannot simulate every nuance in the real samples domain, resulting in a domain gap. Techniques such as domain adaptation and domain randomization have been proposed to address this issue. This paper uses domain adaptation techniques to reduce the domain gap between synthetic and real samples.

Pose estimation has many practical applications, including robot grasping, virtual reality, and robot localization. Various contexts have been proposed to evaluate different working scenarios, such as pose estimation in cluttered scenes [Buk20a, Ste13a, YuX17a], relative camera pose estimation [Che21b], and pose refinement [YiL20a]. This article focuses on multiview probabilistic uncluttered pose estimation of perspective-symmetric objects, as presented in [Oma21a].

The data capturing environment for this work revolves around a quality inspection system [Jua18a], which comprises several cameras distributed along a sphere that captures objects under inspection in a controlled environment. The objects are free-falling, and the cameras capture them without occluded faces. However, the 6D pose is uncontrolled, and some keypoints are not visible from all perspectives. Hence, the algorithm's first stage must account for auto-occlusions and partially informative perspectives, which might only allow the algorithm to narrow down the range of possible solutions. Therefore, each image prediction should be an estimation of the true 6D pose, and the final model's prediction is computed by combining multiple views.

This paper evaluates how different state-or-the-art domain adaptation techniques affect probabilistic pose estimation, which is also compared against the baseline, i.e., not handling domain gap. Objects under inspection might have perspective-relative symmetry axes, although, considering the object as a whole, they might not be symmetric (see Figure 1). Therefore, the probabilistic pose estimation model must be able to model and detect those axes internally. The domain adaption algorithm must not interfere with that model's capacity.

Furthermore, this article compares pose estimation performed separately in the synthetic and real domains. Thus, it presents the results in several scenarios:

- Case 0: Training the pose estimation model with synthetic images and evaluating with real images.
- Case 1: Training the pose estimation model with domain-translated synthetic images, i.e., synthetic-to-real images, and evaluating with real images.
- Case 2: Training with synthetic images and testing with domain-translated real images, i.e., real-to-synthetic images.

The work is organized as follows. Section 2 presents the pose estimation's state-of-the-art and several methods to address the domain gap. Section 3 presents the datasets evaluated along with the algorithms used, their structure, and their losses. Section 4 presents the results achieved by the different domain adaption algorithms for the pose estimation tasks related to the datasets. Section 4.3 compares and discusses the results achieved with previous work in pose estimation and domain adaptation. Section 5 summarizes the articles' findings.

2 RECENT SOLUTIONS

2.1 Pose Estimation

Pose estimation is a well-established field that has been extensively studied [Wad17a, Tom18a, Buk20a]. Recent approaches include per-pixel pose inference, bounding box detection, prediction refinements, and direct pose regression. However, many of these models do not effectively handle symmetries. Some models manually address symmetries during training [Wad17a, Buk20a], while others account for them during metric computation, such as ADD and ADDS [Tom18a, Bug18a], used in the LINEMOD dataset [Ste13a]. However, none of these approaches automatically model an object's symmetries. Therefore, we selected the work in [Oma21a], which proposes a probabilistic multiview approach for pose estimation. This approach enables the combination of multiple hypotheses based on uncertainty by modeling the probability distribution of the object's rotation, which may be uncertain from certain viewpoints.

2.2 Addressing domain gap

Domain bias [Csu2017] is a problem that models often encounter when trained on a single narrow-range dataset, limiting their generalization capabilities. Inconsistencies arise when models are trained with synthetic images and evaluated with real ones.

To address the difference between synthetic and real images, the literature primarily focuses on two approaches: domain randomization and domain adaptation.

Domain randomization [Sve21a, Tob18a, Tob17a] involves modifying synthetic generation hyperparameters to increase the diversity of synthetic image generation. In the context of pose estimation, such hyperparameters may include the position of the cameras, object textures, lighting, or scene context. Although this technique attempts to emulate the possible variability in real scenarios, it does not leverage the information from real samples and may not address crucial variability that could impact the model's generalization capability. For example,[Sve21a] generates multiple scenarios with cat and dog models in Unity, with different camera positions, object textures, and occlusion configurations. It evaluates the model using the public Kaggle Cats-Dogs dataset.[Tob17a] employs synthetic training with various textures and occlusions to address the task of object localization for robotic manipulation.

Domain adaptation [Che21a, Sha22a, Jac21a] minimizes the domain gap by decreasing the difference

between both domains' images or features. Usually, this is addressed using image-to-image transformation between domains, mainly using unsupervised techniques like autoencoders or GANs. Such approaches include employing CycleGANs [Che21b] or Contrastive Unpaired Translation (CUT) [Tae20a].

In [Jac21a], pose estimation and domain adaptation are addressed from a single-perspective non-symmetric approach. It asserts that CUT outperforms CycleGAN for domain adaptation in pose estimation. Domain adaptation techniques must keep keypoints along with the object symmetries intact.

Similar to [Jac21a], this paper evaluates different domain adaptation techniques, although it focuses on different symmetric objects. Thus, a slight variation of the CycleGANs [Che21b] and the CUT algorithm [Tae20a] are employed to reduce the domain gap in probabilistic pose estimation.

3 PROPOSED SOLUTION

3.1 Dataset

This paper's dataset expands some of the datasets presented in [Oma21a] and published in Kaggle [ITI21a]. We selected the cylinder, which has a perspective symmetry but a single valid object-level prediction, and the sphere, whose perspectives are mostly uninformative, except for the ones containing the "T".

The cylindrical object's bases contain a carved triangle on one side and a square on the other. The square and the triangle are aligned to create the object's reference point, albeit no camera can see both simultaneously. Therefore, the pose of the cylinder can only be calculated by combining the predictions from the cameras that were able to see both polygons separately. As presented in Figure 1, any camera that captures some polygon can predict the X-axis orientation. Nonetheless, for the Y-axis, the cameras that captured the triangle should return three equally likely Y-axis predictions. In comparison, the cameras that captured the square should return four equally likely Y-axis predictions. Finally, the combination of all separate Y-axis predictions must return a single likeliest Y-axis.

The sphere's only informative keypoint is the carved "T" on its surface. Most views are unable to see the "T" due to auto-occlusions. Those views only provide the information that they cannot see the "T". Thus, the prediction should be an equally like distribution over the subspace of 3D rotations that yield perspectives without the "T". Nonetheless, any perspective that completely captures the keypoint should return a unique 6D pose prediction.

The corresponding 3D CAD files were employed to generate real objects using a 3D printer. The files describe a cylinder with 50 millimeters of height and a diameter of 25 mm and a sphere of 30mm of diameter. Like any other generation process, this procedure is prone to generate singularities in the object's texture that may disclose the 6D pose instead of using the intended keypoints. Pose estimation algorithms can then overfit these singularities leading to errors in a real environment. Moreover, even the texture and keypoints present in the object's ideal mesh have some variability in the real generated objects. A comparison between ideal and real object captures can be seen in figure 1. For instance, the printed sphere has a stain on one side that may affect the pose estimation algorithm (see Figure 1e). This object provides a more complex non-uniform texture to evaluate the domain adaptation algorithm.

The printed objects were captured in an industrial quality inspection system, presented in [Jua18a], that comprises a multicamera environment. Capturing objects inside this system leads to sixteen camera images per capture. As the pose estimation algorithm presented in [Oma21a] leverages multiview images, their individual predictions are joined to a unique 6D pose prediction in testing phase.

The images from the printed objects contain the object in many different poses, as the objects are captured free falling inside a controlled environment [Jua18a]. Thus, the groundtruth pose for each capture must be manually labeled. Using OpenCV's CVAT labeling tool, some reference keypoints were selected from multiple views. These keypoints' correspondence was used to compute de 6D pose of each launch.

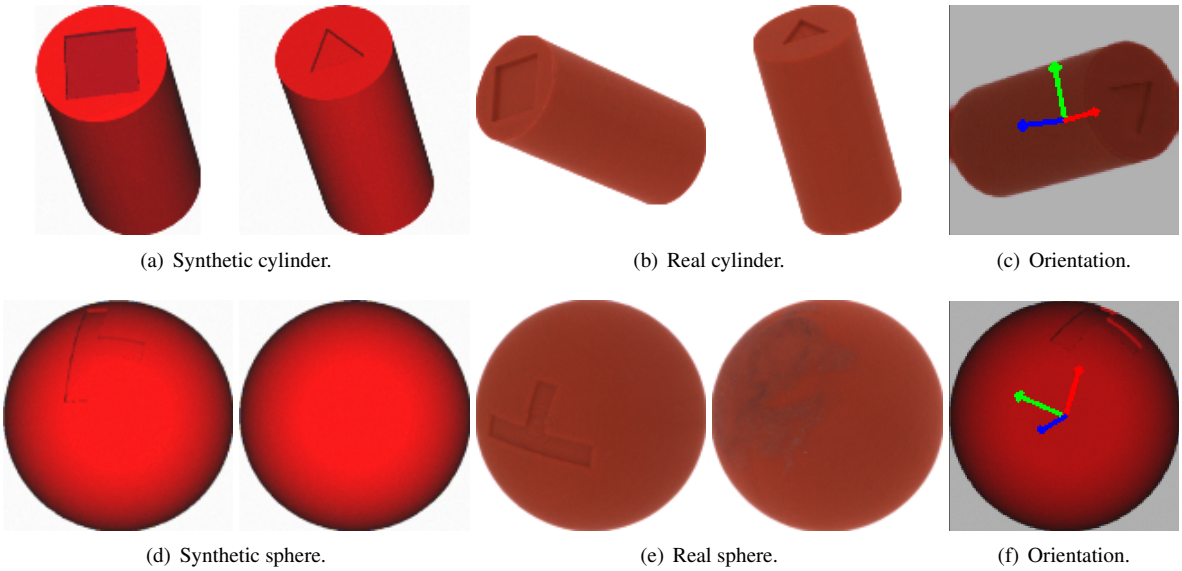
To train the models, this article synthetically simulates the real environment to capture the 3D CAD files, similar to other domain randomization authors [Sve21a]. The real-world objects are captured in a 16-camera sphere-distributed environment with diffuse white lighting; thus, the backgrounds are entirely white. Some sample images can be seen in Figure 1.

This dataset was made publicly available in Kaggle [ITI23a] [1].

3.2 Probabilistic pose estimation algorithm

This algorithm takes an image and some cropping-related information, i.e., pixel coordinate of the bounding box center, and a scaling factor applied to resize the image to 128×128 resolution. Then, the algorithm predicts the object's translation and a probability distribution for the rotation. This probability distribution is a discretization of the rotation's SO(3) group $\mathscr{R}^9$ space, which is compressed up to $\mathscr{R}^6$. In other words, it predicts a discretization of the rotation matrices' X and Y

[1] https://www.kaggle.com/ds/2947388



(a) Synthetic cylinder. (b) Real cylinder. (c) Orientation.

(d) Synthetic sphere. (e) Real sphere. (f) Orientation.

Figure 1: Train and test capture comparison to show the domain gap. The cylinder's X-axis corresponds to the object's longitudinal axis. The Y-axis points from the polygons' centers to the center of the only side of the polygon that aligns with some opposite polygon's side. The sphere's X-axis is the vector from the object's centroid to the "T". The Y-axis corresponds to the vertical orientation of the "T". In (c) and (f), red, green and blue axes correspond to X, Y and Z axes, respectively.

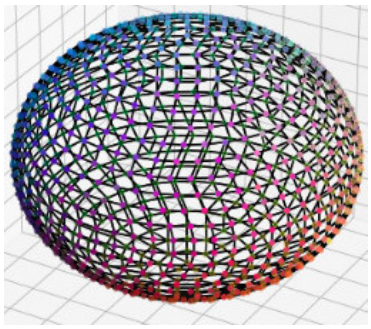


Figure 2: Discretization of the rotations' axis space [Oma21a].

axes components. As described in [Oma21a], this can be done without information loss due to the rotation matrices orthonormality.

These axes, belonging to the $\mathscr{R}^3$ unitary sphere, are discretized sampling N equidistant points. Hence, the algorithm's output is a tensor of $N \times 2$ for each image. The space discretization and how each point is connected as a graph can be seen in figure 2. The predictions are combined for all the images in a capture. Finally, this distribution is queried for the likeliest positions for the X and Y axes; thus, the Z axes can be inferred. More detail of the algorithm can be seen in [Oma21a].

The predictions can be unwrapped for visualization purposes. This projection consists of extracting the azimuth and elevation of each $\mathscr{R}^3$ point (i.e., a rotation matrix axis) and printing them in an image, being the intensity of each pixel related to the axis likelihood. We

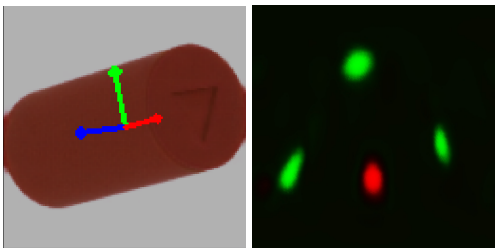


Figure 3: Prediction when the triangle is visible. The predicted distributions, i.e., one for the X-axis and one for the Y-axis, are projected to 2D space for visualization purposes. Red color corresponds to the X-axis activations and green to the Y-axis activations. It shows a single activation for the X-axis, as the direction of the longitudinal axis can be determined, and three activations for the symmetric triangle rotations.

can visualize each axis in different color channels, i.e., red for X-axes and green for Y-axes. Figure 3 shows the three possible Y-axis clustered activations that occur when predicting using the triangle due to its symmetries. Three equally likely rotations can be predicted using the likeliest axis of each green y-axis cluster and the likeliest axis of the red X-axis cluster. All those three equally likely activations are related to an identical image representation.

3.3 CycleGAN

To perform cross-domain evaluation, we must address the problem of the domain gap between training and testing sets. This paper employs CycleGANs as they force the image transformation to keep the information

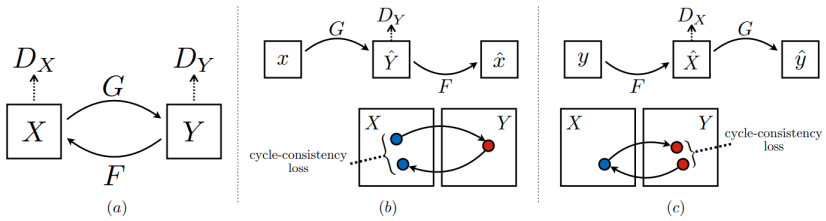


Figure 4: Cycle consistency from CycleGAN extracted from the original paper [Jun17a]. a) shows the structure of the network, X and Y being the domains. b) shows the cycle consistency loss for images from the X domain. c) shows the cycle consistency loss for the images from the Y domain.

when transforming the domain to be able to reconstruct the original image.

CycleGANs are divided into two different transformations, each with their corresponding generator G and discriminator D. Having two different domains d_A and d_B, we call $G_{A\to B}$ to the generator that transforms from domain d_A to d_B, $G_{B\to A}$ to the one that transforms domain d_B to d_A, D_A to the domain d_A discriminator, and D_B to the domain d_B discriminator.

3.3.1 Losses

CycleGANs have two losses: a cycle consistency loss and a discrimination confidence loss. Cycle consistency loss corresponds to the loss of projecting an image from one domain to the other and reprojecting it back to the original domain. This is, being X_A images from domain d_A and X_B images of domain d_B:

$$\hat{X}_A = G_{B\to A}(G_{A\to B}(X_A)) \tag{1}$$

$$\hat{X}_B = G_{A\to B}(G_{B\to A}(X_B)) \tag{2}$$

$$\mathscr{L}_{GCC} = ||\hat{X}_A - X_A||_1 + ||\hat{X}_B - X_B||_1 \tag{3}$$

A visual representation of the process can be seen in Figure 4.

Confidence loss corresponds to the traditional GAN loss originating from the discrimination of synthetic and real images. It is different for the generator $\mathscr{L}_{GC}$ and the discriminator $\mathscr{L}_{DC}$. Thus, these losses are described as follows:

$$\mathscr{L}_{GC} = ||D_B(G_{A\to B}(X_A))||_2 + ||D_A(G_{B\to A}(X_B))||_2 \tag{4}$$

$$\begin{aligned}\mathscr{L}_{DC} = &||1 - D_B(G_{A\to B}(X_A))||_2 + \\ &||1 - D_A(G_{B\to A}(X_B))||_2 + \\ &||D_A(X_A)||_2 + ||D_B(X_B)||_2\end{aligned} \tag{5}$$

The generator networks are updated with the sum of $\mathscr{L}_{GC}$ and $\mathscr{L}_{GCC}$, whilst the discriminators are updated with $\mathscr{L}_{DC}$.

3.3.2 Generators

As other authors have previously proposed [Tae20a, Dmi22a], this article uses UNet-like [Ola15a] structure for the generative network. A pretrained VGG16 performs as the network's encoder, extracting features at four different resolutions. Those features are then processed and upsampled by a decoder network, similar to a traditional UNet. The only trainable part of the network is the decoder.

Additionally, slight variations must be considered to keep the keypoints in place as best as possible. Therefore, patches of 32×32 were employed to force the network to store the information in an enclosed space. The image is reconstructed from its patches after each transformation. This process can be seen in Figure 5.

3.3.3 Discriminators

The discriminator also employs a pretrained VGG16 network that returns the features at two different resolutions: $32 \times 32 \times 256$ and $16 \times 16 \times 512$. At each resolution, the discriminator extracts trainable features and predicts their probability of being generated. Those probability maps are resized to the largest resolution and averaged.

It is a common practice to add noise to the discriminator input images to avoid overfitting. Thus, 0.1 intensity uniform noise was applied as a preprocess to the discriminator images.

In summary, discriminators take noisy $128 \times 128 \times 3$ images and return a 32×32 tensor with the probability of an image region being generated.

3.4 Contrastive Unpaired Translation

Contrastive Unpaired Translation (CUT) [Tae20a] is a domain adaptation technique that learns a transformation between two domains. It is designed without CycleGAN's cycle consistency loss and without reprojecting to the original domain. Therefore, it is faster than CycleGAN and is reported to achieve better results [Tae20a]. It relies on mutual information maximization using PatchNCE loss. This loss minimizes the distance between equivalent image patches in both domains, i.e., comparing the original patch with its translated version (positive) while maximizing the distance

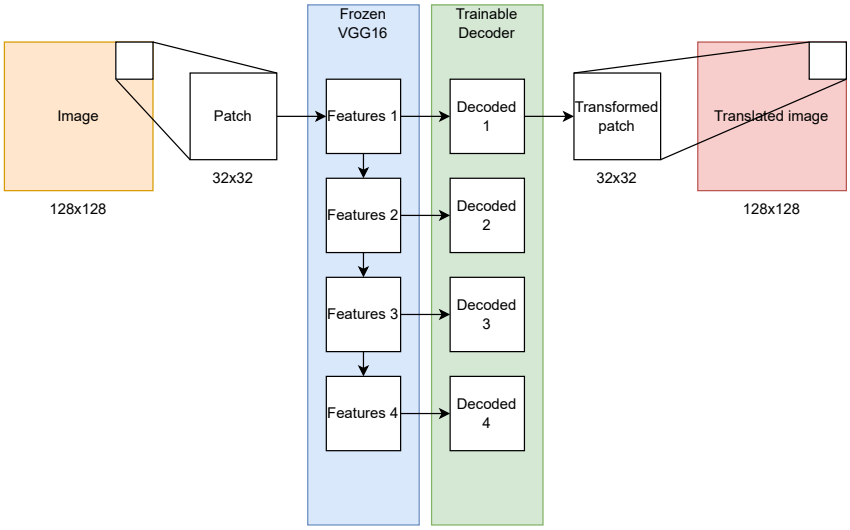


Figure 5: Image domain transformation process.

with the remaining image patches (negatives). This paper also studies the inclusion of negatives from other images but concludes that it hampers training.

4 RESULTS

4.1 Experimental setup

This article's experiments were designed using Tensorflow 2.10. Domain adaptation techniques were trained using 3000 synthetic images and 160 real images. Random rotations were applied to both domain images for data augmentation. The CUT model was trained using the code provided by the authors in Github[1]. It was trained for 400 epochs, removing the default random cropping and scaling and adding random rotations. Thus, the original $128 \times 128 \times 3$ image resolution is kept and there is some data augmentation. CycleGAN was trained by adding 0.1 standard deviation gaussian noise to the upsampling decoder inside synthetic-to-real generator, and data was augmented with random rotations.

The pose estimation models were trained using 256000 synthetic images, validated using 1600 synthetic images and tested using 880 real images. During training, random noise, dropout and L2 norm were added to increase generalization.

4.2 Case results

Three cases are proposed to test the two different domain transformations scenarios:

- Case 0 - no domain transformation. The model was trained with synthetic images, data augmentation, and regularization added to the model.
- Case 1 - training with real samples. Training images are translated to the real domain. The last four training checkpoints were used to include some variability in the domain transformation using CUT models. Therefore, the same synthetic image results in different real images. CycleGAN model has some embedded randomness while projecting from synthetic to real domains.
- Case 2 - evaluating with synthetic samples. Real images are projected to the synthetic domain to remove production singularities from the object's texture.

Some domain transformation samples can be seen in Figure 6. On the one hand, as seen from a visual inspection, when projecting from the real to the synthetic domain (Case 2), images still have some artifacts that do not belong to the synthetic domain. These include some modifications in the object's keypoints (Figure 6e) for CUT and random dirt (Figure 6f) for CycleGAN. On the other hand, both domain adaptation techniques perform correctly for this task when projecting from synthetic to real domain (Case 1). Moreover, as CycleGAN has intrinsic randomness, it learned to add some artifacts (see the square inside in Figure 6c).

Despite our best efforts, we could not train a valid CUT domain adaptation model for the sphere dataset. A sphere image without a "T" is still valid for the discriminator. Thus, these models converge to a state where all images projected do not keep the "T" keypoint.

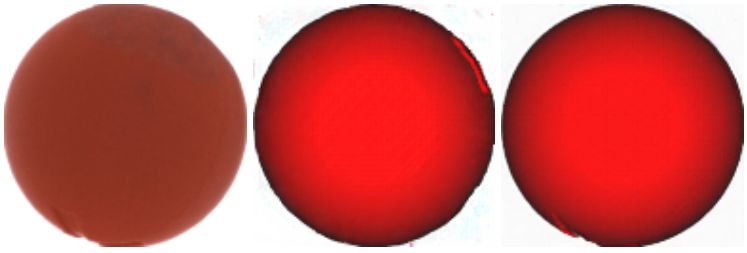
For a quantitative comparison, Frechet-Inception-Distance (FID) [Mar17a] was used to measure which domain adaptation algorithm performed best. This score measures the distance between the original and domain-shifted image distributions. The score's magnitude depends on the dataset; thus, different dataset's FID scores cannot be compared.

[1] https://github.com/taesungp/contrastive-unpaired-translation/. Commit: 7 Jun 2022.

(a) Real test image. (b) Translated real image using CUT. (c) Translated real image using CycleGAN.

(d) Synthetic equivalent. (e) Translated synthetic image using CUT. (f) Translated synthetic image using CycleGAN.

Figure 6: Results for cylinder domain transformation.



(a) Real test image. (b) Translated real image using CUT. (c) Translated real image using CycleGAN.

(d) Synthetic equivalent. (e) Translated synthetic image using CUT. (f) Translated synthetic image using CycleGAN.

Figure 7: Results for cylinder domain transformation.

As seen for the cylinder case in Table 1, CycleGAN performs better than CUT in Case 1, but CUT outperforms CycleGAN in Case 2. This difference is probably because CycleGAN has higher variability in the real domain due to its randomness. However, as stated above, CycleGAN has proven to be more robust than CUT for the sphere.

As seen in Table 2, training the network directly with real images improves the model's accuracy for the cylinder dataset. However, the sphere dataset results show an improvement when the pose estimation model is trained with synthetic images. A more in-depth analysis of the results is performed in the following section.

4.3 Discussion

As stated above, one of the main findings of this work is that training the model with synthetic-to-real images or using synthetic images provides different results depending on the object under inspection.

Unlike having to translate incoming images dynamically, the synthetic-to-real training approach has the

Case	Domain adaptation	FID	
		Cylinder	Sphere
Case 1	CycleGAN	49.33	47.25
Case 1	CUT	53.13	96.41*
Case 2	CycleGAN	82.45	43.45
Case 2	CUT	64.77	108.37*

Table 1: Frechet-Inception-Distance (FID) between different domain adaptation methods. (*) marks the experiments where the domain adaptation did not yield visually acceptable results.

Validation					
Case	Domain adaptation	Rotation loss (degrees)		Translation loss (mm)	
		Cylinder	Sphere	Cylinder	Sphere
Case 0 & 2	None	0.9 ± 0.05	2.0 ± 0.09	0.7 ± 0.03	1.3 ± 0.06
Case 1	CUT	1.4 ± 0.08	1.5 ± 0.08*	0.7 ± 0.03	1.3 ± 0.06
Case 1	CycleGAN	1.0 ± 0.06	1.2 ± 0.06	0.7 ± 0.03	1.3 ± 0.06
Test					
Case 0	None	111.9 ± 7.23	130.9 ± 4.31	2.6 ± 0.14	3.8 ± 0.63
Case 1	CUT	1.9 ± 0.11	158.3 ± 4.57*	2.7 ± 0.14	3.8 ± 0.63
Case 1	CycleGAN	5.2 ± 2.03	10.9 ± 3.05	2.6 ± 0.14	3.8 ± 0.63
Case 2	CUT	23.0 ± 5.45	123.1 ± 5.12*	2.7 ± 0.14	3.8 ± 0.63
Case 2	CycleGAN	23.7 ± 5.82	5.7 ± 0.39	2.7 ± 0.14	3.8 ± 0.63

Table 2: Validation and testing phase pose errors for each case and domain adaptation method. (*) marks the experiments where the domain adaptation did not yield visually acceptable results. Case 0 and case 2 share validation results because their validation set is synthetic, although test sets are different, i.e., synthetic and real images, respectively.

added benefit that no overhead is added after deploying the model; all overhead devoted to domain adaptation is performed during training. Contradicting the hypotheses presented by [Sha22a], at least for some objects without high variability textures, these findings show that training using synthetic-to-real images might be helpful to emphasize features that are more subtle in the synthetic domain by translating them to the real domain. The synthetic image generation algorithm might have a limited capacity for generating hyper-realistic object captures with precise shadows. Thus, an algorithm that projects those synthetic images to a domain with more perceptible features will improve the model's performance.

However, for cases where the textures are more complex, for instance, high variability textures such as wood or textures with anomalies, working only on the synthetic domain improves the pose estimation's results. This improvement arises because domain adaptation and pose estimation techniques trained on the real domain would require to be fit to a more challenging texture distribution. The synthetic domain provides a more accessible working environment for both models.

Furthermore, as can be seen in Table 2 from the comparison of Case 0 with any other case, applying some domain adaptation method significantly improves the model's generalization capacity. In most cases, the domain gap between synthetic and real samples cannot be solved by adding traditional regularization to the model.

CycleGAN, due to its patch-wise domain transformation to the original domain, keeps the object's keypoints in place better than CUT. This difference leads to a more robust domain adaptation algorithm for different objects. However, using CUT improves the model's synthetic-real generalization capacity for some objects, which is reflected in the model's cylinder test accuracy. Not only is it CUT a lighter algorithm, as it only performs one-way domain transformation, but it also achieves a better transformation, though less generic. Thus, as found in [Tae20a], CUT's PatchNCE loss outperforms CycleGAN's cycle consistency in performing domain transformations to similar features, but the fact that those features do not require reprojection to the original domain leads to information loss, which is highly detrimental for pose estimation.

5 CONCLUSION

Considering the results, the main conclusion is that domain gap should be addressed somehow to train a model with synthetic samples to solve the pose estimation task effectively, being CycleGAN and CUT valid for this purpose. From the baseline, where the model could not estimate the object rotation, adding some domain adaptation technique (either in case 1 or case 2 scenarios) reduced the rotation error below 6º for both datasets. Additionally, to address the domain gap effectively, this work found that training in the synthetic

or real domains depends mainly on the object texture's complexity. However, performing domain adaptation in training phase, i.e., training with real images, reduces computational costs after deployment. Computational costs in industrial applications are crucial, and adding computational costs to the offline training phase has a lower impact than domain transforming every incoming image to infer the 6D pose.

Moreover, this paper demonstrates that CycleGAN largely preserves the keypoint's positions so that a pose estimation algorithm can be trained to a reasonable accuracy. This fact is especially remarkable in this dataset due to the scarcity of keypoints from which to extract rotational information. CUT performs a better feature transformation, as it outperforms CycleGAN in the cylinder dataset. However, it might not prevent the loss or modification of keypoints during domain transformation, which leads to poor results for the sphere dataset.

As further research, including objects with anomalous textures to the domain adaptation and pose estimation algorithms, might provide valuable insight to improve the algorithms' robustness.

5.1 Acknowledgements

This work was funded with grants from the Generalitat Valencia with grant number IMAMCA/2023/11, the European Regional Development Fund (ERDF) with grant number IMDEEA/2022/46, and CDTI's CEL.IA project with grant number CER-20211022.

6 REFERENCES

[Csu2017] Gabriela Csurka A comprehensive survey on domain adaptation for visual applications. *Advances in Computer Vision and Pattern Recognition*, 1-35, 2017.

[Ste13a] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 548–562. Springer Berlin Heidelberg, 2013.

[Ola15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[Jun17a] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017

[Tob17a] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[Wad17a] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 1530–1538, nov 2017.

[YuX17a] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes *Robotics: Science and Systems (RSS)*, 2018.

[Bug18a] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 292–301, nov 2018.

[Jua18a] Juan-Carlos Perez-Cortes, Alberto Perez, Sergio Saez-Barona, Jose-Luis Guardiola, Ismael Salvador Igual, and Sergio Sáez. A system for in-line 3d inspection without hidden surfaces. *Sensors*, 18:2993, 09 2018.

[Tob18a] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Domain randomization and generative models for robotic grasping. *IEEE International Conference on Intelligent Robots and Systems*, pages 3482–3489, 2018.

[Tom18a] Tomáš Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation *Computer Vision – ECCV*, 2018.

[Buk20a] Yannick Bukschat and Marcus Vetter. EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach *Preprint*, nov 2020.

[YiL20a] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. *International Journal of Computer Vision*, 128(3):657–678, mar 2020.

[YiL20a] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. *International Journal of Computer Vision*, 128(3):657–678, mar 2020.

[Tae20a] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.

[Che21a] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2021.

[Che21b] Chenhao Yang, Yuyi Liu, and Andreas Zell. Relative camera pose estimation using synthetic data with domain adaptation via cycle-consistent adversarial networks. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 102, 8 2021.

[ITI21a] ITI-Instituto Tecnologico de Informatica. Pose estimation *Kaggle Dataset*, 2021.

[Jac21a] Jack Langerman, Ziming Qiu, Gábor Sörös, Dávid Sebők, Yao Wang, and Howard Huang. Domain adaptation of networks for camera pose estimation: Learning camera pose estimation without pose labels *Preprint*, 2021.

[Oma21a] Omar Del-Tejo-Catala, Jose-Luis Guardiola, Javier Perez, David Millan Escriva, Alberto J. Perez, and Juan-Carlos Perez-Cortes. Probabilistic pose estimation from multiple hypotheses, *Preprint*, 2021.

[Sve21a] Svetozar Zarko Valtchev and Jianhong Wu. Domain randomization for neural network classification. *Journal of Big Data*, 8:94, 12 2021.

[Dmi22a] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[Sha22a] Sota Shoman, Tomohiro Mashita, Alexander Plopski, Photchara Ratsamee, and Yuki Uranishi. Real-to-synthetic feature transform for illumination invariant camera localization. *IEEE Computer Graphics and Applications*, 42:47–55, 2022.

[ITI23a] ITI-Instituto Tecnologico de Informatica. Pose estimation with domain adaptation *Kaggle Dataset*, 2023.

[Mar17a] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium *Advances in Neural Information Processing Systems*, 30, 2017.