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

Advanced Machine Learning

April 23, 2020


**Team Members**
Deepan Chakravarthi Padmanabhan
Kishaan Jeeveswaran
Swaroop Bhandary K

# Table of Contents

# Introduction

# Introduction

- Keypoints: Points of interest in an image.
- Geometric and semantic invariance.
- Applications: Pose estimation, 3D reconstruction, Simultaneous Localization and Mapping (SLAM).

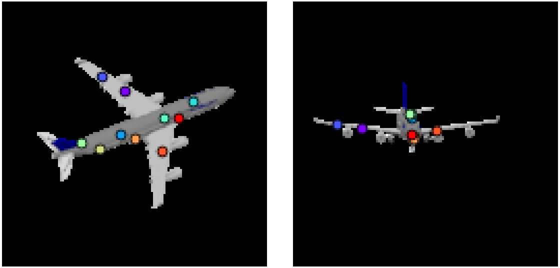
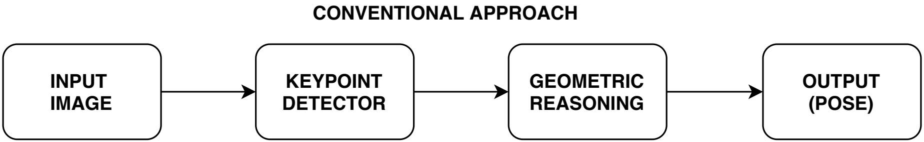


Figure 1: Geometrically and semantically consistent keypoints across viewpoints.

# Problem statement

- Current state of the art approaches are supervised requiring numerous annotated keypoint data [1].

- The prior works include a stand-alone keypoint detector and geometric reasoning framework [1] [2].



# Problem statement

- Current state of the art approaches are supervised requiring numerous annotated keypoint data [1].
- The prior works include a stand-alone keypoint detector and geometric reasoning framework [1] [2].

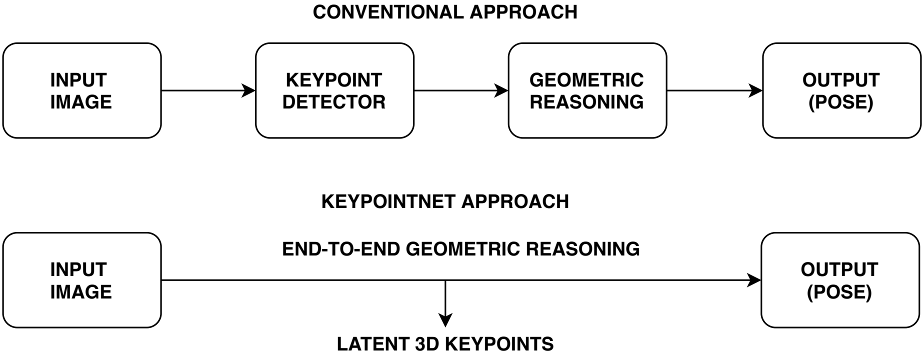



Figure 2: Problem statement and solution by Keypointnet approach.

# Motivation

- Useful for various downstream tasks such as pose estimation and object detection.
- Explicit annotation of keypoints are required for training the current state of the art networks, which is laborious.

# Motivation

- Useful for various downstream tasks such as pose estimation and object detection.
- Explicit annotation of keypoints are required for training the current state of the art networks, which is laborious.
- The KeypointNet approach learns from synthetic data significantly reducing the cost of collecting real data.
- KeypointNet paper illustrates consistent keypoints for various object categories in different view points.

# Related work

# Related work

- 3D human keypoint detection from monocular RGB images: 3D structural priors [3], 2D-to-3D lifting [4], depth images [5].
- Convolutional Neural Networks (CNN) to predict correspondence between different objects of the same class using 3D models [6].
- Landmark localization by attribute prediction and equivariant landmark prediction [7].

# Overview of KeypointNet

# KeypointNet - Main idea

- Input: A single image of known object.
- Output: Specified number of 3D keypoints - (x, y) spatial position and depth values.

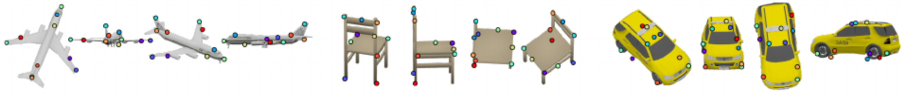


Figure 3: Illustration of geometric and semantic consistency across viewing angles and object instances [1]

# KeypointNet - Architecture

- 2 CNN based deep neural networks.
- 13 layers, 3 x 3 filters with different dilation rates [1, 1, 2, 4, 8, 16, 1, 2, 4, 8, 16, 1, 1].
- 64 filters for Keypointnet and 32 filters for OrientNet.
- OrientNet:
  - Predicts the global orientation of the object instance.
- KeypointNet:
  - Predicts the keypoints given the image and orientation of the object instance in the image.

## Training details

- $P_1$ and $P_2$ are two set of points predicted by keypointnet on the training pairs.
- $R$ and $\hat{R}$ are ground-truth and rotation estimated from the prediction respectively.

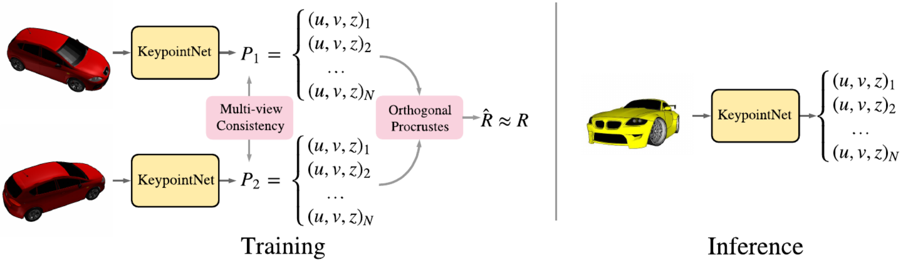
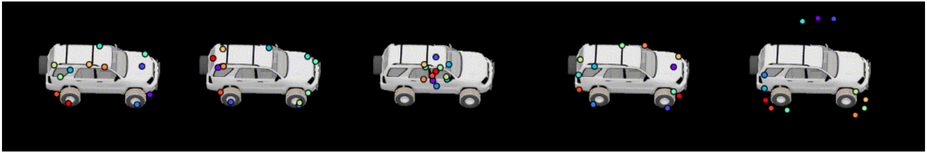



Figure 4: Training and inference methodology followed in the KeypointNet approach [1].

# KeypointNet - Loss functions

- Multi-view consistency: Disagreement between $P_1$ and $P_2$ in grou nd-truth $R$.

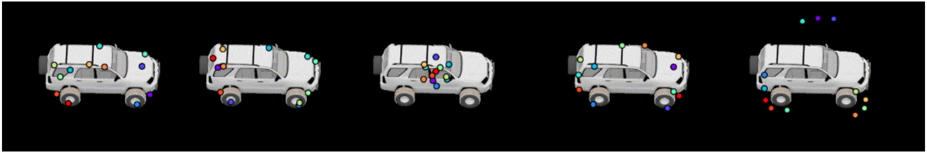


Baseline

No multi-view consistency

# KeypointNet - Loss functions

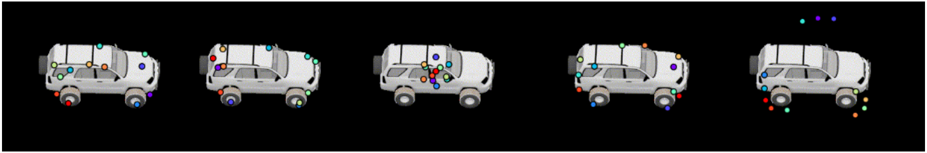
- Multi-view consistency: Disagreement between $P_1$ and $P_2$ in grou nd-truth $R$.
- Relative pose estimation: Penalize angular difference between $R$ and $\hat{R}$ between $P_1$ and $P_2$.



Baseline | No multi-view consistency | No pose estimation

# KeypointNet - Loss functions

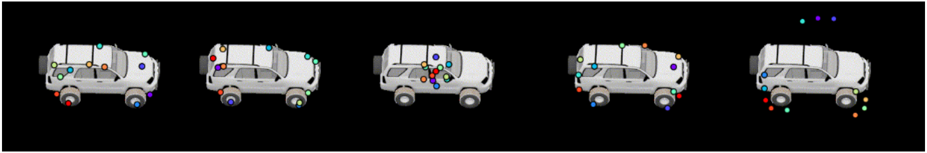
- Multi-view consistency: Disagreement between $P_1$ and $P_2$ in grou nd-truth $R$.
- Relative pose estimation: Penalize angular difference between $R$ and $\hat{R}$ between $P_1$ and $P_2$.
- Separation: Penalizes keypoints closer than a specified distance limit $\delta$.



Baseline | No multi-view consistency | No pose estimation | More noise in pose loss

# KeypointNet - Loss functions

- Multi-view consistency: Disagreement between $P_1$ and $P_2$ in grou nd-truth $R$.
- Relative pose estimation: Penalize angular difference between $R$ and $\hat{R}$ between $P_1$ and $P_2$.
- Separation: Penalizes keypoints closer than a specified distance limit $\delta$.
- Silhouette: Penalizes any keypoint predicted outside the object.

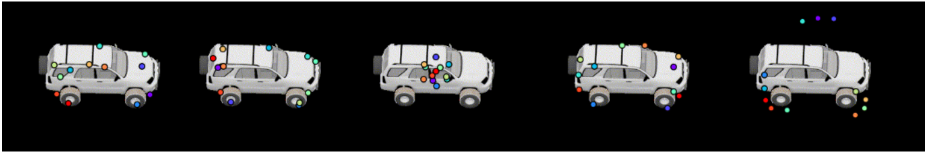


| Baseline | No multi-view consistency | No pose estimation | More noise in pose loss | No silhouette |
|---|---|---|---|---|

Figure 5: Ablation study output without a particular loss function [8].

# KeypointNet - Loss functions

- Multi-view consistency: Disagreement between $P_1$ and $P_2$ in grou nd-truth $R$.
- Relative pose estimation: Penalize angular difference between $R$ and $\hat{R}$ between $P_1$ and $P_2$.
- Separation: Penalizes keypoints closer than a specified distance limit $\delta$.
- Silhouette: Penalizes any keypoint predicted outside the object.
- Variance: Loss to minimize the variance in the output probability distribution.



Baseline | No multi-view consistency | No pose estimation | More noise in pose loss | No silhouette

Figure 5: Ablation study output without a particular loss function [8].

# Experimental setup

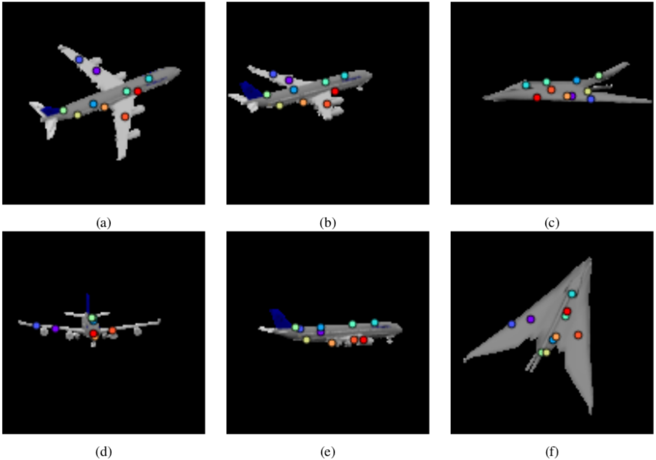
# Experimental setup

- Dataset:
  - ShapeNet core.
  - Only one object instance per image.
- Object category: Cars and Planes.
- Training details:
  - Batch size: 16
  - Learning rate: $10^{-4}$
  - Number of steps: 700K (Plane) and 600K (Car)
- Non-rigid deformation of objects: Blender

# Results

# Planes

- Planes - Working Example



(a) (b) (c) (d) (e) (f)

Figure 6: Working examples for the object class of airplanes in the ShapeNet dataset using our implementation of KeypointNet model.

# Planes

- Planes - Failing Cases

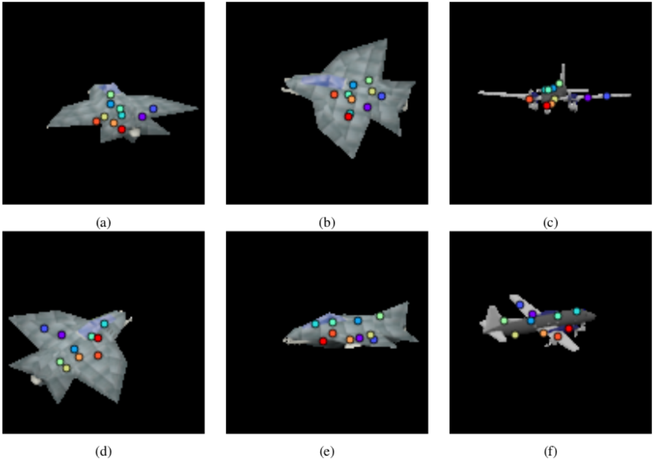
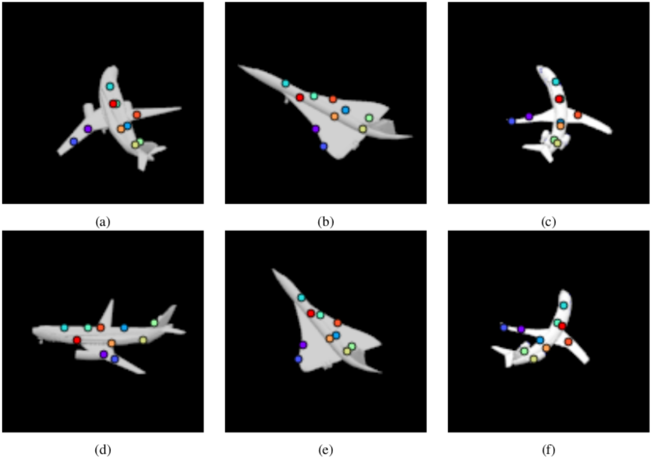


Figure 7: Failing cases for the object class of airplanes in the ShapeNet dataset using our implementation of KeypointNet model.

# Planes

- Planes - Deformed Examples



(a) (b) (c) (d) (e) (f)

Figure 8: Working examples for deformed plane objects from the dataset.

# Planes

- Planes - Real Examples

(a) (b) (c)

Figure 9: Prediction of the model on new images scrapped from the internet[1].

[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.

# Cars

- Cars - Working Examples

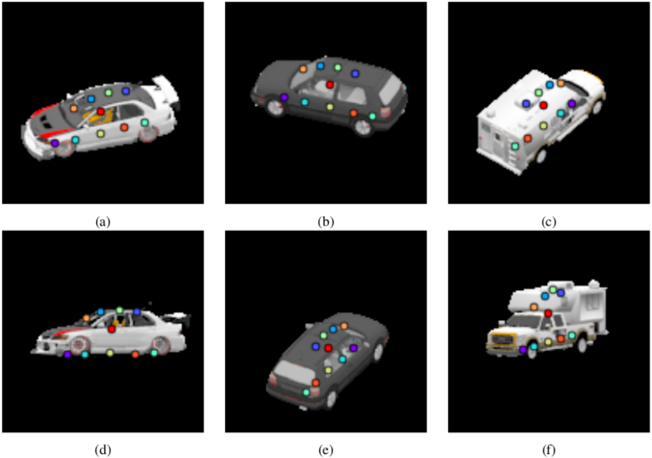



Figure 10: Working examples for the object class of cars in the ShapeNet dataset using our implementation of KeypointNet model.

# Cars

- Cars - Failing Cases

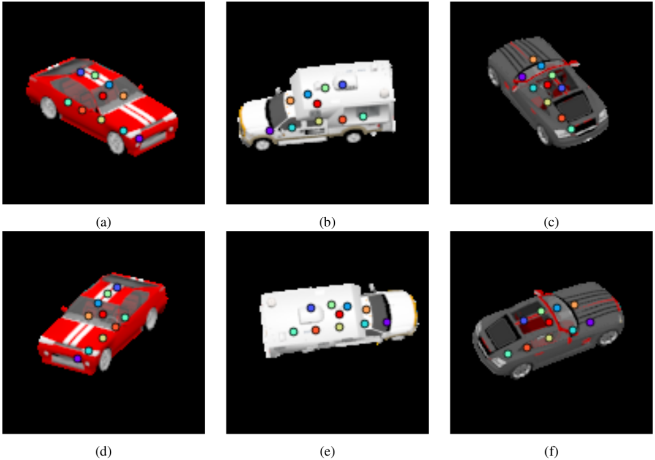


Figure 11: Failing cases for the object class of cars in the ShapeNet dataset using our implementation of KeypointNet model.

# Conclusion

# Contributions

- Re-implemented the KeypointNet base paper in TensorFlow 2.1.
- Evaluated the implementation on two object categories - planes and cars.
- Evaluated the model on monocular RGB real images, while the network is trained on synthetic data.
- The implemented model has been evaluated on non-rigidly deformed objects to test the robustness of the network.

# Challenges and Lessons learned

- Challenges:
  - Figuring out the values for the extra KeypointNet loss hyperparameters (e.g. the threshold distance for separation loss).
  - Understanding the conventions for the transformations in OpenGL.
  - Visual inspection and evaluation of the correctness of 3D keypoints in a 2D image.
- Lessons learned:
  - Novel loss functions could lead to significant increase in performance and save manual labor.
  - Callback functions to monitor individual losses while optimizing multi-loss objective functions could save much time.

# Future work

- A study on how training the network on rendered images of 3D objects generalize for real world object images.
- Look into domain adaptation methods or training with real image pairs with relative pose labels to overcome the failures in keypoint prediction on real images.
- Incorporate keypoint descriptor along with the detector implemented.

# References I


[1] Supasorn Suwajanakorn et al. "Discovery of latent 3d keypoints via end-to-end geometric reasoning". In: Advances in Neural Information Processing Systems. 2018, pp. 2059–2070.

[2] Yan Li, Leon Gu, and Takeo Kanade. "A robust shape model for multi-view car alignment". In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE. 2009, pp. 2466–2473.

[3] Dushyant Mehta et al. "Vnect: Real-time 3d human pose estimation with a single rgb camera". In: ACM Transactions on Graphics (TOG) 36.4 (2017), pp. 1–14.

[4] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. "Reconstructing 3d human pose from 2d image landmarks". In: European conference on computer vision. Springer. 2012, pp. 573–586.

[5] Gerard Pons-Moll et al. "Metric regression forests for correspondence estimation". In: International Journal of Computer Vision 113.3 (2015), pp. 163–175.

[6] Tinghui Zhou et al. "Learning Dense Correspondence via 3D-guided Cycle Consistency". In: CoRR abs/1604.05383 (2016). arXiv: 1604.05383. URL: http://arxiv.org/abs/1604.05383.

[7] Sina Honari et al. "Improving landmark localization with semi-supervised learning". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 1546–1555.

[8] Supasorn Suwajanakorn et al. Discovery of latent 3d keypoints via end-to-end geometric reasoning. URL: https://keypointnet.github.io/.

Thank you for your time!