Implementation of Latent 3D Keypoints via End-to-end Geometric Reasoning

Deepan Chakravarthi P, Kishaan Jeeveswaran, Swaroop Bhandary K
Masters in Autonomous systems
Hochschule Bonn-Rhein-Sieg
Bonn, Germany
deepan.padmanabhan@smail.inf.h-brs.de

Abstract

This study focuses on the implementation of KeypointNet, provided in the paper, 'Discovery of Latent 3D Keypoints via End-to-end Geometric Reasoning'. KeypointNet incorporates two deep networks, namely- keypoint and orientation network, to extract optimal 3D keypoints for a particular category. The keypoints which can be used for any downstream tasks are optimized in KeypointNet. The optimal set of keypoints between different viewing angles and category-specific object instances are obtained using a differentiable objective with five loss function components. The geometrically and semantically consistent keypoints are estimated in an unsupervised manner with no ground-truth keypoint annotations. The base work investigates the performance of the KeypointNet framework on chairs, cars, and planes category of the ShapeNet core dataset using an implementation in Tensorflow 1.14. This study concentrates on re-implementing the various components of the loss function, the two network architectures, and working of the model in Tensorflow 2.1 in order to understand the inner working of the keypoint network. The project integrates studying the performance of the implementation for the airplane category of the ShapeNet core dataset. In addition, a qualitative analysis of the results is provided. The code is available at https://github.com/swaroop1904/keypointnet.

Index Terms

keypoints detection, ShapeNet, geometric reasoning, unsupervised learning, pose estimation.

I. Introduction

Keypoints are the points of interest in an image. Keypoint detection is an imperative part in many computer vision tasks such as pose estimation, face detection, 3D reconstruction, simultaneous localization and mapping (SLAM), structure from motion (SfM), and object detection. The critical requirement of a good keypoint extraction method is to provide geometric and photometric invariance. The former assumes an invariance to image translation, rotation and scale while the latter should assure an invariance to changes in brightness, contrast and color [5].

The solution to many computer vision applications such as 3D reconstruction [6] and shape alignment [7] include a stand-alone keypoint detection method followed by a geometric reasoning framework. However, the conventional solution to the problem of 3D keypoint detection is a supervised method involving manual keypoint annotations of an object category or deploying model-based fitting methods. The limitations of other approaches include substantial expense in terms of manual effort for annotation. In addition, both manual annotations and model-based keypoint detection are associated with an error component [1]. The selection of optimal set of keypoints depends on the downstream task the keypoints will be used for. Therefore, taking into account the downstream task in selecting the keypoints with desirable properties such as distinctiveness, diversity, ease of detection, robust to noise, transformation invariant, etc. is important [1].

On the other hand, end-to-end learning of feature representations has led to advances in image classification [2], modelling images by generative networks [3], and developing game play agents that outperform humans [4]. The paper, 'Discovery of Latent 3d Keypoints via End-to-end Geometric Reasoning' [1], popularly called as KeypointNet, suggests an end-to-end geometric reasoning framework that has the ability to learn optimal set of 3D keypoints for various downstream tasks.

In this study, the paper KeypointNet is discussed with an implementation in Tensorflow 2.1 from scratch. This paper provides a summary of the KeypointNet paper in the section II, followed by experiments to illustrate the geometric and semantic consistency of the implemented model.

II. Overview of KeypointNet

This section summarizes the imperative aspects of the KeypointNet paper.

A. Main Idea

KeypointNet estimates a set of 3D keypoints defined as pixel coordinates and associated depth values given a single image of a known object category. The keypoints are predicted in such a manner that they are geometrically and semantically consistent across different viewing angles and instances of an object category as illustrated in the figure1.

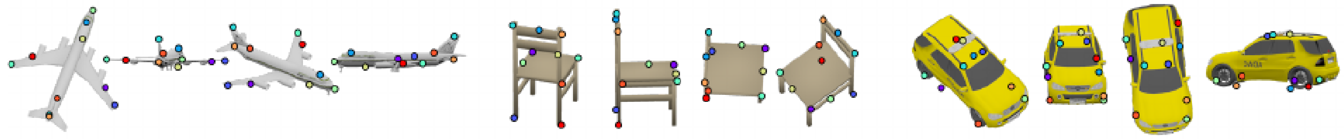


Fig. 1. Illustration of geometric and semantic consistency across viewing angles and object instances. Keypoints of the same color correspond to the same geometric and semantic locations in an object category. For instance, the red keypoint of chairs is consistently predicted on the the right leg of the chair even in entirely occluded conditions at different viewing angles. Image taken from [1].

B. Novelty of the paper

The paper explains an approach to estimate the optimal keypoints for any downstream task by incorporating end-to-end learning as illustrated in figure 2. This approach is novel on the basis that the keypoints are estimated in an unsupervised manner without any explicit keypoint annotations. The keypoints illustrate properties that are more suitable to solve the downstream task. The end-to-end geometric reasoning aids in identifying keypoints with useful properties suitable for the downstream task of pose estimation. In addition, the model was trained solely on synthetic data rather than real images.

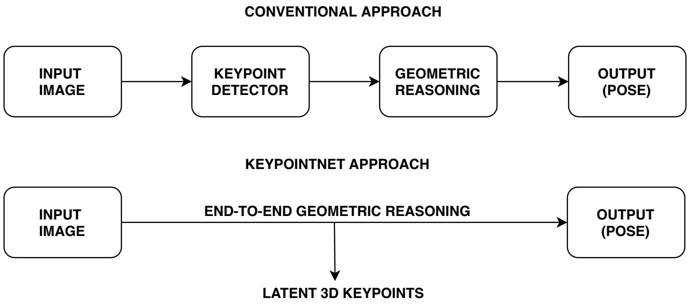


Fig. 2. Difference between the approach followed in Keypointnet and conventional geometric reasoning applications for the task of pose estimation.

The authors do not define a keypoint position a priori and learn a supervised mapping of keypoints between images.

C. Architecture

KeypointNet is a CNN based model which takes an image along with the global orientation of the object in it as input and predicts a predefined number of 3D keypoints on the image plane. The orientation helps the model by providing a dominant direction of the object, thus making the model more resilient when predicting the keypoints for objects which are locally symmetric in a particular axis. Locally symmetric objects are challenging as the keypoints prediction task becomes difficult without knowing the global direction the object is facing toward. The global orientation information aids the learning ability of the model to predict consistent keypoints across different views. The authors came up with two separate models, one to predict the orientation of the object in the image (OrientNet) and the other model to predict the keypoints given the image and the orientation (KeypointNet).

Both OrientNet and KeypointNet are CNN based models with 13 layers of 3 × 3 filters with dilation rates of 1, 1, 2, 4, 8, 16, 1, 2, 4, 8, 16, 1, 1. Dilated convolutional layers are used to preserve the image dimensionality without compromising on the effective receptive field. KeypointNet has 64 filters in every layer, while OrientNet has half the number of filters in each layer. The number of filters in the last layer of the KeypointNet is 2 × the number of keypoints being predicted. All the layers except the last layer uses ReLU activation function and batch normalization.

The KeypointNet predicts a probability distribution rather than directly regressing over the keypoint locations. This helps making the network equivariant to translation. The expected values of these distributions are used as the predicted keypoint locations. This probability distribution map $g_i(u, v)$ is obtained by applying a spatial softmax. Similarly, the depth information is estimated in a similar manner. The expected value of the spatial coordinates and depth value provides the 3D keypoint positions.

D. Training and Prediction

The OrientNet model is trained using mean squared loss between orientation and ground truth with learning rate of $1e^{-4}$ and batch size of 32. The KeypointNet model is trained using primarily relative pose estimation loss and multi-view consistency loss [1]. In addition to these losses, the authors suggest three more loss components to improve the performance of the network. The losses are described below:

1) Multi-view consistency: This loss measures the disagreement between the two set of points under the ground truth transformation. The loss encourages consistent keypoint detection across 3D transformations of an object. This loss function makes sure that the position of the keypoints are consistent.
2) Relative pose estimation loss: This loss penalizes the angular difference between the ground truth rotation R versus the rotation $\hat{R}$ recovered from P_1 and P_2 using orthogonal procrustes as shown in the figure 3 where P_1 and P_2 are the set of 3D keypoints predicted for the two images.
3) Separation loss : This loss penalizes keypoints that are closer than a specified limit δ. This makes sure that the predicted keypoints do not degenerate and converge to a single location in both the images.
4) Silhouette loss: The probability distribution predicted by the model is multiplied with the binary mask of the object to decide which keypoint locations might lie outside the object. The loss penalizes any keypoint predicted outside the object boundary. This loss ensures all the keypoints lie within the boundaries of the objects. The loss is calculated as described in the equation 1.

$$L_{sill} = \frac{1}{N} \sum_{i=1}^{N} -log \sum_{u,v} b(u,v) g_i(u,v) \tag{1}$$

$b(u, v) \in \{0, 1\}$ is the binary segmentation mask of the object instance in the training pairs. $g_i(u, v)$ is the spatial probability distribution map.

5) Variance loss: This loss is used to minimize the output probability distribution variance. The variance loss aids in obtaining a single peaky distribution and in turn achieving a mean within the silhouette of the object instance.

The weights of multi-view consistency, relative pose estimation, separation loss, silhouette loss and variance loss are set to 1, 1, 1, 0.2, 0.5.

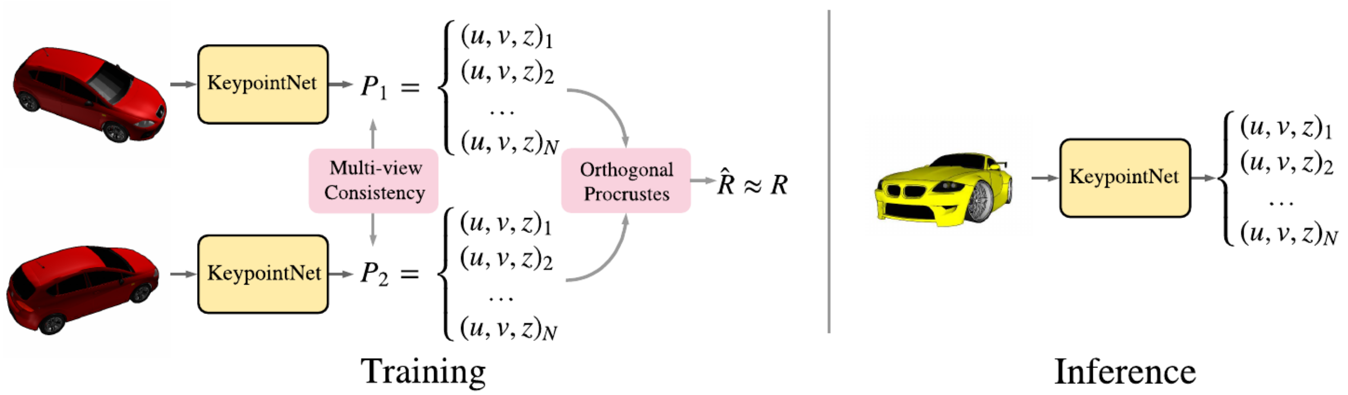
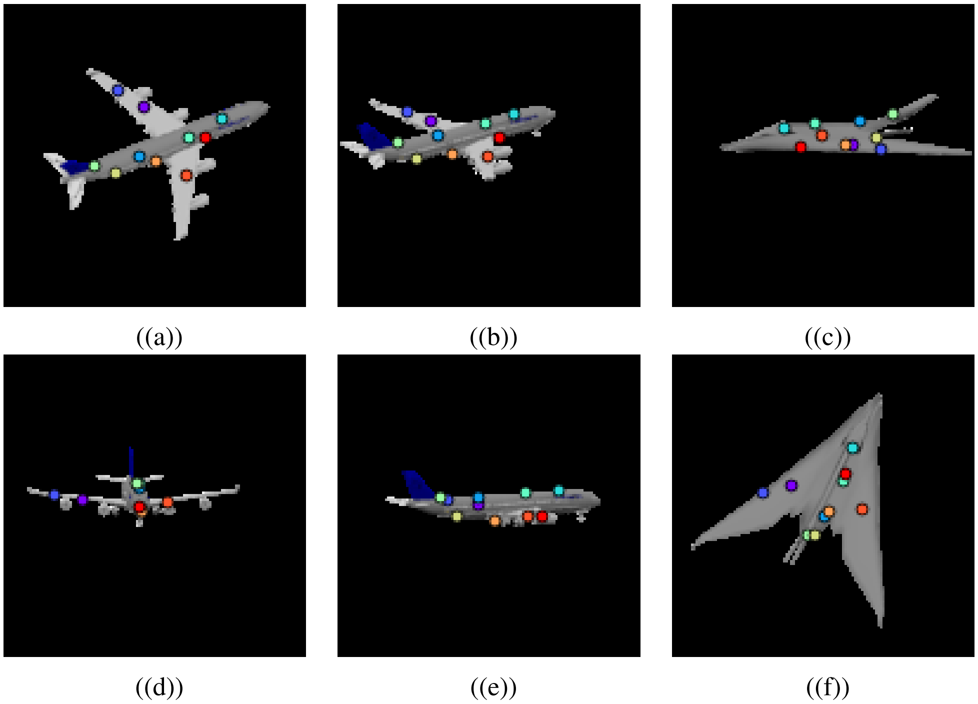

KeypointNet
$P_1 = \{(u,v,z)_1, (u,v,z)_2, \dots, (u,v,z)_N$
Multi-view Consistency
Orthogonal Procrustes
$\hat{R} \approx R$
KeypointNet
$P_2 = \{(u,v,z)_1, (u,v,z)_2, \dots, (u,v,z)_N$
Training
KeypointNet
$(u,v,z)_1, (u,v,z)_2, \dots, (u,v,z)_N$
Inference

Fig. 3. Training and inference methodology followed in the KeypointNet approach. The training illustrate the determination of optimal set of keypoints P_1 and P_2 each with 'N' keypoints preserving multi-view consistency for different viewing angles. The rotational difference between the keypoints on the different viewing angles are taken into consideration for calculating the relative pose estimation loss. A single 128x128 image of an object instance is used to predict the 'N' number of keypoints. Image taken from [1].

III. Results and Discussion

A. *Evaluation of the category - Planes*

We have evaluated the model for the class of airplanes from the ShapeNet dataset. Below are the results and observations.

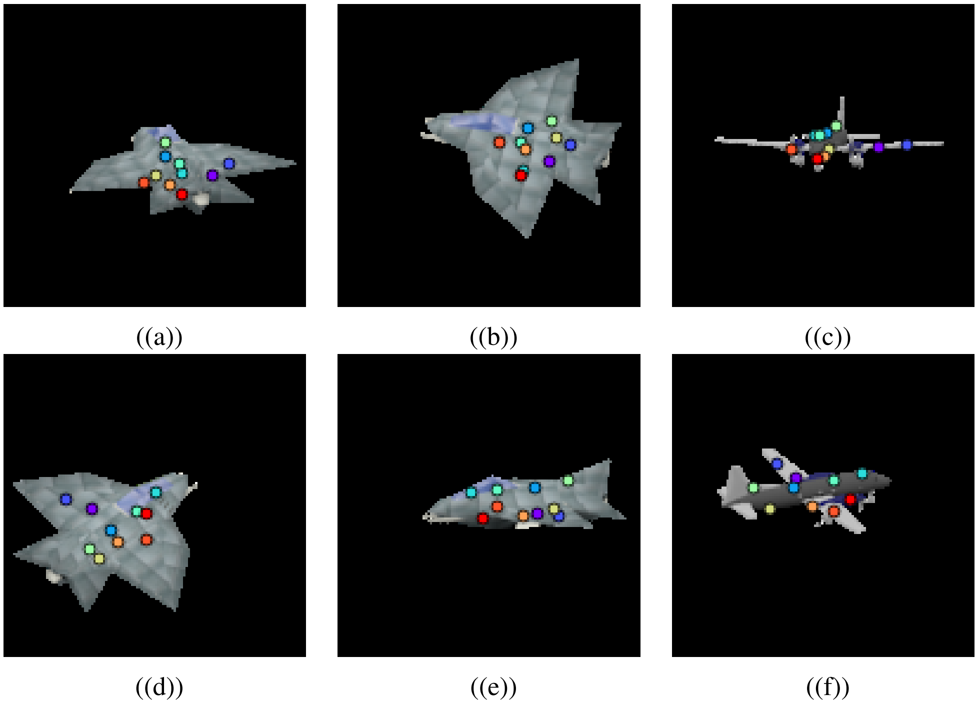


((a)) ((b)) ((c))

((d)) ((e)) ((f))

Fig. 4. Working examples for the object class of airplanes in the ShapeNet dataset using our implementation of KeypointNet model.

We can see in Fig 4 that the KeypointNet model is able to consistently able to predict keypoints even with out-of-plane rotations. Despite a high intraclass variance in the shape of the objects, the model is able to predict useful and consistent keypoints. For example, the purple and blue keypoints are consistently predicted on the left wing of the planes.

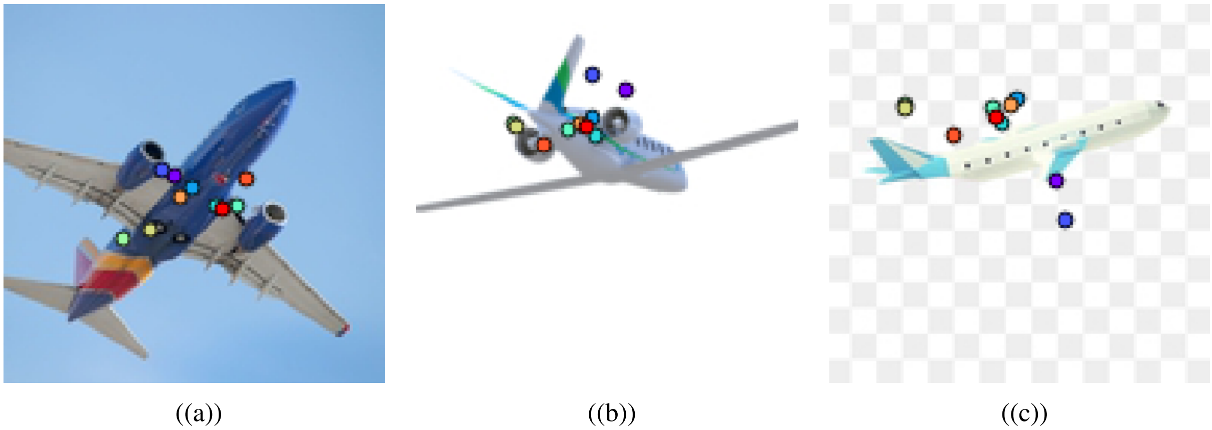


((a)) ((b)) ((c))

((d)) ((e)) ((f))

Fig. 5. Failing cases for the object class of airplanes in the ShapeNet dataset using our implementation of KeypointNet model.

As can be seen in Fig 5, the predictions in the first row are incorrect and the predictions in the second row are correct and have been added here for the purpose of comparison. For the first example, the keypoints predicted are not consistent with the correct predictions for the other viewpoints in the same object. One possible explanation for this is lack of geometric shape features in the object. Having a distinct geometric shape makes the task of predicting keypoints easier for the model when compared to a plain object with less texture based features in it.



((a)) ((b)) ((c))

Fig. 6. Prediction of the model on new images scrapped from the internet[1].

In addition to images in the dataset, we tested the model on some images which we scrapped from the internet. In Fig 6 the predictions of our model on such images can be seen.

IV. Future work

This work primarily focuses on the implementation of the KeypointNet in Tensorflow 2.1. Though we could see many working examples and some failure cases, a thorough study on what properties of the object leads to easier keypoint detection and lack of which features leads to bad predictions needs to be done.

One of the research possibilities is studying how training the network on ShapeNet images with random background still generalize for real world object instance images. In addition, one potential way to overcome the failed cases with real images object instance in a random background is by studying the recent advances in domain adaptation methods or by training directly on real image pairs with relative pose labels.

The implemented framework incorporates only keypoint detector. However, the keypoint detector and descriptor can also be integrated by a post-processing task or by an end-to-end optimization approach.

Acknowledgment

We thank Prof. Dr. Paul G Plöger, Dr. Matias Valdenegro- Toro and M.Sc. Luis Octavio Arriaga Camargo who gave the opportunity to work on this challenging task and get our hand dirty by implementing the project. We express our gratitude to Matias and Octavio, who patiently supported us during the entire course of this project work. Finally, we would like to extend our gratitude to our family and friends, whose love and moral support helped us in steering this project at the right pace.

References

[1] Suwajanakorn, Supasorn, Noah Snavely, Jonathan J. Tompson, and Mohammad Norouzi. ”Discovery of latent 3d keypoints via end-to-end geometric reasoning.” In Advances in Neural Information Processing Systems, pp. 2059-2070. 2018.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2012.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2014.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 2015.

[5] Bojanić, David, Kristijan Bartol, Tomislav Pribanić, Tomislav Petković, Yago Diez Donoso, and Joaquim Salvi Mas. ”On the Comparison of Classic and Deep Keypoint Detector and Descriptor Methods.” In 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), pp. 64-69. IEEE, 2019.

[6] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. ACM transactions on graphics (TOG), 2006.

[7] Yan Li, Leon Gu, and Takeo Kanade. A robust shape model for multi-view car alignment. CVPR, 2009.

[1] These images have been taken from https://www.theverge.com/2018/4/17/17249990/southwest-airlines-engine-explosion-passenger-partially-ejected-depressurization, https://dlpng.com/png/1171128, https://pngtree.com/free-png-vectors/air-plane.