Letting 3D Guide the Way: 3D Guided 2D Few-Shot Image Classification

Supplementary Material

Jiajing Chen, Minmin Yang, Senem Velipasalar
Syracuse University, Electrical Engineering and Computer Science Dept., Syracuse, NY, USA
{jchen152, myang47, svelipas}@syr.edu

1. Overview

In this supplementary document, we provide the results of additional studies we performed to see the effects of different factors on the overall performance. More specifically, in Section 2, we study the benefit of considering angle variety when forming the support set by comparing our method with the traditional approach of ProtoNet. We analyze the effect of increasing the number of classes and increasing the intra-class variance in the support set in Sections 3 and 4, respectively. Analysis of using different number of angle categories during query view angle inference is provided in Section 5. The effect of pretraining the backbone on the performance is analyzed in Section 6.

For the experiments in Sections 2, 3 and 4, we use RGB images collected from the Web as query images. These images are split into set (a) and set (b) shown in Fig. 1. Classes used for these experiments are listed in Table 1. More specifically, instead of providing a projection image, the user only provides 2D RGB images as queries, and we use an existing set of 3D meshes to form a support library of projection images from different angles. These experiments also show that our method can be applied when RGB images are used as queries.

Class ID	Class Name	Class ID	Class Name	Class ID	Class Name
0	Airplane	5	Bed	10	Bookshelf
1	Bottle	6	Bench	11	Bread
2	Bowl	7	Crab	12	Curtain
3	Chair	8	Bus	13	Banana
4	Cone	9	Bicycle	14	Chair

Table 1. List of classes used in the experiments, wherein RGB images are employed as queries, and projection images from 3D meshes are employed as support set.

2. Comparison with ProtoNet

As described in our paper, in our proposed 3DG2D approach and the proposed Angle Inference Network (AINet), support projections, which are obtained from different angles, provide a more complete description of an object's shape. Our proposed AINet first infers a query image's view angle, and then gives more weight to the support projections, which are taken under similar view angles as the query image, during testing phase. In contrast, ProtoNet [1] and other traditional approaches, randomly pick support and query images from the same RGB image pool, without considering variations in view angle, and treat all the support images equally.

	Airplane	Bottle	Bowl	Chair	Cone	Mean
ProtoNet	0.00%	7.14%	20.00%	0.00%	4.76%	6.38%
Ours	100.00%	80.00%	81.82%	81.82%	69.23%	82.57%

(a) Experiment result on query set (a)

	Airplane	Bottle	Bowl	Chair	Cone	Mean
ProtoNet	0.00%	26.32%	7.69%	0.00%	4.76%	7.75%
Ours	90.00%	90.91%	90.00%	90.91%	100.00%	92.36%

(b) Experiment result on query set (b)

Table 2. IoU value for each class for the experiments performed on query sets (a) and (b).

In this experiment, we perform a comparison with the approach of original ProtoNet to demonstrate the effectiveness of our method. First, we pick the first five classes (with IDs 0-4) from Table 1 to perform 5-way,3-shot (42 projections), 10-query experiment. For our proposed AINet, we use 42 projections for each class as the support set. For ProtoNet, since it does not consider angle variety when forming the support set, for each class, an angle category (Bottom, Horizontal or Top) is picked first. Then, 42 projections obtained from that specific angle category are used as the support set for ProtoNet. For fair comparison, both networks are used with the same well-trained parameters.

In Table 2, the IoU value for each class is shown for the experiments performed on query sets (a) and (b). As can be seen, our proposed method performs well on RGB images collected from the Web. As for ProtoNet, since the support

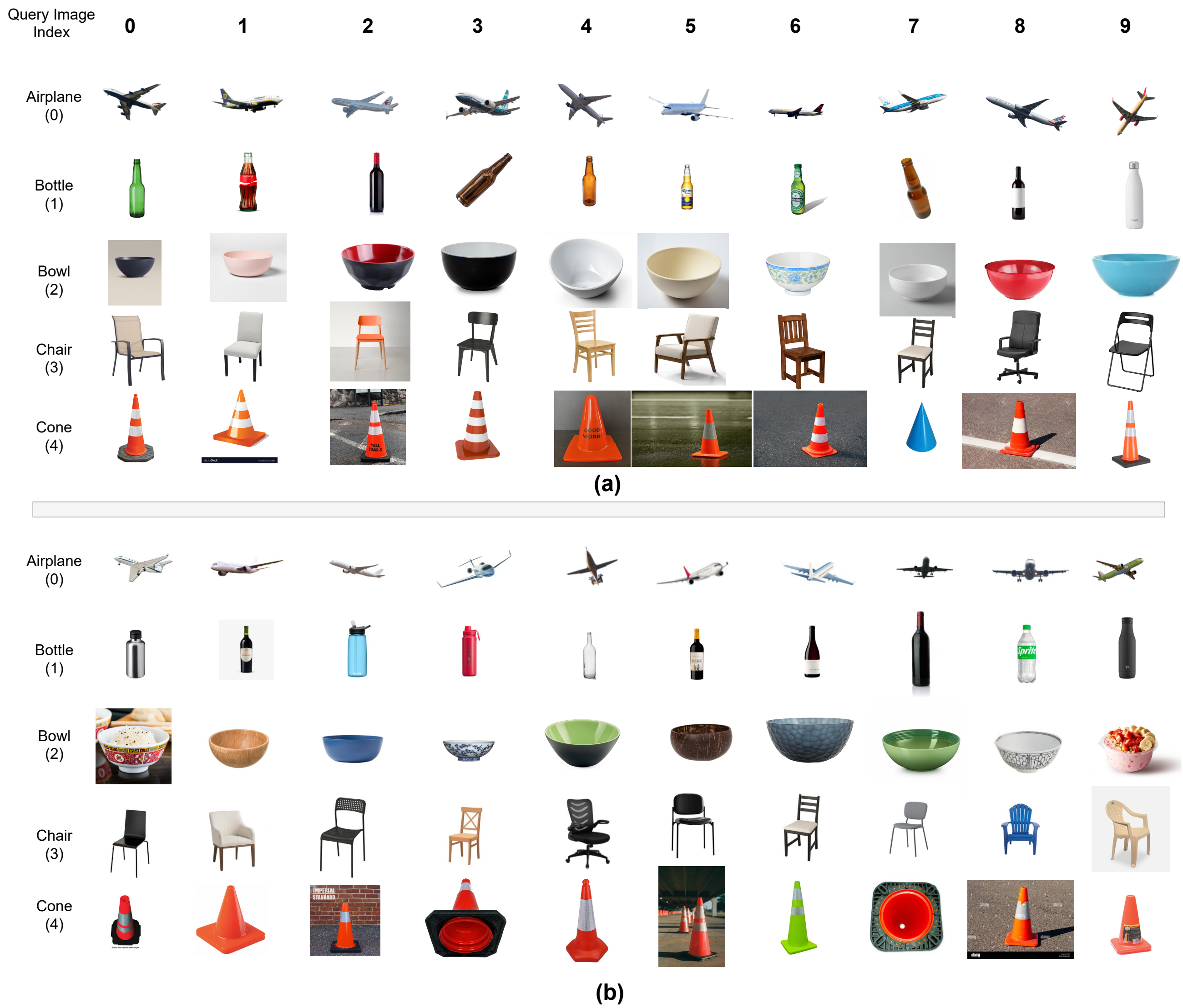

Figure 1. Query RGB images collected from the Web. The images are from five classes, namely 'Airplane', 'Bottle', 'Bowl', 'Chair' and 'Cone'. These query images are divided into set (a) and set (b), where each set contains a total of 50 images, with 10 images for each class.

set is composed of images from one view angle, the model fails on this task.

3. Number of Classes in the Support Set

In traditional few-shot learning settings, a query image belongs to one of the classes in the support set, and it is matched to the closest image in the support set. Let's make the problem more challenging especially considering the 3DG2D approach we propose. Let's say a user has only RGB images and no 3D mesh data, and wants to use these RGB images as queries. Then, we can use an existing library of projection images as the support set. However, since we do not know the class of the query, this library should ideally contain all the projection images we have available. To simulate this scenario, we use query images from 5 different classes, and increase the number of classes in the support set from 5 to 10 and then to 15 to study how the size of the support set affects the overall performance. The results are provided in Table 3. For both experiments performed with query sets (a) and (b), when the number of classes in the support set is increased from 5 to 10, the mean IoU drops, as expected. When the number of classes is further increased to 15, the amount of drop decreases.

4. Intra-Class Variance in Support Set

In this section, we study how the intra-class variance in the support set affects the overall performance of AINet. We use the query set (a) shown in Fig. 1 as query images.

Class Number	Airplane	Bottle	Bowl	Chair	Cone	Mean
5	100.00%	80.00%	81.82%	81.82%	69.23%	82.57%
10	80.00%	70.00%	70.00%	81.82%	61.53%	72.67%
15	80.00%	50.00%	70.00%	81.82%	53.84%	67.13%

(a) Experiment result on query set (a)

Class Number	Airplane	Bottle	Bowl	Chair	Cone	Mean
5	90.00%	90.91%	90.00%	90.91%	100.00%	92.36%
10	70.00%	54.54%	80.00%	100.00%	60.00%	72.91%
15	70.00%	20.00%	80.00%	100.00%	50.00%	64.00%

(b) Experiment result on query set (b)

Table 3. (a) and (b) show IoU values for each class and mean IoU for the experiments performed on query sets (a) and (b), respectively, when the number of classes in the support set is varied.

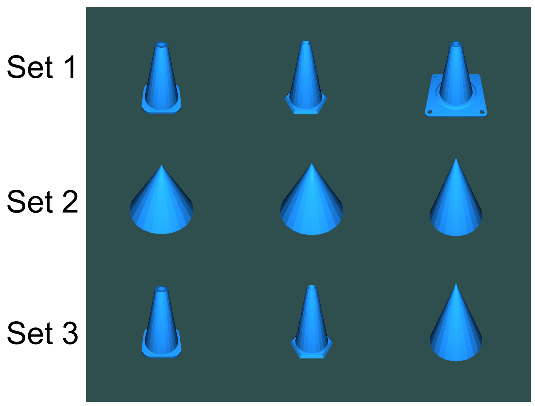

Figure 2. Three different support sets of 3D mesh data for the cone class. Here, we only show one projection from each 3D mesh data for visualization.

As for the support set of 3D meshes, we use three different sets for the cone class while keeping the support set of other classes the same as Section 2. The three different support sets used for the cone class are shown in Fig. 2. In Set 1, the support set of 3D meshes only contains traffic cones. In Set 2, we have the general geometric representations of cones, and no traffic cones. Finally, in Set 3, we have a mix of these two types of cones. The results of the experiments performed by using these three different sets of cones, are shown in Table 4. The results show that using projections from Set 3, i.e. including both types of cones together in the support set, provides the best performance. When we only use the projections from Set 2 as the support set for the cone class, the bottom projections are confused with the bowl class, since they are circular. This causes a drop in the IoU values of both cone and bowl classes. Similar phenomenon occurs when we use Set 1 as support. In this case, some projections of the cone are wrongly matched with a bottle, causing drop in the IoU values of both the cone and bottle classes (compared to using Set 3). Thus, it is beneficial to represent the variety of shapes for a class in the support set.

	Airplane	Bottle	Bowl	Chair	Cone	Mean
Set 1	100.00%	70.00%	75.00%	81.82%	46.67%	74.70%
Set 2	100.00%	80.00%	66.67%	81.82%	54.54%	76.61%
Set 3	100.00%	80.00%	81.82%	81.82%	69.23%	82.57%

Table 4. IoU values of each class when different sets (shown in Fig. 2) of cones serve as support data.

5. Analysis of Number of Angle Categories

In our proposed AINet, 14 view angles are divided into three angle categories of bottom (B), horizontal (H) and top (T) to infer the view angle of a query image. In this section, we analyze how using different number of angle categories affects the performance. For this, we perform an experiment wherein we use all of the 14 angles as separate classes. The results of this experiment, performed on ModelNet40 dataset, are shown in Table 5. As can be seen, AINet with three angle categories provides the best performance. Treating each view as a separate angle category increases the angle category estimation error, compared to using three classes. At the same time, whether three or 14 angle categories are used, AINet outperforms ProtoNet in both cases, since considering view angles of query and support images is better than not considering it at all.

Model	Number of Angle Categories	Fold 0	Fold 1	Fold 2	Fold 3	Mean
ProtoNet	/	72.04%	68.53%	55.71%	59.18%	63.86%
AINet	14	73.01%	68.64%	55.83%	59.48%	64.24%
AINet	3	**73.47%**	**69.56%**	**56.59%**	**61.77%**	**65.34%**

Table 5. Performance comparison of AINet when different number of angle categories are used.

6. Analysis of Pretraining

In this experiment, we study the effect of pretraining the backbone. We use the ModelNet40 dataset for a 5-way-1-shot classification task. The ProtoNet and AINet are tested with and without pretraining the ResNet backbone. The results are summarized in Table 6. It can be seen that, with the pretrained backbone, the performance of both ProtoNet and our proposed AINet are boosted. AINet outperforms ProtoNet in both cases and for all folds. More specifically it provides performance improvement of 1.77% and 1.48% with and without pretraining, respectively.

Model	Pretraining	Fold 0	Fold 1	Fold 2	Fold 3	Mean
ProtoNet	w/o. Pretraining	70.42%	60.29%	51.60%	55.18%	59.37%
	w. Pretraining	72.04%	68.53%	55.71%	59.18%	63.86%
AINet	w/o. Pretraining	71.35%	62.65%	53.18%	57.39%	61.14%
	w. Pretraining	73.47%	69.56%	56.59%	61.77%	65.34%

Table 6. Experiment results comparing AINet and ProtoNet with and without a well-pretrained backbone.

References

[1] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 1