Exploit Clues from Views: Self-Supervised and Regularized Learning for Multiview Object Recognition

Chih-Hui Ho Bo Liu Tz-Ying Wu Nuno Vasconcelos
University of California, San Diego
{chh279,boliu,tzw001,nvasconcelos}@ucsd.edu

A. Baseline implementation details

In this section, we provide more implementation details for autoencoder, egomotion and shapecode baselines.

Autoencoder [2] is trained to reconstruct an input image x, with dimension 224×224×3. We adopt the autoencoder architecture design of VGG16 from [3]. However, the architecture in [3] is similar to Unet [5], which has intermediate connections between different intermediate feature outputs. We remove those intermediate connections to resemble autoencoder, whose latent feature has 4096 dimension. L2 loss is used to measure the difference between ground truth $y = x$ and the reconstructed image $\hat{y}$, which can be formulated as.

$$\mathcal{L} = ||\hat{y} - y||^2. \tag{1}$$

Egomotion [1] predicts the camera motion between 2 images. Assuming only V viewpoints exist in the dataset, the model will output $V - 1$ probabilities, corresponding to the $V - 1$ viewpoints differences. Given a pair of images x_1 and x_2, two 4096 dimention features f_1 and f_2 are extracted from the last layer of VGG16. These 2 feature are then concatenated into a 8192 dimension feature, which is then fed into a stacked fully connected layers ($8192 -> 4096 -> 1024 -> V - 1$) to predict the relative view point difference using softmax.

ShapeCode [4]. We again use architecture similar to autoencoder, but instead of outputting a reconstructed image $\hat{y}$, the network outputs $\{\hat{y}^j\}_{j=1}^{V}$, where V is the total number of images associated to the same object. Specifically, given the input image x with 224×224×3 dimension, the output of the network has dimension 224×224×3×V and the loss function becomes

$$\mathcal{L} = \sum_{j=1}^{V} ||\hat{y}^j - y^j||^2 \tag{2}$$

Note that these V images are organized sequentially and support missing views as suggested in [4].

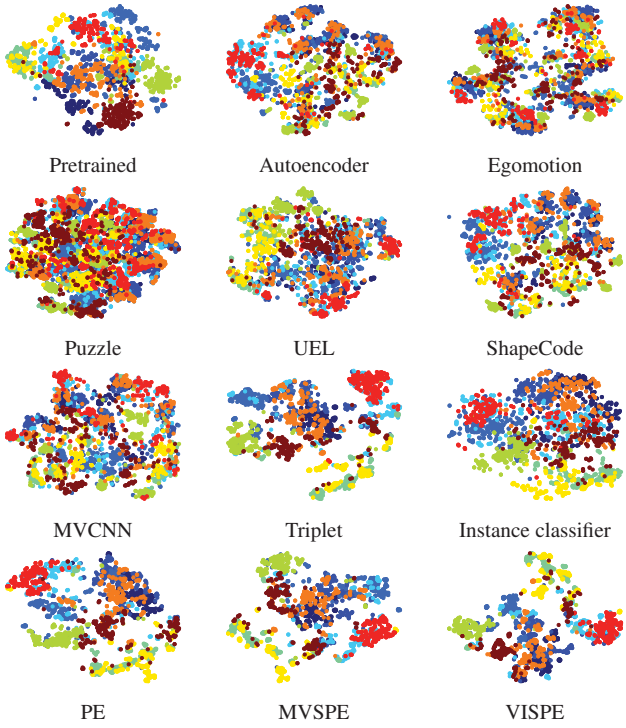

Figure 1: TSNE visualization of the **unseen class** embedding produced by different baselines. Each color represent a class. All three proposed approaches have better embedding structure as embedding from same class are more compact, while embedding from different classes are spread out.

B. TSNE plots

Aside from the subplots in the main paper, we also visualize the features from all the methods in Figure1.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 37–45, Dec 2015.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[3] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with VGG11 encoder pre-trained on imagenet for image segmentation. *CoRR*, abs/1801.05746, 2018.

[4] Dinesh Jayaraman, Ruohan Gao, and Kristen Grauman. Unsupervised learning through one-shot image-based shape reconstruction. *CoRR*, abs/1709.00505, 2017.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.