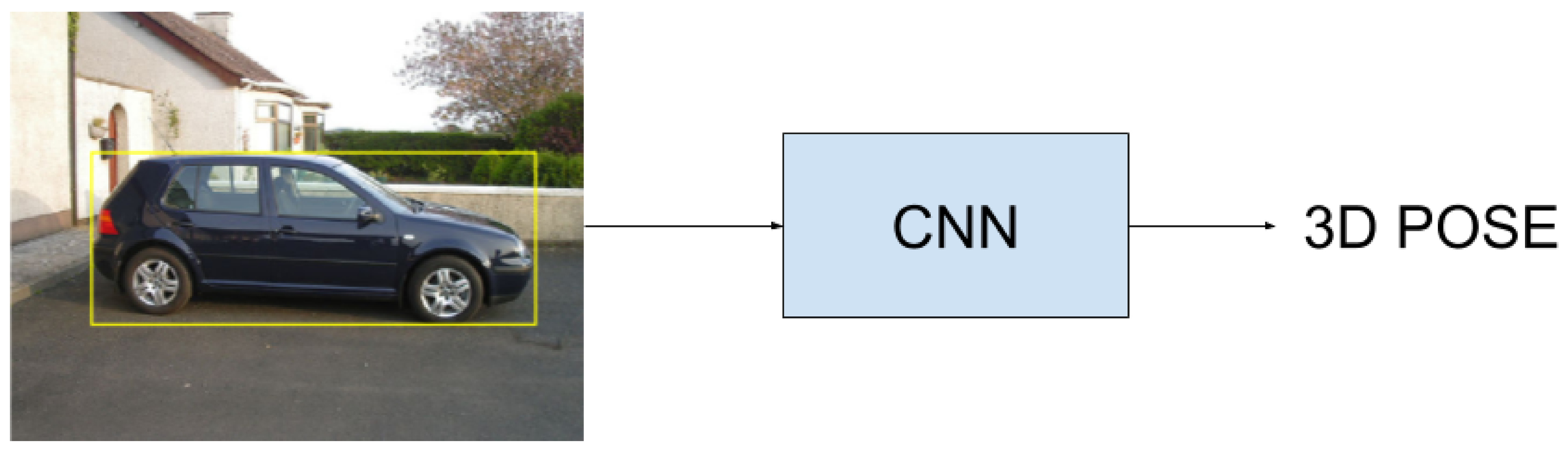
3D Pose Regression using Convolutional Neural Networks

Siddharth Mahendran   Haider Ali   René Vidal
Center for Imaging Science, Johns Hopkins University

Motivation and Problem Statement

Motivation: 3D pose estimation is a key component of challenging vision problems like scene understanding and autonomous navigation



Problem statement: *Given a 2D image and a bounding box around an object in the image, estimate the 3D rotation R between the object and the camera*

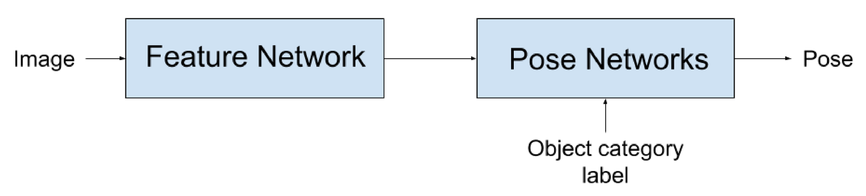
Introduction and Related Work

Prior work discretizes the pose space into key poses and treats the pose estimation problem as a classification problem

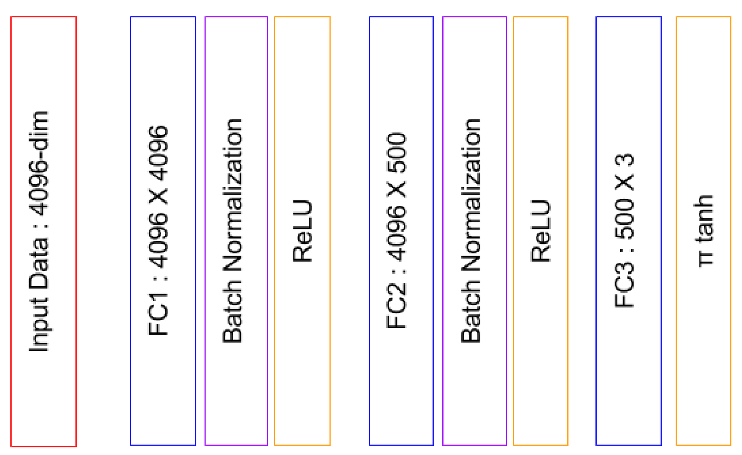
	V&K [2]	Render-for-CNN [1]	Ours
Problem formulation	Classification	Fine-grained classification	Regression
Representation	Discretized angles (21 bins)	Discretized angles (360 bins)	Axis-angle
Loss function	Cross-entropy	Weighted cross-entropy	Geodesic loss
Data augmentation	2D jittering	Rendered images	3D pose jittering + rendered images
Network architecture	VGG-Net (FC7)	AlexNet (FC7)	VGG-M (FC6)

Contribution: Instead of breaking up pose space into discrete key poses, we propose a regression formulation using representations (Axis-angle and Quaternion), loss functions (Geodesic loss between rotation matrices) and data augmentation techniques (3D pose jittering) that respect and exploit the non-Euclidean structure of the space of rotations.

Network Architecture



Overall Network



Pose Network

Representing 3D Rotations

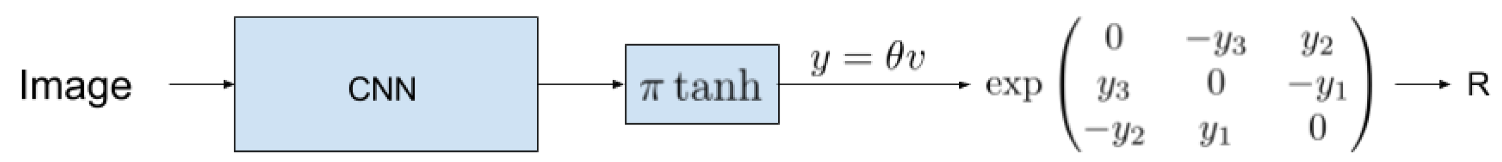
Rotation matrices lie in the Special Orthogonal group:

$$SO(3) := \{R \in \mathbb{R}^{3\times 3} : R^T R = I_3, \det(R) = 1\} \quad (1)$$

Geodesic loss function on the space of rotation matrices:

$$\mathcal{L}(R_1, R_2) = \frac{\| \log R_1 R_2^T \|_F}{\sqrt{2}} \quad (2)$$

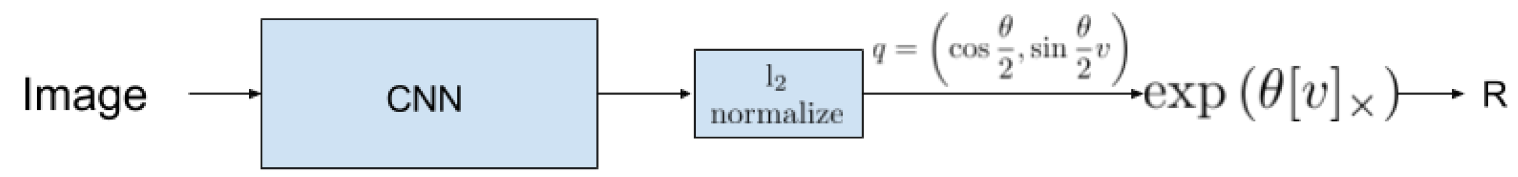
Axis-angle



Geodesic loss between ground-truth and predicted rotations:

$$\mathcal{L}(R, \hat{R}) = |\cos^{-1}\left[\frac{1}{2}(\mathrm{trace}(R^T\hat{R}) - 1)\right]| \quad (3)$$

Quaternion



Geodesic loss between ground-truth and predicted quaternions:

$$\mathcal{L}(q, \hat{q}) = \cos^{-1} |\langle q, \hat{q} \rangle| \quad (4)$$

3D Pose Jittering

For every image, 3D pose annotations of azimuth az, elevation el and camera-tilt ct, give 3D rotation $R(az, el, ct) = R_Z(ct)R_X(el)R(az)$. We perturb around ground-truth 3D pose using these transformations:

- Flips: $R(-az, el, -ct)$
- In-plane rotations: $R(az, el, ct \pm \delta ct)$
- Out-of-plane rotations: $R(az \pm \delta az, el, ct)$

Original   Flipped $\delta ct = +10°$ $\delta az = +10°$

[1] H. Su, C. Qi, Y. Li, and L. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. *ICCV*, 2015.

[2] S. Tulsiani and J. Malik. Viewpoints and keypoints. *CVPR*, 2015.

[3] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3d object detection in the wild. *WACV*, 2014.

Experiments

- Dataset: Pascal3D+[3] consists of ImageNet and Pascal VOC2012 images with 3D pose annotations. ImageNet trainval, VOC2012-train, and VOC2012-val images are used as training, validation, and testing data respectively.
- Training: Two step learning procedure: (i) Train the pose networks (with feature network fixed) using augmented and rendered data, and (ii) Finetune the overall network using original and flipped images

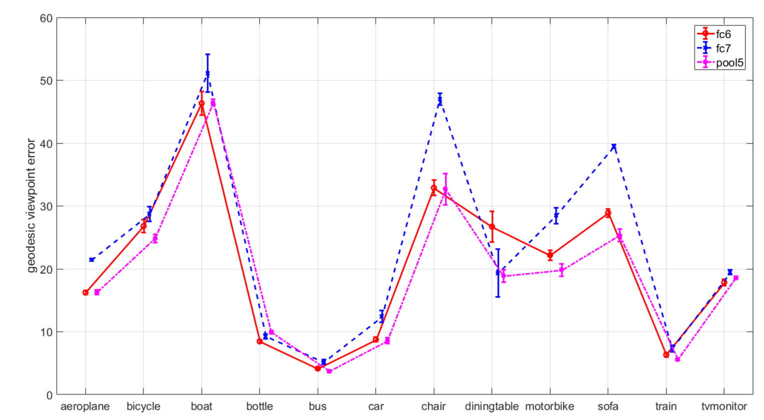
Expt.	aero	bike	boat	bottle	bus	car	chair	dtable	mbike	sofa	train	tv	Mean
V&K [2]	**13.80**	17.70	**21.30**	12.90	5.80	9.10	14.80	15.20	**14.70**	13.70	8.70	15.40	13.59
Render [1]	15.40	**14.80**	25.60	9.30	**3.60**	**6.00**	**9.70**	**10.80**	16.70	**9.50**	**6.10**	**12.60**	**11.67**
Axis-angle	13.97	21.07	35.52	**8.99**	4.08	7.56	21.18	17.74	17.87	12.70	8.22	15.68	15.38
Quaternion	14.53	22.55	35.78	9.29	4.28	8.06	19.11	30.62	18.80	13.22	7.32	16.01	16.63
Detected	14.71	21.31	45.07	9.47	4.20	8.93	26.36	20.70	19.16	18.80	8.72	15.65	17.76

Median geodesic viewpoint error (in degrees) using ground-truth bounding boxes for un-occluded and un-truncated objects

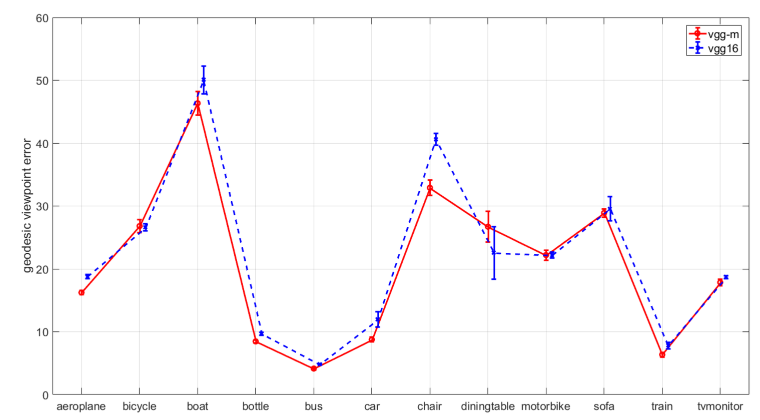
Expt.	aero	bike	boat	bottle	bus	car	chair	dtable	mbike	sofa	train	tv	Mean
[2]-ARP	64.0	53.2	21.0	-	69.3	55.1	24.6	16.9	54.0	42.5	59.4	51.2	46.5
Ours-ARP	61.95	49.07	20.02	35.18	66.24	49.89	19.78	15.36	49.38	40.92	56.68	49.87	42.86

Average Rotation Precision ($\Delta(R, \hat{R}) < 30°$ and $I/U > 0.5$)

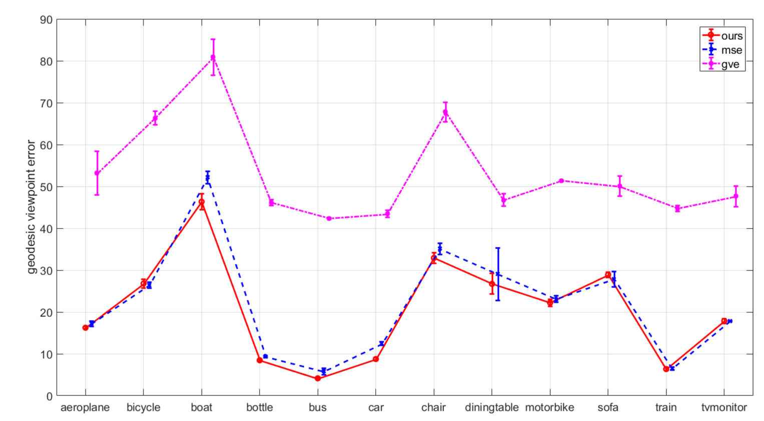
Ablation Analysis



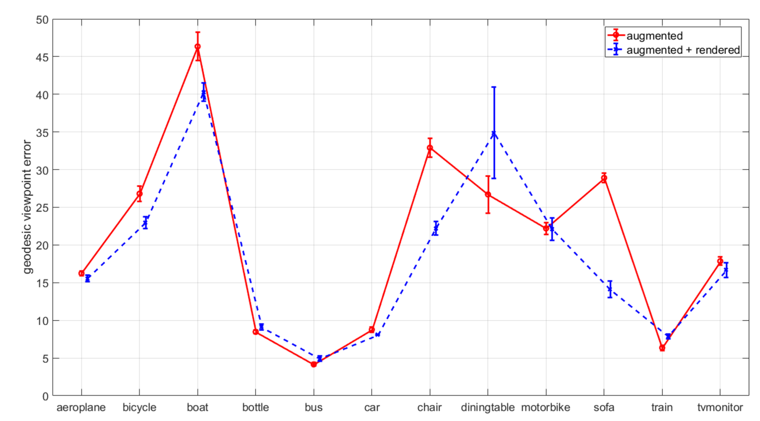
Choice of Feature Network: VGGM- FC6 vs FC7 vs POOL5



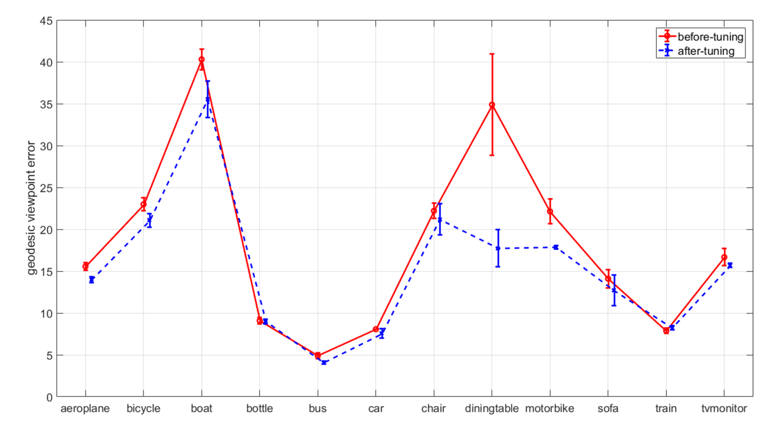
Choice of Feature Network: VGGM-FC6 vs VGG16-FC6



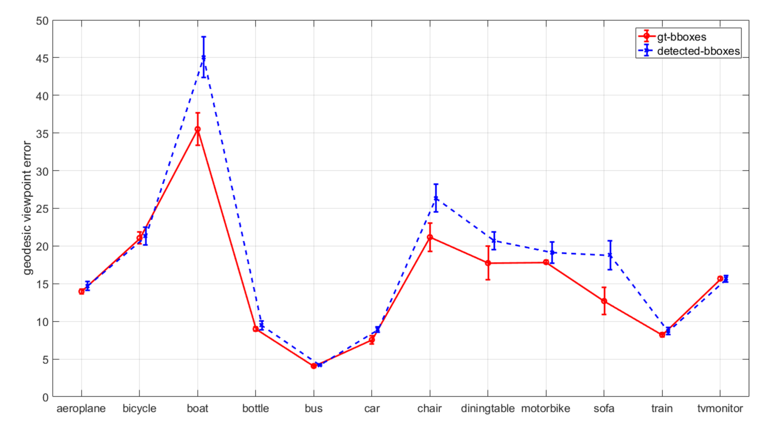
Optimization for Pose Networks: MSE(20) vs GVE(20) vs MSE(10) + GVE(10). $\mathrm{MSE} \doteq \|y - \hat{y}\|_2^2$



Data Augmentation: With and without rendered images



Optimization: With and without finetuning



With detected bounding boxes

This work was supported by NSF grant 1527340.