Dynamic Few-Shot Visual Learning without Forgetting: Supplementary Material

Spyros Gidaris
University Paris-Est, LIGM
Ecole des Ponts ParisTech
spyros.gidaris@enpc.fr

Nikos Komodakis
University Paris-Est, LIGM
Ecole des Ponts ParisTech
nikos.komodakis@enpc.fr

Contents

1. Qualitative comparison of cosine similarity based features using t-SNE scatter plots **1**

2. Implementation details of training procedure followed during the 2nd training stage **1**

1. Qualitative comparison of cosine similarity based features using t-SNE scatter plots

Here we compare qualitatively the feature representations learned by the proposed cosine-similarity based ConvNet recognition model with those learned by the typical dot-product based ConvNet recognition model. For that purpose in Figure 1 we provide the t-SNE [1] scatter plots that visualize the local-structures of the feature representations learned in those two cases. Note that the visualized features are from the validation categories of the Mini-ImageNet dataset that are "unseen" during training. Also, in the case of the cosine-similarity based ConvNet recognition model, we visualize the l_2-normalized features, which are the features that are actually learned by the feature extractor.

We observe that the feature extractor learned with the cosine-similarity based ConvNet recognition model, when applied on the images of "unseen" categories (in this case the validation categories of Mini-ImageNet), it generates features that form more compact and distinctive category-specific clusters (i.e., more discriminative features). Due to that, as it was argued in section 3.1 of the main paper, the features learned with the proposed cosine-similarity based recognition model generalize better on the "unseen" categories than the features learned with the typical dot-product based recognition model.

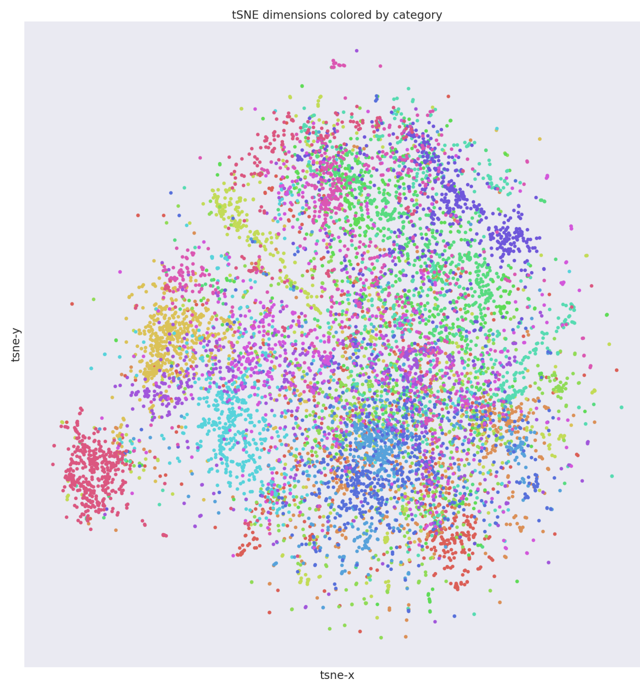
2. Implementation details of training procedure followed during the 2nd training stage

As explained in section 3.3, in order to train the few-shot classification weight generator, during the 2nd training stage we sample K_{novel} "fake" novel categories from the base training categories and we treat them in the same way as we will treat the actual novel categories after training. More specifically, during the 2nd training stage we form "training episodes"; each "training episode" is created by sampling: **(a)** K_{novel} "fake" novel categories with N' training examples per "fake" novel category, **(b)** T_{novel} test image examples from the "fake" novel categories, and **(c)** T_{base} test image examples from the remaining base categories (i.e., the base categories without the "fake" novel categories). Given such a "training episode", we first use the N' training examples of each "fake" novel category to infer with the few-shot weight generator a "fake" novel classification weight vector for that category and then we use the union of "fake" novel classification weight vectors and the classification weight vectors of the remaining base categories in order to learn to classify the $T = T_{novel} + T_{base}$ test image examples. To conclude, in order to train the few-shot classification weight generator and the classification weight vectors of the base categories we use stochastic gradient descent based routines with training batches that include multiple different instances of the above "training episodes".

References

[1] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 1, 2

(a) Cosine-similarity based features of validation categories



(b) Dot-product based features of validation categories

Figure 1: Here we visualize the t-SNE [1] scatter plots of the feature representations learned with **(a)** the cosine-similarity based ConvNet recognition model, and **(b)** the dot-product based ConvNet recognition model. The visualized feature data points are from the "unseen" during training validation categories of Mini-ImageNet. Each data point in the t-SNE scatter plots is collored according to its category.