Training Dense Object Nets While Not Training Dense Object Nets

A Preprint

Kanishk Navale, Ralf Gulde, Marc Tuscher
Sereact,
Stuttgart, Germany
firstname.lastname@sereact.ai

March 22, 2023

Abstract

We propose a framework to train Dense Object Nets (DON) with no intent to train DON; instead, we mine the dense visual object descriptors produced by DON while training another network unrelated to creating dense visual object descriptors. The dense visual object descriptors from the DON are object view-invariant, configuring and generalising the object's geometrical structure. However, an object image pair is required to train DON with the corresponding mapping, and recent research developments proves that the DON is as better as the number of correspondence supplied to it while training. The computation costs increase as the number of image-pair correspondences increases with the descriptor dimension, limiting one to produce descriptors of less dense dimension. Our framework does not require any image-pair correspondence mapping. It yields denser visual descriptors while consuming lesser computation resources while not compromising the robustness of the dense visual object descriptors compared to DON.

Keywords Dense Object Nets · Second keyword · More

1 Introduction

As of this writing, the ideal object representation for robot grasping and manipulation tasks is yet unknown. The existing representations may not be the best for tackling more complex tasks as they lack actual object information belonging to the same class and configuration (shape, color and size). In industrial robot-based automation, the objects are specifically coded for their visual features using 2D and 3D vision systems. The downside of this lies in the fact that the robot has to be taught to pick every other part with its visual representation. This process comes with the tedious schedule of teaching the robot to pick every part irrespective of the part's configuration, and viewpoint. The solution lies in using artificial intelligence (AI) equipped robots. A deep learning neural network (DNN) is based on artificial neurons capable to learning a task and is good as the task related data it is trained on. The data used to train DNN is often expensive as it requires engineered features that DNN can predict or regress. SIFT [1], SURF [2] and ORB [3] produce dense local descriptors of an object in an image and serve as target features to train DNN to yield object representation for robot grasping furthermore, these features computed by [1, 2, 3] come with its own inert limitations and cannot generalize objects well. Our interests of work is on reducing efforts to develop hand engineered features to train DNN and developing DNN that can generalize plathora of objects such that we spend less time teaching robot how to tend objects in realtime.

In 2018, Florence et al. [4] introduced a novel visual object representation to the robotics community, terming it "dense visual object descriptors". DON, an aritificial intelligence framework proposed by Florence et al. [4] produces dense visual object descriptors. In detail, the DON converts every pixel in the image ($I[u, v] \in \mathbb{R}^3$) to a higher dimensional embedding ($I_D[u, v] \in \mathbb{R}^D$) such that $D \in \mathbb{N}^+$ which are nothing but dense local descriptors of that pixel respective to the image. The dense visual object descriptor generalize an object up to a certain extent and have been recently applied to rope manipulation [5], block manipulation [6], robot control [7], fabric manipulation [8] and robot grasp pose estimation [9, 10]. Suwajanakorn et al. [11] propose self-supervised geometrically consistent keypoints, exploring the idea of optimizing a representation based on a sparse collection of keypoints or landmarks, but without access to keypoint annotations. The authors of [11] devise an end-to-end geometric reasoning framework first introduced by [12] to regresses a set of geometrically consistent keypoints coined as KeypointNet. This means that KeypointNet is capable of generalizing objects without the need of hand engineered features. Suwajanakorn et al. [11] show that using two unique objective loss functions, namely, a relative pose estimation loss and a multi-view consistency goal, uncovers the consistent keypoints across multiple views and object instances. Their affine translation-equivariant design may extend to previously unknown object instances trained on ShapeNet [13] dataset.

At first, we present modifications to the DNN inspired from [4] and [11] such that we seemlessly train and mine object representations composed of object generalizing dense local descriptors while training for KeypointNet task. Second, we develop synthetic dataset using [14] to train the DNN and prove that the mined dense local descriptors from our framework is as robust as dense visual object descriptors produced from DON while consuming less computation resources. Additionally, we demonstrate an self-supervised framework to train DON with semantically equivalent objects which is not previously demonstrated in [4, 15, 9, 10, 16, 17] to train DON.

2 Related Work

We are solely interested in computing dense visual object descriptors of an object. The DON training strategy in [4] relies on the depth information for computing correspondences in an image pair using camera intrinsics and pose information [18]. However, when employing consumer-grade depth cameras for capturing the depth information, the depth cameras capture noisy depth in cases of tiny, reflecting objects, which are common in industrial environments. In the meantime, Kupcsik et al. [9] used Laplacian Eigenmaps [19] to embed a 3D object model into an optimally generated embedding space acting as an target to train DON in a supervised fashion. The optimal embeddings brings in more domain knowledge by associating 3D object model to images views. Kupcsik et al. [9] efficiently apply it to smaller, texture-less and reflective objects by eliminating the need of the depth information. Kupcsik et al. [9] further compare training strategies for producing 6D grasps for industrial objects and show that a unique supervised training approach increases pick-and-place resilience in industry-relevant tasks.

Florence [15] has found that the pixelwise contrastive loss function used to train DON might not perform well if a computed correspondence is spatially inconsistent (analogously to the case of noisy depth information). This further highlights that the precision of contrastive-trained models can be sensitive to the relative weighting between positive-negative sampled pixels. Instead, the Florence [15] introduces a new continuous sampling-based loss function called "Pixelwise Distribution Loss". The pixelwise distribution loss is much more effective as it is a smooth continuous pixel space sampling method compared to the discrete pixel space sampling method based on pixelwise contrastive loss. The pixelwise distribution loss regresses a set of probability distribution heatmaps aiming to minimize the divergence between the predicted heatmap and the ground truth heatmap mitigating errors in correspondences. Futhermore, the pixelwise distribution loss does not need non-matching correspondences compared to the the pixelwise contrastive loss. Differently, Hadjivelichkov and Kanoulas [16] extends the DON training using semantic correspondences between objects in multi-object or cluttered scenes overcoming the limitations of [18, 19]. The authors, Hadjivelichkov and Kanoulas [16] employ offline unsupervised clustering based on confidence in object similarities to generate hard and soft correspondence labels. The computed hard and soft labels lead DON in learning class-aware dense object descriptors, introducing hard and soft margin constraints in the proposed pixelwise contrastive loss to train DON. Further eliminating the need for camera pose and intrinsic information along with depth information to compute correspondences in an image pair, Yen-Chen et al. [17] used NeRF [20] to train DON. The NeRF [20] recreates a 3D scene from a sequence of images captured by the smartphone camera. The correspondences are extracted from the synthetically reconstructed scene to train DON. Recently, based on SIMCLR inspired frameworks [21, 22], Adrian et al. [10] introduced similar architecture and another novel loss function called "Pixelwise NT-Xent loss" to train DON more robustly. The pixelwise ntxent loss consumes synthetic correspondences independent of depth cameras computed from image augmentations to train DON. Adrian et al.'s experiments show that the novel loss function is invariant with respect to the batch size. Additionally adopted "$PCK@k$" metric has been adopted as in preceedings [23, 24] to evaluate and benchmark DON on cluttered scenes previously not benchmarked.

In the proposed framework we do not use any loss functions in [4, 15, 9, 10, 16, 17] to train DON however we adopt the network architecture from [4] and train on the task of the "KeypointNet"[11] with adaption of the loss functions proposed in [11, 25].

3 Methodology

3.1 DNN Framework and Mining Methodology

As a backbone, we employ ResNet-34 architecture [26]. We preserve the last dense feature layer in the ResNet-34 DNN and remove the last flatten feature in the forward method and linear layer that is popularly used for image classification tasks. The backbone downsamples the RGB image $I_{RGB} \in \mathbb{R}^{H \times W \times 3}$ to dense features $I_d \in \mathbb{R}^{h \times w \times D}$ such that $h \ll H, w \ll W$ and $D \in \mathbb{N}^+$. We directly upsample the dense features from the identity layer as illustrated in the Figure 1 in page 3 as follows:

$$f_U : I \in \mathbb{R}^{h \times w \times D} \rightarrow I_D \in \mathbb{R}^{H \times W \times D}, \tag{1}$$

the upsampled dense features substitutes as dense visual local descriptors produced from the DON. Similarly as in [11], we stack spatial-probability regressing layer and depth regressing layer on top of the identity layer to predict $N \in \mathbb{N}^+$ number of keypoint's spatial-probability as follows:

$$f_S : I_d \in \mathbb{R}^{h \times w \times D} \rightarrow I_s^N \in \mathbb{R}^{h \times w \times N} \text{ , such that } \sum^{h} \sum^{w} I_s^N = 1.0^N, \tag{2}$$

and depth as follows:

$$f_D : I_d \in \mathbb{R}^{h \times w \times D} \rightarrow I_{\hat{d}} \in \mathbb{R}^{h \times w \times N}. \tag{3}$$

We incorporate continuous sampling method f_E from [15, 11] to convert the upsampled predicted spatial-probability and depth of a keypoint to spatial-depth expectation as follows:

$$f_E \circ g_E : [I_s, I_{\hat{d}}] \rightarrow [u, v, d]^T \in \mathbb{R}^3 \text{ , where } g_E : I \in \mathbb{R}^{h \times w \times N} \rightarrow I \in \mathbb{R}^{H \times W \times N}. \tag{4}$$

Furthermore, we train the DNN in a twin architecture fashion as depicted in the Figure 2 in page 3 as proposed in [21, 22, 4, 15, 9, 10, 16, 17] on the KeypointNet task.

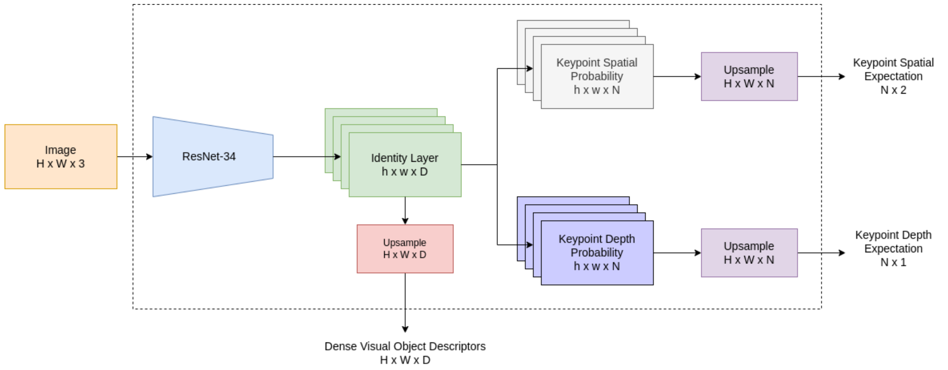

Figure 1: Illustration of novel DNN architecture designed to efficiently compute and seamlessly extract dense visual object descriptors. During inference we extract dense visual object descriptors directly from the network and ignore predicted spatial-depth expectation of the keypoints.

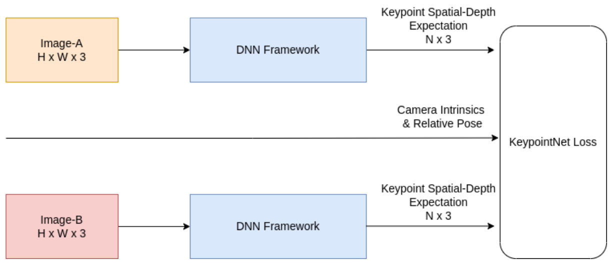

Figure 2: Depiction of twin DNN architecture's training strategy.

3.2 Loss Function Modifications

For training, we directly adopt silhoutte consistency loss ($\mathcal{L}_{obj}$), variance loss ($\mathcal{L}_{var}$) and separation loss ($\mathcal{L}_{sep}$) functions from [11] to train the network on the keypoint prediction task. However, we modify the multi-view consistent loss and relative pose estimation loss. In the case of multi-view consistency loss we project the predicted spatial-depth expectation using camera intrinsics as follows:

$$X_{cam} \in \mathbb{R}^{3\times 1} = \mathcal{I}_{cam}^{-1}\, [u, v, 1.0]^T \otimes d \text{ , where } \mathcal{I}_{cam} \in \mathbb{R}^{3\times 3} \text{ and } u, v, d \in \mathbb{R}^{+}. \tag{5}$$

Furthermore, we project the camera coordinates of the keypoints regressed on both images to the world coordinates using camera transformation and compute Huber Loss [27] represented as $\mathcal{H}$ in Equation 6 as multi-view consistency loss as follows:

$$\mathcal{L}_{mvc} \in \mathbb{R} = \mathcal{H}(\mathcal{T}^{A}_{C\to W}\hat{X}^{A}_{cam}, \mathcal{T}^{B}_{C\to W}\hat{X}^{B}_{cam}) \text{ , where } \mathcal{T}_{C\to W} \in SE(3) \text{ and } \hat{X}_{cam} = [X_{cam}, 1.0]^T \in \mathbb{R}^{4\times 1}, \tag{6}$$

this modification is geometrically more intuitive as all the keypoints projected from different camera viewpoints into world coordinates occupy the same value addtionally, using Huber Loss creates smoother gradients to optimize the DNN compared to the original implementation of Euclidean distance. In Equation 6 $SE(3) \in \mathbb{R}^{4\times 4}$ is a "Special Euclidean Group" [28]. We do not discard the relative transformation information to calculate the realative pose loss as suggested in [11] and being influenced from [25] we modified the relative pose loss as follows:

$$\mathcal{L}_{pose} = \|log(\mathcal{T}^{\dagger}_{truth}\mathcal{T}_{pred})\| \text{ , where } log : SE(3) \to \mathfrak{se}(3) \text{ and } \mathcal{T}^{\dagger} = \begin{bmatrix} R^T & -R^T t \\ 0^T & 1 \end{bmatrix} \in SE(3). \tag{7}$$

3.3 Controlled Dataset Engineering

We have chosen the cap object for creating synthetic dataset as the cap mesh models are readily available in the "Shapenet" library [13] as it possess rich object information including textures. Blenderproc [14] is used to generate the synthetic cap dataset by using of the 10 number of cap models from [13] library. Futhermore, the caps are chosen such that each of them have distinct shapes, designs and colors. For this controlled dataset, 100 random cameras are added in the environment with random poses capturing depth, camera extrinsics information (referring to $\mathcal{T}_{C\to W}$ in Equation 6) and object mask for each viewpoint for each cap model. To make the training more robust such that networks are more object-centric, the images are additionally augmented with random backgrounds and noisy backgrounds as depicted in Figure 3 in page 4 in addition to the color jitter and greyscale augmentations. For color jitter and greyscale augmentation we use available "Torchvision" [29] library.

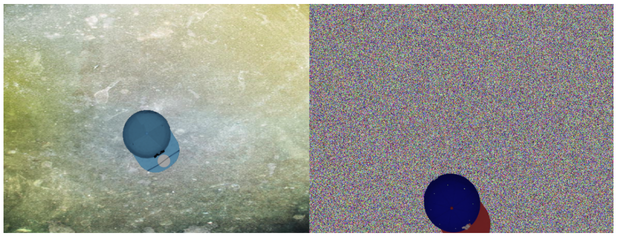

Figure 3: The image in the right illustrates the noisy background augmentation and the image in left depicts random background augmentation.

4 Results and Benchmarking

We implemented out training and benchmarking using "PyTorch-Lightning"[30] and "PyTorch"[31] libraries. Futhermore, we employ "ADAM"[32] optimizer to optimize our model for 1000 epochs with learning rate of $\alpha = 10^{-3}, \beta_1 = 0.9$ and $\beta_2 = 0.999$ with weight decay $\eta = 10^{-6}$. Addtionally, we reduce the learning rate by a factor of 0.9 every 2500 opimtization steps. We set the all the loss weights to 1.0 except variance loss weight to $w_{var} = 10^{-4}$ and mean reduce our batch-wise losses. The proposed novel DNN model is benchamarked against standard DON model for computed dense visual object descriptors and computation resources used where batch size plays a major influence. For training standard DON model, we import training settings, evaluation metrics and the method of generating synthetic correspondences as in [10][1].

[1] GitHub link to our implementation of generating synthetic correspondences: https://github.com/KanishkNavale/Mapping-Synthetic-Correspondences-in-an-Image-Pair

References

[1] D. G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. “Speeded-up robust features (SURF)”. In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[4] P. R. Florence, L. Manuelli, and R. Tedrake. “Dense object nets: Learning dense visual object descriptors by and for robotic manipulation”. In: *arXiv preprint arXiv:1806.08756* (2018).

[5] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg. “Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data”. In: *CoRR* abs/2003.01835 (2020). arXiv: `2003.01835`.

[6] C.-Y. Chai, K.-F. Hsu, and S.-L. Tsao. “Multi-step Pick-and-Place Tasks Using Object-centric Dense Correspondences”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 4004–4011.

[7] P. Florence, L. Manuelli, and R. Tedrake. “Self-supervised correspondence in visuomotor policy learning”. In: *IEEE Robotics and Automation Letters* 5.2 (2019), pp. 492–499.

[8] A. Ganapathi et al. “Learning Dense Visual Correspondences in Simulation to Smooth and Fold Real Fabrics”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 11515–11522.

[9] A. Kupcsik, M. Spies, A. Klein, M. Todescato, N. Waniek, P. Schillinger, and M. Bürger. “Supervised Training of Dense Object Nets using Optimal Descriptors for Industrial Robotic Applications”. In: *arXiv preprint arXiv:2102.08096* (2021).

[10] D. B. Adrian, A. G. Kupcsik, M. Spies, and H. Neumann. “Efficient and Robust Training of Dense Object Nets for Multi-Object Robot Manipulation”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 1562–1568.

[11] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi. “Discovery of latent 3d keypoints via end-to-end geometric reasoning”. In: *Advances in neural information processing systems* 31 (2018).

[12] S. Levine, C. Finn, T. Darrell, and P. Abbeel. “End-to-end training of deep visuomotor policies”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[13] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* (2015).

[14] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam. “Blenderproc”. In: *arXiv preprint arXiv:1911.01911* (2019).

[15] P. R. Florence. “Dense visual learning for robot manipulation”. PhD thesis. Massachusetts Institute of Technology, 2020.

[16] D. Hadjivelichkov and D. Kanoulas. “Fully Self-Supervised Class Awareness in Dense Object Descriptors”. In: *5th Annual Conference on Robot Learning*. 2021.

[17] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola. *NeRF-Supervision: Learning Dense Object Descriptors from Neural Radiance Fields*. 2022.

[18] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[19] M. Belkin and P. Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation”. In: *Neural computation* 15.6 (2003), pp. 1373–1396.

[20] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. “Nerf: Representing scenes as neural radiance fields for view synthesis”. In: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A simple framework for contrastive learning of visual representations”. In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[22] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. “Barlow twins: Self-supervised learning via redundancy reduction”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 12310–12320.

[23] C.-Y. Chai, K.-F. Hsu, and S.-L. Tsao. “Multi-step pick-and-place tasks using object-centric dense correspondences”. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 4004–4011.

[24] M. E. Fathy, Q.-H. Tran, M. Z. Zia, P. Vernaza, and M. Chandraker. “Hierarchical metric learning and matching for 2d and 3d geometric correspondences”. In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, pp. 803–819.

[25] W. Zhao, S. Zhang, Z. Guan, W. Zhao, J. Peng, and J. Fan. "Learning deep network for detecting 3d object keypoints and 6d poses". In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2020, pp. 14134–14142.

[26] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: (2016), pp. 770–778.

[27] P. J. Huber. "Robust estimation of a location parameter". In: *Breakthroughs in statistics: Methodology and distribution* (1992), pp. 492–518.

[28] W. P. Thurston. "Three-Dimensional Geometry and Topology, Volume 1". In: *Three-Dimensional Geometry and Topology, Volume 1*. Princeton university press, 2014.

[29] S. Marcel and Y. Rodriguez. "Torchvision the machine-vision package of torch". In: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 1485–1488.

[30] W. A. Falcon. "Pytorch lightning". In: *GitHub* 3 (2019).

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[32] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).