IRGen: Generative Modeling for Image Retrieval

Yidan Zhang[12], Ting Zhang[1*], Dong Chen[3], Yujing Wang[3], Qi Chen[3], Xing Xie[3], Hao Sun[3], Weiwei Deng[3], Qi Zhang[3], Fan Yang[3], Mao Yang[3], Qingmin Liao[5], Jingdong Wang[4], and Baining Guo[3]

[1]Beijing Normal University [2]The University of Tokyo [3]Microsoft [4]Baidu
[5]Shenzhen International Graduate School, Tsinghua University

1 Image Tokenizer Detail

Fig. 1 shows the architecture of our image tokenizer, which is an encoder only network where we enforce classification loss over feature embeddings. We indeed utilize the feature from the "class token" to perform residual quantization.

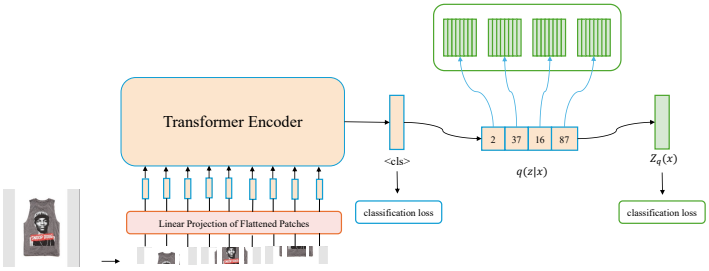

Fig. 1: The framework of the image tokenizer.

2 Dataset Detail

In-shop Clothes retrieval dataset [3] is a large subset of DeepFashion with large pose and scale variations. This dataset consists of a training set containing 25,882 images with 3997 classes, a gallery set containing 12,612 images with 3985 classes and a query set containing 14,218 images with 3985 classes. The goal is to retrieve the same clothes from the gallery set given a fashion image from the query set. We use both the training set and the gallery set for training in our experiments.

CUB200 [6] is a fine-grained dataset containing 11,788 images with 200 classes belong to birds. There are 5,994 images for training and 5,794 images for testing.

Cars196 [2] is also a fine-grained dataset about cars. It contains 16,185 images with 196 car classes, which is split into 8,144 images for training and 8,041 images for testing.
ImageNet dataset [1] contains 1,281,167 images for training and 50,000 validation images for testing, in which we randomly sample 5,000 images as queries to speed up the evaluation process.
Places365-Standard [7] includes about $1.8M$ images from 365 scene categories, where there are at most 5000 images per category.

3 Implementation Detail

We adopt ViT-B for encoder and similar architecture for decoder (12 transformer decoder block with dimension 768). The input image is of resolution 224×224 and is partitioned to 14×14 patches with each patch sized 16×16. Intuitively, a warm initialization of encoder should largely stable the training process. We thus warm-start the model with encoder initialized by the pretrained CLIP model [5]. We randomly initialize the remaining fully connected layer and the decoder. The semantic image tokenizer is trained with a batch size of 128 on 8 V100 GPUs with 32G memory per card for 200 epochs. We adopt an AdamW optimizer [4] with betas as $(0.9, 0.96)$ and weight decay as 0.05. We use cosine learning rate scheduling. Note that we set the initial learning rate as $5e-4$ for the FC layers. The learning rate of the encoder is set as one percentage of the learning rate of FC layers. We train our models with 20 warming-up epochs and the initial learning rate is $5e-7$. For training autoregressive model, we select similar image pairs (x_1, x_2). Since current retrieval datasets are usually labeled with class information, we randomly sample an image x_2 which shares the same class with x_1 as the nearest neighbor. For autoregressive model, we use batch size of 64 on 8 V100 GPUs with 32G memory per card for 200 epochs. The optimizer and the scheduler are same as the semantic image tokenizer mentioned above. The initial learning rate is $4e-5$ for the decoder and the learning rate for encoder is always one percentage of that for decoder. The hyperparameter for quantization is set to $M = 4$ and $L = 256$ for fast inference. For ImageNet and Places365, the experimental settings are the same as before except that we enlarge the layer of decoder to 24 to increase the capacity for AR modeling.

4 Ablation Study

The effect of sequence length. We further investigate the length of identifier in our image tokenizer. We experiment different lengths and report the results in Tab. 1. We can see that if the length of the identifier is too small (for example 2), the model gets inferior performance. As with the length gets longer to 4 or 6, the model gets better performance. At last the performance drops a little bit if the length is too long (8). We think 4-6 would be a good choice in most cases and we simply use 4 in all our experiments.

Table 1: Ablation study on the sequence length T.

T	Precision				Recall			
	1	10	20	30	1	10	20	30
2	72.1	69.6	68.9	68.6	72.1	95.1	96.6	97.1
4	92.4	87.0	86.6	86.5	92.4	96.8	**97.6**	**97.9**
6	92.8	87.2	86.8	86.7	92.8	96.7	97.4	97.8
8	**92.9**	**87.4**	**87.0**	**86.9**	**92.9**	**96.9**	97.5	97.8

The effect of autoregressive decoder. One natural baseline is to directly apply beam search to the prefix tree derived from RQ codes learned by the tokenizer, rather than remodeling the semantic relationship between RQ codes through the autoregressive decoder. The comparison results on ImageNet dataset are summarized in Tab. 2. The performance of this baseline is significantly inferior to our proposed method, highlighting the importance of our autoregressive decoder.

Table 2: Ablation study on the autoregressive decoder.

Method	MAP@100
RQ prefix-tree	56.7
IRGen(Ours)	**76.0**

The effect of reconstruction loss. During the training of image identifier, we propose to utilize M levels of partial reconstruction loss besides the traditional classification loss, as shown in Equation (3). Tab. 3 ablates this reconstruction loss and shows that this loss is beneficial for learning semantic image identifier.

Table 3: Ablation study on the reconstruction loss during the training of image identifiers.

Loss	Precision			
	1	10	20	30
Only classification loss	92.1	85.1	83.4	82.5
Full loss	**92.4**	**87.0**	**86.6**	**86.5**

5 Qualitative Retrieval Results

In this section, we provide several retrieval examples that showcase the performance of our approach compared to baselines. The retrieval results on In-shop Clothes, Cars196, and ImageNet using different methods are depicted in Fig. 4, Fig. 2, and Fig. 3, respectively. Correctly retrieved images are highlighted with green borders, while incorrectly retrieved ones are marked with red borders. Upon examining the results presented in these figures, it becomes evident that our proposed method performs exceptionally well and is capable of handling even the most challenging examples.

Fig. 2: Examples on Cars196 dataset.Results of CGD, IRT, FT-CLIP, our IRGen are shown from top to bottom.The results of CGD, IRT, FT-CLIP are retrieved by SPANN.

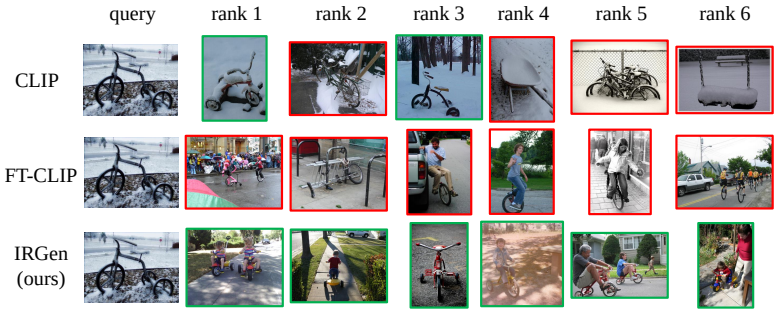

Fig. 3: Examples on ImageNet dataset.Results of CLIP, FT-CLIP, our IRGen are shown from top to bottom. The results of CLIP, FT-CLIP are retrieved by SPANN.

Fig. 4: Examples on In-shop Clothes dataset.Results of CGD, IRT, FT-CLIP, our IRGen are shown from top to bottom.The results of CGD, IRT, FT-CLIP are retrieved by SPANN.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
3. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1096–1104 (2016)
4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
5. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
6. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
7. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)