Reaction Report for "RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints"

Nishant Raj

What I like about this paper?

The paper builds upon idea from MVCNN to use existing 2D architectures for this 3D task allowing to reuse pretrained weights. Since it also uses existing 2D architectures, it becomes easy to use pretrained weights trained on large 2D datasets. The major contribution from the paper is the idea that different view configurations can be treated as latent variables and that these latent variables can be used further to jointly estimate the category of the object as well as the best pose. One good component that I liked about the design of the final likelihood function (equation 6) in the paper is that it makes it possible for test image views to be fed sequentially and hence this could be very easily extended for use in the real-world settings with a moving camera (https://kanezaki.github.io/media/CVPR2018_RotationNet_supp.mp4). The fact that the paper figures out the best view and focusses on interclass pose alignment is also helpful because this will help to differentiate features in a better way (It makes more sense to compare a cup (with a handle) and a coffee-mug (without handle) in a view where the differentiating features i.e., the handle is easily visible). The datasets present in 3D Computer Vision are very limited. This paper has also discussed a dataset that authors created i.e., "MIRO" which could potentially overcome the downsides of RGBD datasets i.e., inconsistencies in different upright orientation and insufficient number of object classes per category.

What I do not like about this paper? One of the major explorations that I found was missing from the paper was exploration from non-homogeneous view settings. It is unclear which setting out of homogeneous positions of camera or the heterogeneous positions would yield more optimal settings and this needs to be investigated further. One other thing that I did not find interesting was that the performance of the paper was lesser in base setting with upright orientation. If the number of views in the test cases are less, then the architecture might not yield the best performance which was evident from the fact that in upright setting with lesser views (~1-5), it's performance was lesser than the MVCNN architecture. However, in second case with full 3d rotation, it does overcome that limitation.

Future Directions: There are two-three avenues that can be investigated and explored further. One of the explorations as discussed in the final paragraph of paper is determination of best camera orientation. This can be an iterative process to check which placements in general yield better performance, but the number of iterations would make this approach difficult. Hence, a method that automates the selection of camera positions for best performance might be needed and research in geometric placement of cameras might be taken further. Closely related to this, non-homogeneous view positions of the camera can be explored further to capture non-obstructive views as well and see if it yields a boost in performance. Multiple cameras can be used at varying positions and in order to find the best suitable view, we can leverage the help of Graph Convolutional Networks. Since the views would form a graph like structure, we can further coarsen the graph to capture those features that bring about differentiation amongst different views. Clustering of nodes and then pooling of information in these coarsened graphs from different views is something that can be further investigated. Apart from this, it would be interesting to see how these latent variables can be leveraged in some way for the segmentation tasks.