Neural Inpainting for Domain Gap in Space

Jeff Park
Stanford University
tpark94@stanford.edu

Abstract

In spaceborne vision applications, domain gap between training and test must be overcome as one must train a neural network relying exclusively on synthetic images, which tend to have very different characteristics compared to spaceborne images. This project studies how neural inpainting, an unsupervised learning technique, may bridge the gap by forcing the network to learn the underlying representation of the spacecraft image regardless of the image domain. The preliminary results indicate the current implementation does not improve the network's performance on spaceborne images compared to the baseline, therefore we end with future directions to improve neural inpainting-based method and to better bridge the domain gap. The implementation is available in `https://github.com/tpark94/CS231AFinalProject.git`[1].

1. Introduction

Machine learning and artificial intelligence are attracting ever-increasing attention from the space community in various applications. Examples include 3-body problem of celestial bodies [2], satellite swarm control and maneuver, and spacecraft navigation about resident space objects such as another spacecraft [15] and asteroids [24]. Specifically, many researchers have focused on using a monocular camera as the main sensor for navigation due to its small mass and power requirements. The on-board capability of autonomously estimating the position and orientation (*e.g*. pose) of the main spacecraft with respect to the target could significantly aid future missions such as or-orbit servicing [18] and active debris removal [9].

Unlike images captured on Earth, spaceborne images are characterized by extreme lighting conditions, low signal-to-noise ratio, high contrast, and often symmetric shape of the target spacecraft. Naturally, previous works on pose stimation using classical image processing methods, such as edge and corner detections, have shown to fall short in terms of performance and robustnes [6, 23]. Such challenge has led researchers to instead apply machine learning for image-based spacecraft pose estimation. However, machine learning has its own difficulty in spaceborne applications; namely, it is practically impossible to acquire large-scale dataset of the interested target in relevant scenarios with accurate pose annotations. In response, Sharma and D'Amico [21] presented Spacecraft Pose Estimation (SPEED) [22], the first publicly available benchmark dataset of the Tango spacecraft from PRISMA mission [6, 7]. Specifically, it comprises 15,000 *synthetic* images rendered with OpenGL and 300 *real* images of a mockup model captured in a high-fidelity robotics facility. SPEED was used in the 2019 Satellite Pose Estimation Challenge (SPEC) co-hosted by the Space Rednezvous Laboratory (SLAB) of Stanford University and the Advanced Concepts Team (ACT) of the European Space Agency (ESA) [21, 10]. In the competition, all top contesters presented algorithms based on Convolutional Neural Networks (CNN) to perform pose estimation [15, 4] with remarkable pose accuracy.

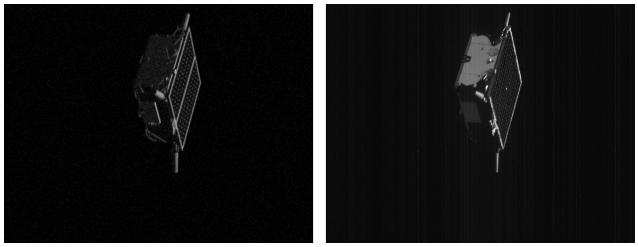

Figure 1. Images of the Tango spacecraft with approximately same pose and illumination. *Left*: Synthetic image from OpenGL. *Right*: Spaceborne image captured during the rendezvous phase of the PRISMA mission.

Despite the success, these CNN-based algorithms tend to fall short when tested on the spaceborne images from the actual missions. It happens because a CNN naturally overfits to the features exclusive to the synthetic imagery, as synthetic and spaceborne images have inherently different characteristics as shown in Figure 1. While a number of works have attempted to improve a CNN's performance

[1] CAs, please do **not** upload this report online.

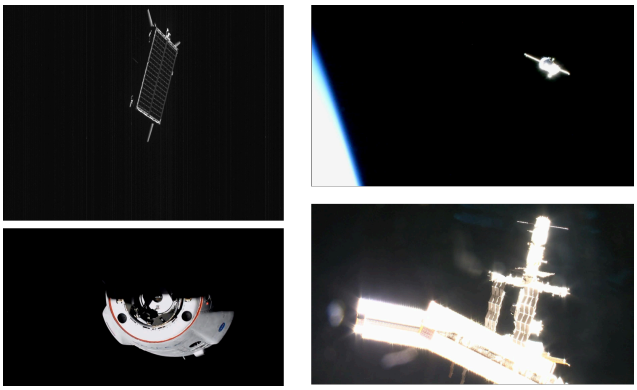

Figure 2. *Left*: Images with shadowing occlusion. *Right*: Images with glare occlusion.

by designing a better CNN architecture and pose extraction scheme [26] or more robust navigation algorithm [3], only a few addressed the specific issue of *domain gap* between training and application images for spaceborne missions.

In this project, we will investigate how representation learning could help bridge the domain gap *in the absence of* the target spaceborne images. The idea is to force a neural network to extract relevant features for pose estimation based on the spacecraft's geometry or other common representations instead of characteristics specific to the synthetic imagery. The representation learning technique we study is neural inpainting [16], which crops out random patches out of an image and trains an autoencoder to "fill in the gap." The motivation is that, on top of the differences in low-level features such as spacecraft textures, synthetic renderer often cannot faithfully replicate the extreme illumination conditions often seen in space. These conditions generally cause severe occlusion due to shadowing or extreme glares by direct sunlight (see Figure 2). Therefore, neural inpainting may help cope with these occlusions and thus the absence of visible keypoints.

Once the autoencoder is trained for neural inpainting, we take the trained encoder (dubbed *context encode* in [16]) as the feature extractor of our pose estimation network. We report the network's performances, both with and without using the pre-trained context encoder, on spaceborne images and show that unfortunately the current implementation does not improve the network's generalization capability.

2. Related Work

2.1. Domain Gap

The issue of *domain gap* arises when the training (*i.e.* source) and testing (*i.e.* target) data are drawn from different distributions. Domain adaptation [1] technique adjusts the labeled source dataset to the partially-labeled or unlabeled

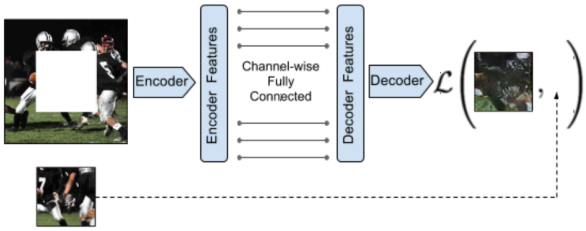

Figure 3. Visualization of neural inpainting. Figure from [16].

target dataset by studying the underlying common distribution. While it is well studied and equipped with theoretical bounds on generalization error, domain adaptation is not suited for spaceborne application since we do not have the target dataset for the relevant task (*e.g.* pose estimation of a specific spacecraft).

Therefore, some works [20, 17] have attempted to create datasets of photorealistic synthetic images of a spacecraft using 3D graphics renderer based on physically based rendering (PBR). Other works [15] studied domain randomization technique [25] by applying random style transfer to randomize the spacecraft texture, which is one of the distinguishing features between synthetic and spaceborne images.

2.2. Representation Learning

Representation learning, in an unsupervised setting without labels, forces a network to learn about the underlying representaiton of the dataset. Neural inpainting [16], or context encoder, is one example where the network must "paint" the patched-out area of the images, thereby learning the image context. Similar applications include jigsaw puzzle [14] and image colorization [11]. While these methods are restricted to specific learning tasks, other methods such as contrastive learning [5] uses a contrastive loss between two differently augmented data to learn more general visual representations.

3. Methods

In this project, we use a context encoder [16] trained for neural inpainting task as a pre-trained feature extractor of the pose estimation network. Such network is trained on synthetic images and tested on spaceborne images to gauge the generalization effect of the neural inpainting task.

3.1. Context Encoder

We first train a context encoder similar to [16], whose visualization is provided in Figure 3, by training an autoencoder to fill in the patched-out areas of the images. The encoder is the feature extractor of MobileNetV2 [19] up until

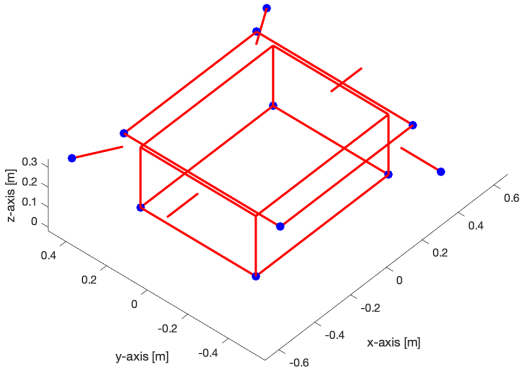

Figure 4. 11 keypoint locations on the Tango spacecraft to be detected by KRN.

the end of the bottleneck operations, *i.e.*, the encoder receives $224^2 \times 3$ and outputs $7^2 \times 320$ feature tensor. The decoder simply mirrors the architecture of the encoder.

3.2. Pose Estimation CNN

The pose estimation CNN is based on the keypoint regression network (KRN) in [15], which uses the same MobileNetV2-based freature extractor. The extracted features are then processed to output a $2N \times 1$ vector, whose elements correspond to the 2D coordinates of the locations of the spacecraft's pre-selected keypoints (Figure 4). These keypoints can be used along with the corresponding 3D locations in the model space to compute the 6D pose solution by solving the Perspective-n-Point (PnP) problem [12].

4. Experiments

4.1. Dataset

To train both autoencoder and KRN, 12,000 synthetic images of the Tango spacecraft are created. These images have the same characteristics as SPEED [21], but are rendered using the intrinsic properties of the camera actually used during the PRISMA mission. We also prepare the *clean* version of the dataset, which have exactly the same pose labels but without any spacecraft textures (Figure 5). The autoencoder will be trained to predict the clean images without any spacecraft textures, effectively forcing the network to disregard any visual features of the spacecraft (*e.g.*, solar panel patterns) that do not concern its geometric features.

For evaluation, 25 spaceborne images captured during the rendezvous phase of the PRISMA mission are used, as they are available with accurate labels and have been used for evaluation in other literatures [23].

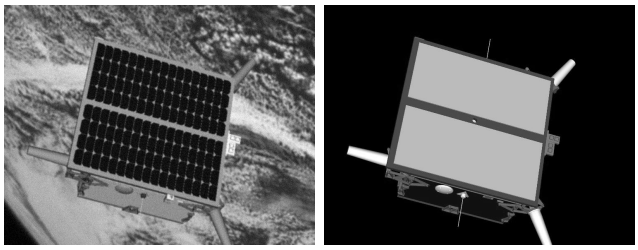

Figure 5. *Left*: Synthetic image. *Right*: Clean version of the synthetic image.

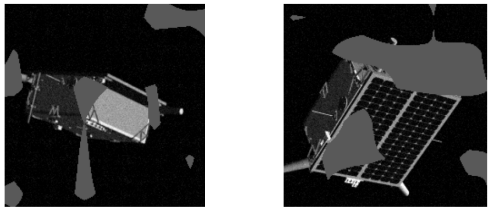

Figure 6. Random patches on two example images

4.2. Preprocessing

When training the autoencoder, we use the synthetic images with random background from the images of Pascal VOC 2012 dataset [8]. Various data augmentations are applied including color jittering, blurring and Gaussian noise. Lastly, the random patches are generated by creating random areas whose sum is no more than 30% of the entire image. Then, random uniform values are assigned to these patched-out regions, as occluded regions in spaceborne images often come in various intensities from dark (due to shadow) to bright (due to glare). Two of such examples are given in Figure 6.

For KRN, the same patched images are used for training as well. For both autoencoder and KRN, note that we first crop out a random square area around the spacecraft by using the ground-truth bounding box tightly fitting around the spacecraft. We take the tightest-fitting square area around the spacecraft, randomly enlarge the area by 20%, then randomly shift the spacecraft. The final area is cropped out before applying all data augmentation and random patches. The motivation is that there will be a separate object detection pipeline prior to pose estimation, which helps in pose estimation scenario where the spacecraft may be far away and thus only occupies a small portion of the image pixels.

4.3. Loss Functions

For autoencoder, we use the Smooth L1 loss defined as

$$\text{SmoothL1Loss} = \frac{1}{n} \sum_i z_i, \tag{1}$$

where

$$z_i = \begin{cases} \frac{1}{2}(x_i - y_i)^2, & |x_i - y_i| < 1 \\ |x_i - y_i| - \frac{1}{2}, & \text{otherwise} \end{cases} \quad (2)$$

which is equivalent to ℓ_2-loss if a pixel's intensity difference is less than 1 and ℓ_1-loss otherwise. While the original paper for context encoder [16] also suggests using an adversarial loss by employing a separate discriminator network, we do not pursue it in this project.

The loss for KRN outputs are simply the ℓ_2-loss between the predicted and ground-truth pixel locations of each keypoint.

4.4. Training

We train both autoencoder and KRN using the AdamW [13] optimizer with learning rate of 0.001. They are trained for 200 and 300 epochs, respectively. The project is implemented in PyTorch 1.7.

4.5. Evaluation

When evaluating the performance of KRN, we convert the keypoint locations into the orientation error (E_R) and translation error (E_T) defined as the following:

$$E_\mathrm{R} = 2 \arccos |\boldsymbol{q}_{\mathcal{BC}} \cdot \tilde{\boldsymbol{q}}_{\mathcal{BC}}| \quad (3)$$

$$E_\mathrm{T} = \|\tilde{\boldsymbol{t}}_{\mathcal{BC}} - \boldsymbol{t}_{\mathcal{BC}}\|_2, \quad (4)$$

where $\boldsymbol{q}_{\mathcal{BC}}$ is a unit quaternion aligning the spacecraft's body frame ($\mathcal{B}$) with the camera frame ($\mathcal{C}$), and $\boldsymbol{t}_{\mathcal{BC}}$ is the position vector from the origin of $\mathcal{B}$ to the origin of $\mathcal{C}$. These are obtained by running the OpenCV's EPnP algorithm [12] on a pair of 2D and 3D coordinates of the spacecraft's keypoints.

5. Results

Here we present the results of training the context encoder and the pose estimation network.

5.1. Context Encoder

We first present the result of training the autoencoder based on visualizations only. Figure 7 shows two examples of reconstructed images. We can observe that while the autoencoder can recover the patched-out areas, the result tends to be quite blurry and often cannot discern the exact shape (*e.g.*, bottom-right antennae of the second example). The reconstruction capability can be limited by the use of a single smooth ℓ_1 loss or the limited decoder capacity due to mirroring the MobileNet architecture.

We also note that the implementation in this project is not exactly the same as that of [16]. Whereas the autoencoder in [16] only recovers the patched-out areas, the autoencoder in this project is trained to recover the entire image, both existing and erased areas. Since this is more difficult than just focusing on the erased areas, following the original implementation may improve the reconstruction quality.

5.2. Pose Estimation

Next, Figure 8 reports the performance of KRN in different training and testing configurations. First, the baseline indicates when trained on synthetic (without random patches) and tested on spaceborne images, without employing a context encoder, there is a gap in performance compared to when tested on 3,000 synthetic images, which comes from the same distribution as the training images. The next scenario, when trained on synthetic images with random patches, resembles random erasing data augmentation technique by [27]. Unfortunately, the augmentation does not improve performance over domain gap with given configurations.

The last results indicate the scenario when the pre-trained context encoder is used as the feature extractor of KRN. When the encoder is frozen while training KRN, we see the pose estimation network cannot learn. This is regardless of the size of the network part following the encoder, possibly due to imperfect training result of the context encoder as shown in Figure 7 or because features extracted by the context encoder are insufficient to compute the 2D keypoint locations. Even when the entire KRN is trainable but initialized with the weights of the pre-trained context encoder, we do not see any noticeable improvement.

6. Conclusion

In this project, we have trained a context encoder by training it to fill in the gap created in the synthetic spacecraft images in an unsupervised setting. Then, the pre-trained encoder was used as a feature extractor of a pose estimation network which is tasked to predict the 2D locations of the keypoints pre-designated on the spacecraft model. Unfortunately, the pre-trained encoder was not successful in improving the domain gap posed by the synthetic and spaceborne images.

There are many ways to extend this work. One is to train the autoencoder better using adversarial loss or other means. Then, we can try different pose estimation architectures that may work better with the context encoder. For example, the network of [3] uses an autoencoder-like architecture to predict the heatmaps each corresponding to the keypoint locations. Since the network's task is essentially a reconstruction, architectures like this may be better compatible with the features learned by the context encoder.

Moreover, the fundamental limitation of neural inpainting is that the patching-out of random areas only simulate occlusion, but not other differences like the surface textures of synthetic and real spacecraft. This challenge may be overcome by using more general unsupervised learning

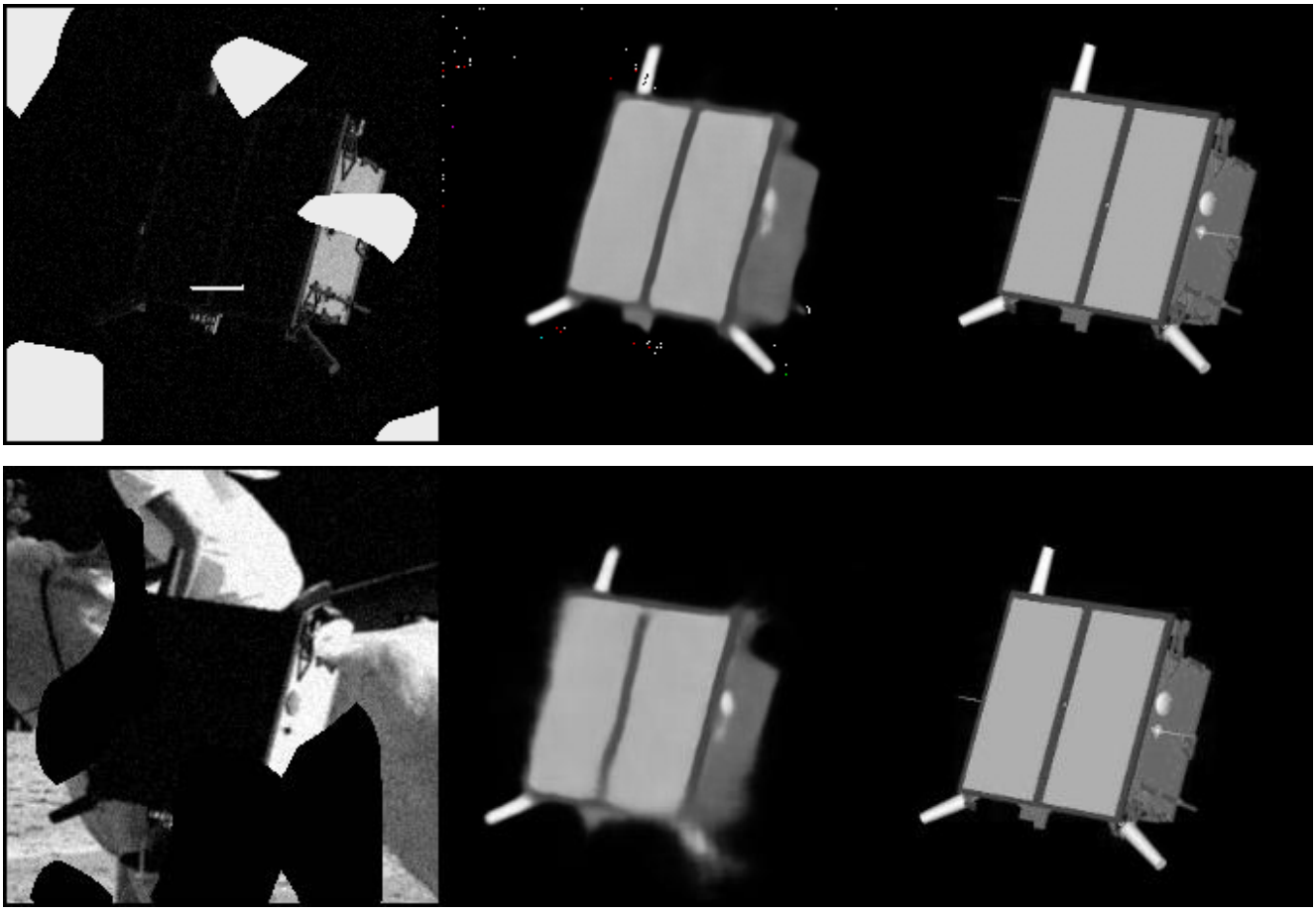

Figure 7. *Left*: Training image. *Middle*: Reconstructed image. *Right*: Ground-truth image.

	Train	Test	E_R [deg]	E_T [m]
Baseline	Synthetic	Synthetic	3.28	0.35
	Synthetic	Spaceborne	27.93	1.86
	Synthetic + Patches	Spaceborne	44.72	4.40
Pre-trained encoder, frozen	Synthetic + Patches	Spaceborne	151.98	45.94
Pre-trained encoder, entire network trainable	Synthetic + Patches	Spaceborne	55.03	3.65

Figure 8. Pose results of KRN.

techniques such as contrastive learning or using the neural inpainting in combination with domain randomization techniques.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

[2] P. G. Breen, C. N. Foley, T. Boekholt, and S. P. Zwart. Newton versus the machine: solving the chaotic three-body problem using deep neural networks. *Monthly Notices of the Royal Astronomical Society*, 494(2):2465–2470, 04 2020.

[3] L. P. Cassinis, R. Fonod, E. Gill, I. Ahrns, and J. G. Fernandez. *CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft*.

[4] B. Chen, J. Cao, Á. P. Bustos, and T.-J. Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2816–2824, 2019.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[6] S. D'Amico, M. Benn, and J. L. Jørgensen. Pose estimation of an uncooperative spacecraft from actual space imagery. *International Journal of Space Science and Engineering*, 2(2):171, 2014.

[7] S. D'Amico, P. Bodin, M. Delpech, and R. Noteborn. PRISMA. In M. D'Errico, editor, *Distributed Space Missions for Earth System Monitoring Space Technology Library*, volume 31, chapter 21, pages 599–637. 2013.

[8] 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.

[9] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm, S. Barraclough, A. Ratcliffe, C. Bernal, F. Chaumette, A. Pollini, and W. H. Steyn. RemoveDEBRIS: An in-orbit active debris removal demonstration mission. *Acta Astronautica*, 127:448–463, 2016.

[10] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico. Satellite pose estimation challenge: Dataset, competition design and results. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–1, 2020.

[11] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2008.

[13] I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[14] M. Noroozi, e. B. Favaro, Paolo", J. Matas, N. Sebe, and M. Welling. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016.

[15] T. H. Park, S. Sharma, and S. D'Amico. Towards robust learning-based pose estimation of noncooperative spacecraft. In *2019 AAS/AIAA Astrodynamics Specialist Conference, Portland, Maine*, August 11-15 2019.

[16] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders: Feature learning by inpainting. 2016.

[17] P. F. Proenca and Y. Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. *arXiv preprint arXiv:1907.04298*, 2019.

[18] B. B. Reed, R. C. Smith, B. J. Naasz, J. F. Pellegrino, and C. E. Bacon. The Restore-L servicing mission. *AIAA Space 2016*, 2016.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[20] C. Schubert, K. Black, D. Fonseka, A. Dhir, J. Deutsch, N. Dhamani, G. Martin, and M. Akella. A pipeline for vision-based on-orbit proximity operations using deep learning and synthetic imagery, 2021.

[21] S. Sharma and S. D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using neural networks. In *2019 AAS/AIAA Space Flight Mechanics Meeting, Ka'anapali, Maui, HI*, January 13-17 2019.

[22] S. Sharma, T. H. Park, and S. D'Amico. Spacecraft pose estimation dataset (speed). Stanford Digital Repository. Available at: `https://doi.org/10.25740/dz692fn7184`, 2019.

[23] S. Sharma, J. Ventura, and S. D'Amico. Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets*, page 1–16, 2018.

[24] N. Stacey and S. D'Amico. Autonomous swarming for simultaneous navigation and asteroid characterization. In *2018 AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT*, August 19-23 2018.

[25] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. 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.

[26] J. Xu, B. Song, X. Yang, and X. Nan. An improved deep keypoint detection network for space targets pose estimation. *Remote Sensing*, 12(23), 2020.

[27] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.