Deep Semantic Coding for Wireless Image Retrieval

Ying Wang, Chenhao Qi
National Mobile Communications Research Laboratory
School of Information Science and Engineering, Southeast University, Nanjing, China
Email: {wying, qch}@seu.edu.cn

2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall) | 978-1-6654-5468-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/VTC2022-Fall57202.2022.10012945

***Abstract*—We address the image retrieval problem for a wireless system including an edge server and an edge device. The query image is first compressed by the edge device, and then transmitted into wireless channel, while the edge server retrieves the received image. Different from conventional schemes directly compressing features via unsupervised learning regardless of the database semantic distribution, we design a deep semantic coding (DSC) scheme by integrating the inverted semantic index structure of the database into the coding process, which can utilize the prior semantic information of the database to reduce the bandwidth. We extract the feature vectors from the images via a convolutional neural network and generate the semantic guided code head, which is followed by the product quantization. The experimental results verify the effectiveness of the DSC scheme in reducing the bandwidth as well as improving the performance of wireless image retrieval.**

***Index Terms*—Convolutional neural network, deep learning, semantic coding, wireless image retrieval**

I. Introduction

The semantic communication systems currently consider the case that one of the intelligent tasks needs to be completed by internet of things (IoT) device [1]. To solve the problem that subjective semantic information is difficult to extract, a series of engineering realizable semantic communication methods are proposed for different types of information sources based on deep learning technology [2]–[9]. For semantic text transmission, deep learning based semantic communications (DeepSC) proposes a communication system based on Transformer [4]. Then a lightweight distributed semantic communication system is further designed so as to deploy DeepSC on IoT devices [5]. Besides, DeepSC is extended to speech signal transmission and a semantic coding method is investigated based on attention mechanism, named as DeepSC-S [6]. The widely used method joint source channel coding (JSCC) based on LSTM network is firstly considered for text and speech transmission [7]. Owing to the superiority of JSCC, some researchers apply it to the image transmission task. For example, a JSCC approach based on convolutional neural network (CNN) realizes image transmission in the wireless channel, which optimizes the semantic coding to improve the performance of image transmission [8]. Inspired by this work, a retrieval-oriented image compression scheme is proposed, which is the first work to study image retrieval over wireless channel [9].

Among machine learning tasks, retrieval is one of the indispensable tasks for remote inference which can be applied to autonomous vehicles or surveillance systems to identify vehicles, persons or scenes according to sensory data. Through retrieval technologies, the query image of a person, a vehicle or a scene is searched in a large database on edge servers as shown in Fig. 1. For the reason that the central large-scale data can not be visited by edge devices and images and videos are too big, the sensory data needs to be compressed. In this work, we study deep semantic coding methods at the wireless edge following the wireless image retrieval scheme [9]. In particular, in order to make use of the prior knowledge from the database to reduce the bandwidth, we focus on the semantic compression method combining the large-scale retrieval technologies, including inverted index and approximate nearest neighbor (ANN) search.

We propose two coding methods for the wireless image retrieval. The first one is the deep unsupervised coding (DUC) method, which is based on the conventional clustering-based inverted index, while the second one is the deep semantic coding (DSC) method, which is based on a inverted semantic index. In both two methods, the query image from an edge device is compressed into feature vector via convolutional neural network (CNN). And then the vector is quantized by an ANN search method product quantization (PQ). PQ has been recognized as the most popular solution in ANN search and inverted index for retrieval task [10]–[12]. When the large-scale database requires extortionate memory size, PQ is applied to compress the data into binary code. Considering that the goal of reducing memory size is consistent with bandwidth constraints, we apply PQ to the source coding of wireless image retrieval. Note that neither the memory savings nor the bandwidth reduction of the wireless retrieval task requires reconstruction of the source image. Thus lossy compression is acceptable in wireless image retrieval. Back to the point, the difference between DUC and DSC is the generation of code head based on inverted index which is used to locate the searched partitions fast and generate the candidate list for retrieval. The code head in DSC is semantic guided, which is proved to outperform the DUC method by simulation results. Our contributions can be summarized as follows

(1) We propose a semantic compression scheme aiming at wireless image retrieval for IoT, which combines the large-scale retrieval technologies, including inverted semantic index and product quantization, with the semantic coding operation.
(2) We extract feature from a source image via CNN, and then introduce a semantic guided code head followed by

978-1-6654-5468-1/22/$31.00 ©2022 IEEE

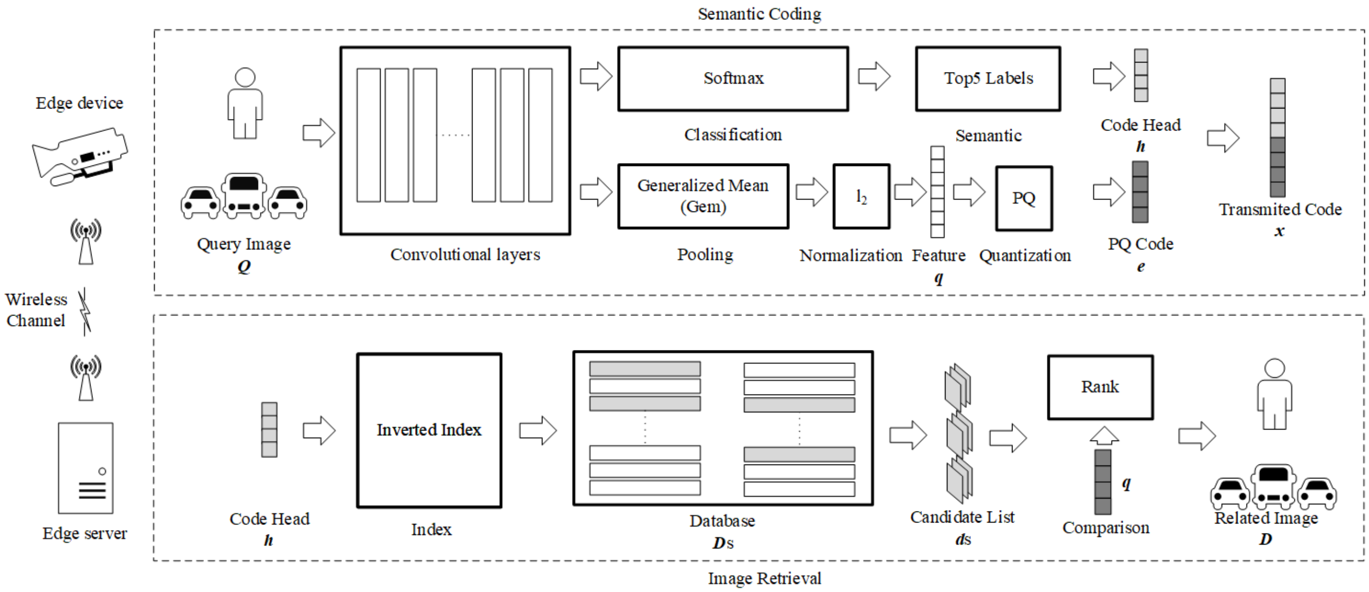

Fig. 1. Framework of wireless image retrieval based on deep semantic coding (DSC)

PQ code. This DSC method allows the wireless channel to transmit query image with a limited bandwidth and robust retrieval performance.

(3) We perform evaluations under different signal-to-noise ratios (SNRs) and different database scales. The result proves that the DSC method can not only improve the retrieval accuracy but also limit the bandwidth.

Notations: Symbols for vectors (lower case) and matrices (upper case) are in boldface. Symbols for sets and models are in calligraphy. For a vector $\boldsymbol{a}$, $\boldsymbol{a}_m$ and $\|\boldsymbol{a}\|_2$ denote its mth entry and l_2-norm. $\{\boldsymbol{a}_m\}$ denotes a set formed by $\boldsymbol{a}_m$. $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I_B})$ denotes the the additive white Gaussian noise with the zero mean and the variance being σ^2.

II. System Model

In wireless image retrieval at the edge [9], it is divided into three stages: learning codebook from the database, coding and transmitting the image captured by the edge device and retrieving the received images in the server database.

In the first stage, the pre-processing is performed on the database. Firstly, each database image $\boldsymbol{D}$ is extracted into feature $\boldsymbol{d}$ via CNN model $\mathcal{M}_f$. The details of $\mathcal{M}_f$ are introduced in (10). Then in DUC, all $\boldsymbol{d}$ are clustered into K partitions by k-means, and the centroids of these partitions are recorded as the codewords $\boldsymbol{c}^i, i = 1, 2 \ldots, K$ in the codebook $\mathcal{C}$ [10], [11]. While in DSC, the database is divided according to semantic labels via ResNet101 which is expressed as $\mathcal{M}_c$. This process is formulated as

$$\boldsymbol{p} = \mathcal{M}_c(\boldsymbol{D}), \boldsymbol{p} \in \mathbb{R}^K, \tag{1}$$

where p_i represents the possibility that $\boldsymbol{D}$ relates to the i-th semantic label. The partition index i that $\boldsymbol{D}$ belongs to is expressed as

$$i = \arg\max_{i} \boldsymbol{p}. \tag{2}$$

Then the codeword $\boldsymbol{c}^i$ of the ith partition needs to be computed to make up the codebook $\mathcal{C}$ by

$$\boldsymbol{c}^i = \arg\min_{\boldsymbol{c}} \sum_{j=1}^{N} \|\boldsymbol{d}_j - \boldsymbol{c}\|_2^2, \boldsymbol{c} \in \mathcal{W}_i \triangleq \{\boldsymbol{d}_1, \boldsymbol{d}_2 \ldots, \boldsymbol{d}_J\}, \tag{3}$$

where J is the amount of the data in the ith partition $\mathcal{W}_i \triangleq \{\boldsymbol{d}_j\}$. Secondly, all $\boldsymbol{d}$ of the ith partition are used to learn the sub-codebooks via PQ. PQ divides a vector $\boldsymbol{d} \in \mathbb{R}^F$ into M sub-vectors

$$\boldsymbol{d} = [\boldsymbol{s}_1, \boldsymbol{s}_2 \ldots, \boldsymbol{s}_M], \boldsymbol{s}_m \in \mathbb{R}^{\frac{F}{M}}, \tag{4}$$

and each sub-vector is clustered into L-bit sub-codewords $\{\boldsymbol{r}_m^1, \boldsymbol{r}_m^2 \ldots, \boldsymbol{r}_m^{2^L}\}, \boldsymbol{r}_m^x \in \mathbb{R}^{\frac{F}{M}}$. The sub-codebook of the mth sub-vector in the ith partition is given by

$$\mathcal{R}_m^i \triangleq \{\boldsymbol{r}_m^{i,1}, \boldsymbol{r}_m^{i,2} \ldots, \boldsymbol{r}_m^{i,2^L}\}. \tag{5}$$

Thus the sub-codebooks of the ith partition generated by PQ are expressed as

$$\mathcal{R}^i \triangleq \{\mathcal{R}_1^i, \mathcal{R}_2^i \ldots, \mathcal{R}_M^i\}. \tag{6}$$

All the $M * K$ sub-codebooks are represented as $\mathcal{R} \triangleq \{\mathcal{R}^i\}$. Generally, L is set to be 8, hence each feature is quantized to a code $\boldsymbol{e} \in \mathbb{R}^M$ in M bytes. Finally, the codebook $\mathcal{C}$ and sub-codebooks $\mathcal{R}$ are fed back to the edge device in the initial phase. The prior information stored in the codebooks can help reduce the bandwidth in the following coding stage.

In the second stage, the query image $\boldsymbol{Q}$ captured by edge device is compressed and quantized into binary code. We propose DUC method and DSC method for the coding of query image, and the details of them are introduced in section III. After coding, the quantized code is transmitted over

Algorithm 1 DSC scheme

```
Input: $\boldsymbol{Q}, \boldsymbol{D}, M, \mathcal{M}_f, \mathcal{M}_c$
(Pre-processing)
Obtain $\boldsymbol{d}$ from $\boldsymbol{D}$ via (10).
Obtain codebook $\mathcal{C}$ via (3).
Obtain sub-codebooks $\mathcal{R}$ via (4)(5)(6).
(Deep Semantic Coding)
Obtain $\boldsymbol{q}$ from $\boldsymbol{Q}$ via (10).
Obtain code head $\boldsymbol{h}$ via (11).
Obtain PQ code $\boldsymbol{e}$ via (6).
Obtain total code $\boldsymbol{x}$ via (12).
(Wireless Image Retrieval)
Transmit $\boldsymbol{x}$ via (7).
Decoding $\boldsymbol{y}$ according to $\mathcal{C}$ and $\mathcal{R}$.
Recover feature $\hat{\boldsymbol{q}}$ via (13).
Retrieve $\hat{\boldsymbol{q}}$ via (14) or (15).
Output: $\widetilde{\boldsymbol{D}}$ related to $\boldsymbol{Q}$
```

wireless channel. This transmission is lossy limited by the finite channel capacity and there is no need to reconstructing the original image for retrieval tasks. We evaluate the wireless performance by additive white Gaussian noise (AWGN) channel in this paper. For an input B-dimensional vector $\boldsymbol{x} \in \mathbb{R}^B$, the output of the channel model $\boldsymbol{y} \in \mathbb{R}^B$ is

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\eta}, \tag{7}$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I_B})$ is the white Gaussian noise component with variance σ^2.

In the third stage, the compressed vector received by the receiver is compared with the vectors from the candidate list. The candidate list is generated by inverted index to avoid exhaustive searching. Through the comparison, the vectors are ranked to find the nearest neighbors, thus the target persons, vehicles or scenes are obtained. We evaluate the transmission and retrieval performance for different channel SNRs given by $\frac{P}{\sigma^2}$ and different database scales.

III. Deep Semantic Coding

In this section, the details of DSC at the second stage are introduced, including the feature extraction via CNN, the generation of semantic guided code head and the quantization compression via PQ.

A. Feature extraction via CNN

Following the state-of-the-art image retrieval method [13], we apply the generalized-mean (GeM) pooling layer to the ResNet101 and train on the dataset used in [14] for feature extraction. If we define f_n as the nth element of the pooling layer output vector $\boldsymbol{f}$ and $\boldsymbol{X}_n$ as the corresponding input array to f_n, the result of generalized-mean (GeM) pooling layer is given by

$$f_n = \left(\frac{1}{|\boldsymbol{X}_n|} \sum_{x \in \boldsymbol{X}_n} x^{p_n} \right)^{\frac{1}{p_n}}, \tag{8}$$

where the computation is equal to max pooling when $p_n \to \infty$ and average pooling for $p_n = 1$. The pooling parameter p_n can be manually set or learned by back-propagation [13]. We train the ResNet101 using Adam with learning rate 10^{-6}, momentum 0.9 and a batch size of 5 training tuples [13]. The last layer of the network is an l_2-normalization layer so that similarity between two images can be evaluated with inner product. In this work, all training images are resized to 362×362 and the output feature size is 2048. The training input consists of image pairs $(\boldsymbol{Q}, \boldsymbol{D})$ and label $l(\boldsymbol{Q}, \boldsymbol{D}) \in \{0, 1\}$ declaring whether the image pair is matched. The loss function is defined as

$$\mathcal{L}(\boldsymbol{Q}, \boldsymbol{D}) = \begin{cases} \frac{1}{2} \left\| \overline{\boldsymbol{q}} - \overline{\boldsymbol{d}} \right\|_2^2, & l(\boldsymbol{Q}, \boldsymbol{D}) = 1, \\ \frac{1}{2} (\max\{0, \tau - \left\| \overline{\boldsymbol{q}} - \overline{\boldsymbol{d}} \right\|_2)^2, & l(\boldsymbol{Q}, \boldsymbol{D}) = 0, \end{cases} \tag{9}$$

where $\overline{\boldsymbol{q}}$ is the ℓ_2-normalized GeM vector of image $\boldsymbol{Q}$, and τ is a threshold to make sure that the distance between the unrelated pairs is large enough so as to be ignored by the loss. The trained feature extraction model is represented as $\mathcal{M}_f$. The feature $\boldsymbol{q}$ of query image $\boldsymbol{Q}$ can be extracted via

$$\boldsymbol{q} = \mathcal{M}_f(\boldsymbol{Q}). \tag{10}$$

B. Semantic guided code head

After feature extraction, the quantization of the feature vectors is needed due to the bandwidth limitation. In this work, we consider two coding methods, including DUC method and DSC method.

In DSC, the classification possibilities of query image $\boldsymbol{Q}$ are predicted via (1). Then the top-5 most possible semantic labels are determined by

$$\{i_1, i_2 \ldots, i_5\} = \arg\max_i^5 \boldsymbol{q}. \tag{11}$$

The experience value 5 is inferred from the property that the top-5 accuracy of ResNet is up to 94.75%. These 5 indices can be stored in 50 bits, because there are $1k$ labels, which can be covered by 10 bits with $2^{10} > 1000 > 2^9$. And 50 bits can be covered by 7 bytes. Thus we generate the 7-byte semantic code head $\boldsymbol{h}$ via (1) and (11) which stores semantic information and can avoid exhaustive searching during retrieval. In DUC, we record the 5 nearest centroid indices in code head $\boldsymbol{h}$ correspondingly. Note that 5 is not the best value for DUC as shown in Fig. 2. However the best value is uncertain. We determine the code head size as 5 in DUC for comparison.

C. Quantized PQ code

For quantization, we employ PQ [15] to compress the feature vector $\boldsymbol{q}$ into binary code. Given the top-5 indices $\{i_1, i_2 \ldots, i_5\}$ in the code head $\boldsymbol{h}$, the codewords $\{\boldsymbol{c}^{i_1}, \boldsymbol{c}^{i_2}, \ldots, \boldsymbol{c}^{i_5}\}$ and sub-codebooks $\{\mathcal{R}^{i_1}, \mathcal{R}^{i_2} \ldots, \mathcal{R}^{i_5}\}$ can be selected. According to each sub-codebook, a code e in M-bytes can be obtained. M is generally set to be 8. Larger M will improve the retrieval accuracy but reduce the transmit

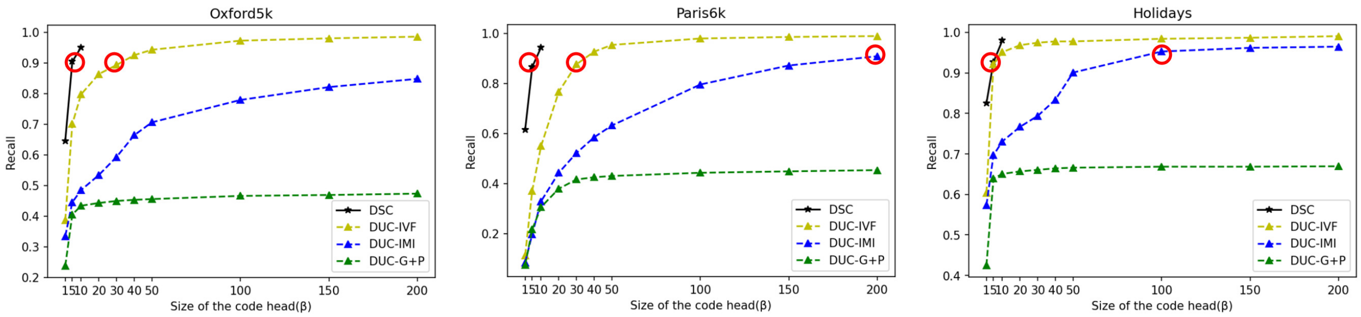

Fig. 2. Different sizes of code head

bandwidth. Thus the total code in $7 + 8 * 5 = 47$ bytes can be expressed as

$$\boldsymbol{x} \triangleq [\boldsymbol{h}, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_5]. \tag{12}$$

During the retrieval in the ith partition on edge server, according the codebook $\mathcal{C}$ and the sub-codebooks $\mathcal{R}^i$, the query feature $\boldsymbol{q}$ can be approximated by a sum

$$\hat{\boldsymbol{q}}^i \triangleq \boldsymbol{c}^i + [\boldsymbol{r}_1^{i,\mathrm{q}}, \boldsymbol{r}_2^{i,\mathrm{q}} \ldots, \boldsymbol{r}_M^{i,\mathrm{q}}], \tag{13}$$

where $\boldsymbol{c}^i$ is obtained by choosing the codeword out of $\mathcal{C}$ according to the indices from $\boldsymbol{h}$ and the sub-codewords $\boldsymbol{r}_m^i$ are gotten from $\mathcal{R}^i$ by the indices recorded in the corresponding $\boldsymbol{e}$. Similarly at the pre-processing stage, the database features $\boldsymbol{d}$s in the ith partition are also quantized into $\hat{\boldsymbol{d}} \triangleq \boldsymbol{c}^i + [\boldsymbol{r}_1^{i,\mathrm{d}}, \boldsymbol{r}_2^{i,\mathrm{d}} \ldots, \boldsymbol{r}_M^{i,\mathrm{d}}]$. The candidate list for retrieval is consist of the points from the 5 database partitions indexed by $\boldsymbol{h}$. Thus, the distance from query $\boldsymbol{q}$ to the point $\boldsymbol{d}$ in the candidate list is approximated by

$$\|\boldsymbol{q} - \boldsymbol{d}\|_2^2 \approx \|\hat{\boldsymbol{q}}^i - \hat{\boldsymbol{d}}\|_2^2 = \sum_{m=1}^{M} \|\boldsymbol{r}_m^{i,\mathrm{q}} - \boldsymbol{r}_m^{i,\mathrm{d}}\|_2^2, \tag{14}$$

while the cosine similarity between the query and the point in database is represented as

$$\begin{aligned} & s(\boldsymbol{q}, \boldsymbol{d}) \approx s(\hat{\boldsymbol{q}}^i, \hat{\boldsymbol{d}}) \\ = & \frac{\|\boldsymbol{c}^i\|_2^2 + \sum_{m=1}^{M} \left(\langle \boldsymbol{c}_m^i, \boldsymbol{r}_m^{i,\mathrm{q}} \rangle + \langle \boldsymbol{c}_m^i, \boldsymbol{r}_m^{i,\mathrm{d}} \rangle + \langle \boldsymbol{r}_m^{i,\mathrm{d}}, \boldsymbol{r}_m^{i,\mathrm{q}} \rangle\right)}{\sqrt{\sum_{m=1}^{M} \|\boldsymbol{c}_m^i + \boldsymbol{r}_m^{i,\mathrm{q}}\|_2^2 \sum_{m=1}^{M} \|\boldsymbol{c}_m^i + \boldsymbol{r}_m^{i,\mathrm{d}}\|_2^2}}. \end{aligned} \tag{15}$$

Typically, since the codeword $\boldsymbol{c}$ and 2^L sub-codewords $\boldsymbol{r}$ are pre-obtained and constant, we can pre-compute these dot-products and norms in (14) and (15), and store the results in lookup tables at the pre-processing stage [10]. Therefore, these terms can be reused directly from the lookup tables during the retrieval procedure. Owing to the lookup tables, the comparison and ranking can speed up greatly. The complexity for each comparison is decreased from $\mathcal{O}(F^2 + F - 1)$ to $\mathcal{O}(M)$ in (14) and to $\mathcal{O}(4M + 4)$ in (15).

IV. Simulation Results

In this section, we evaluate the transmission and retrieval performance of DUC and DSC approaches.

A. Datasets

Firstly, we introduce the standard image retrieval benchmarks, based on which we carry out the experiments.

1) *Oxford5k:* contains 5062 building images captured in Oxford with 55 query images. Each query has 6 to 221 target images. The differences between target images and query images include perspective conversion, light change, occlusion, etc.

2) *Paris6k:* is consist of 6412 architecture images from Paris, which also has 55 query images. The ground-truth amount of each query counts 51 to 289. The challenges are similar to the Oxford5k dataset.

3) *Holidays:* has 1491 personal holiday pictures with 500 query images. Each query has 2 to 13 target images. The retrieval difficulty is relatively low.

4) *Flickr100k and Flickr1M:* are distractors generally combined with the datasets mentioned above, which contain 100 thousand and 1 million images from Flickr, respectively.

B. Different size of code head

For the inverted index of edge server database, we consider the cluster-based IVF [16], IMI [10] and IVF+G+P [11] for DUC and a inverted semantic index based on classification for DSC. We perform experiments on different datasets as shown in Fig. 2. In order to reach the nearly 90% recall, the best number of indices in the code head are $\{5, 5, 5\}$ in DSC, $\{30, 30, 5\}$ in DUC-IVF and $\{-, 200, 100\}$ in DUC-IMI. Therefore, it is difficult to set the code head size to a certain number in DUC, which limit the application of DUC in practice. The robustness of DSC is owing to the high top-5 accuracy of ResNet101.

C. Different SNRs of wireless channel

For the DUC and DSC schemes we consider different channel SNRs for transmission and retrieval, varying from -20 dB to 5 dB. In the testing phase, we send both the original 2048-dimensional feature $\boldsymbol{q}$ and the quantized PQ codes $\boldsymbol{e}$ to the AWGN channel, adding the 7 bytes centroid code head in DUC and the 7 bytes semantic guided code head in DSC. Fig. 3 proves the effectiveness of PQ Compression with only slight drop on the retrieval accuracy mAP. And the addition of semantic code head in DSC improves the performance significantly.

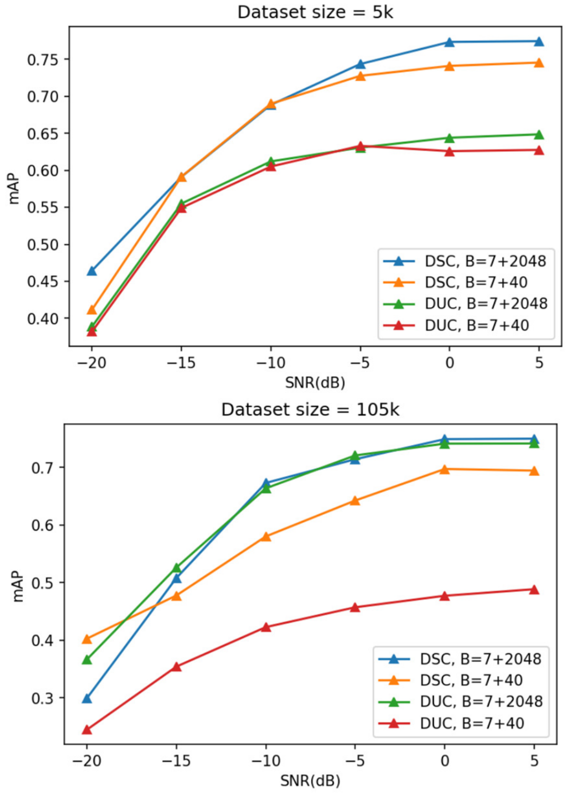

Fig. 3. Different SNRs on AWGN channel, where B represents the channel bandwidth.

D. Experiments on large-scale datasets

In this section, we evaluate DSC method and DUC method at the most ideal wireless channel without noise. The retrieval performance is evaluated on large-scale image retrieval benchmarks by mAP and $Recall$. The results in Tab. I reveal that our DSC method can really improve the retrieval performance with limited bandwidth.

TABLE I
EXPERIMENTS ON LARGE-SCALE DATASETS

Index	M	$Head$	mAP	$Recall$	$R@1$	$R@10$	$R@100$
			Oxford105k				
DUC-G+P [11]	2048	7	0.535	0.581	0.059	0.369	0.566
DUC-IVF [16]	2048	7	0.744	0.874	0.059	0.385	0.761
DUC-IMI [10]	2048	7	0.549	0.574	0.059	0.340	0.566
DSC	2048	7	**0.756**	**0.939**	**0.059**	**0.406**	**0.776**
DUC+PQ [15]	40	-	0.545	0.989	0.054	0.256	0.646
DUC-IVF+PQ [15]	40	7	0.485	0.874	0.027	0.159	0.635
DUC-IMI+PQ [10]	40	7	0.507	0.574	**0.059**	0.299	0.559
DSC+PQ	40	7	**0.702**	**0.939**	0.041	**0.340**	**0.790**
			Oxford1M				
DUC-G+P [11]	2048	7	0.627	0.726	0.059	0.395	0.673
DUC-IVF [16]	2048	7	0.676	0.803	0.059	0.385	0.689
DUC-IMI [10]	2048	7	0.482	0.533	0.059	0.291	0.505
DSC	2048	7	**0.719**	**0.939**	**0.059**	**0.399**	**0.735**
DUC+PQ [15]	40	-	0.133	0.798	0.030	0.078	0.198
DUC-IVF+PQ [15]	40	7	0.102	0.803	0.004	0.013	0.161
DUC-IMI+PQ [10]	40	7	0.399	0.533	**0.059**	**0.239**	0.445
DSC+PQ	40	7	**0.493**	**0.939**	0.028	0.178	**0.623**

V. CONCLUSION

In this work, we have studied the semantic coding method for wireless image retrieval. We have combined the inverted index and the source compression via the codebook of the database. Simulation results have demonstrated that the DSC method we propose can improve the retrieval performance and reduce the transmission bandwidth.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China (NSFC) under Grant 62071116, and by National Key Research and Development Program of China under Grant 2021YFB2900404.

REFERENCES

[1] Q. Wu, F. Liu, H. Xia, and T. Zhang, "Semantic transfer between different tasks in the semantic communication system," in *IEEE Wirl. Commun. and Netw. Conf. (WCNC)*. Austin, TX, USA, Apr. 2022, pp. 566–571.

[2] C. Qi, P. Dong, W. Ma, H. Zhang, Z. Zhang, and G. Y. Li, "Acquisition of channel state information for mmwave massive MIMO: Traditional and machine learning-based approaches," *Sci. China Inf. Sci.*, vol. 64, no. 8, Aug. 2021, Art. no. 181301.

[3] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, May 2020.

[4] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, Apr. 2021.

[5] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Nov. 2020.

[6] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Jun. 2021.

[7] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acous. Spe. Sig. Process. (ICASSP)*. Calgary, AB, Canada, Apr. 2018, pp. 2326–2330.

[8] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, Sep. 2019.

[9] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 89–100, Nov. 2020.

[10] A. Babenko and V. Lempitsky, "The inverted multi-index," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1247–1260, Jun. 2015.

[11] D. Baranchuk, A. Babenko, and Y. Malkov, "Revisiting the inverted indices for billion-scale approximate nearest neighbors," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Munich, Germany, Sep. 2018, pp. 202–216.

[12] H. Noh, T. Kim, and J.-P. Heo, "Product quantizer aware inverted index for scalable nearest neighbor search," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*. Virtual, Oct. 2021, pp. 12 210–12 218.

[13] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jun. 2019.

[14] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm, "From single image query to detailed 3d reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Boston, Massachusetts, Jun. 2015, pp. 5126–5134.

[15] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[16] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.* Nice, France, Oct. 2003, pp. 1470–1477.