AN END-TO-END ADVERSARIAL HASHING METHOD FOR UNSUPERVISED MULTISPECTRAL REMOTE SENSING IMAGE RETRIEVAL

*Xuelei Chen, Cunyue Lu**

Department of Instrument Science and Engineering, Shanghai Jiao Tong University

ABSTRACT

Unlike natural images, remote sensing images are usually multispectral. And the lack of sufficient labeled data puts a limit on supervised learning for remote sensing image retrieval. In this paper, we propose a novel method for unsupervised multispectral remote sensing image retrieval. The proposed method makes use of the unsupervised representation learning ability of GAN. Meanwhile, a new reconstruction loss exploits the latent codes in GAN to make the final output informative and representative. Transfer learning and color histograms are used to generate an estimated similarity matrix to further guide the training. Hash constraints can make the output codes binary and compact. In the testing stage, the hash codes of multispectral images can be computed in an end-to-end manner. Experiments on a multispectral remote sensing image dataset, EuroSAT [1], show the superiority of the proposed method over other state-of-the-art methods.

***Index Terms*—** Image retrieval, remote sensing, GAN, learning to hash, unsupervised learning

1. INTRODUCTION

Remote sensing is the technology that uses aircraft- or satellite-based instruments to observe the objects on the earth by analyzing propagated signals, e.g., electromagnetic radiation. With the development of the remote sensing technology, more and more remote sensing images are produced every day. According to [2], Sentinel satellites which are operated by the European Space Agency can produce approximately 10 TB of Earth Observation (EO) data per day. To better manage and use large-scale remote sensing image data, it is necessary to develop efficient image retrieval methods.

Manual annotation of such big data is time-consuming and expensive. And different from natural images, remote sensing images have much lower spatial resolution, which makes the annotation even harder. The lack of sufficient labeled data can be an obstacle in applying supervised deep learning methods to remote sensing images. Unsupervised learning can be used to mitigate this problem. [3, 4, 5] show that generative adversarial networks can be used not only for modeling the data distribution (generating images), but also for unsupervised representation learning. Representation learning is closely related to image retrieval because the distance of representations can be used as the similarity measure. So, using GAN for unsupervised representation learning may be a practical solution for remote sensing image retrieval.

Remote sensing images have multiple bands. If we use transfer learning to extract the features from CNNs pretrained on other datasets, e.g., ImageNet dataset, only the information stored in RGB bands is used. To make use of all the information stored in all bands, we must find effective methods that can take multiple bands into consideration.

Conventional image retrieval methods use feature descriptors including color features[6, 7], texture features[8], shape features[9] and deep learning features[10, 11]. Image hashing is a more effective way to realize efficient and accurate image retrieval, because hash codes are compact and distance computation between hash codes is fast. Existing unsupervised image hashing methods [12, 13, 14, 15] can map high dimensional vectors to hash space and maintain the similarity or distance relationship in the original space. They require images being projected to feature vectors in advance. Newly proposed unsupervised deep hashing methods [16, 17] can compute hash codes from images in an end-to-end manner. They use rotation invariance or pre-trained CNN features to guide the hash learning. GAN-based hashing methods [18, 19, 20] use GAN as the network backbone and add hash and similarity constraints to the objective functions.

This paper proposes a novel method to address two problems: insufficient labels and multiple bands in remote sensing image data. The contributions are summarized as follows: (1) For the first time, GAN-based method is used for multispectral image retrieval. (2) The existing similarity matrix generation method is improved by considering non-RGB bands of multispectral images. A new reconstruction loss is proposed as part of the objective function. (3) Thorough experiments show the effectiveness of all components of our final method.

2. PROPOSED METHOD

In this section, detailed description of the proposed method is provided from four aspects: Network Design, Similarity Matrix Generation, Objective Functions and Image Retrieval.

The research is supported by NSFC-61371017, NSFC-11174206 and ASN-IF2015-1302. *Corresponding author.

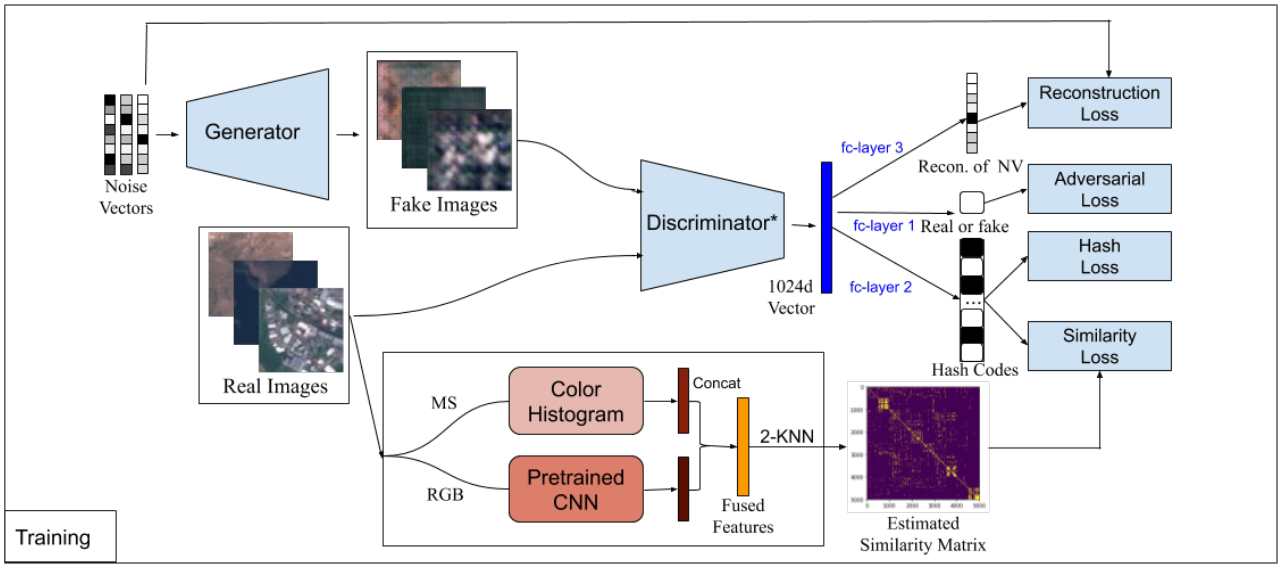

Fig. 1: Flowchart of the proposed method. Fake images and real images in the proposed method are all multispectral images. Only the RGB-bands of them are visualized in this flowchart. The * on the "Discriminator" denotes the discriminator without the final fully connected layer. Discriminator* 's output will be fed into three different fully connected layers.

2.1. Network Design

Our network is depicted in Fig. 1. The backbone is the vanilla GAN, which is made up of a generator and a discriminator. The generator generates fake images from random noise vectors to fool the discriminator. The discriminator learns to distinguish between real images and images generated from the generator.

The discriminator's penultimate layer's output is used as the feature representation of the input image, which is a 1024-dimension vector colored blue in Fig. 1. But this representation is not binary, because the activation function is Leaky ReLU. To overcome this problem, a new *fc-layer 2* is added after the penultimate layer of the discriminator. This *fc-layer 2* uses Tanh as the activation function to restrict the output values between -1 and 1. And it reduces the dimension at the same time. Hash loss and similarity loss work in this stream.

In addition, a novel reconstruction stream is added in the network. A new *fc-layer 3* is added after the penultimate layer of the discriminator. This stream aims to reconstruct latent codes of fake images. Reconstruction loss works in this stream.

2.2. Similarity Matrix Generation

Building an unsupervised similarity matrix to guide the hash learning is firstly proposed in [21, 20]. However, their method only uses RGB bands. We improve this method by also considering non-RGB bands of remote sensing images. As shown in the lower rectangle of Fig. 1, we use color histogram method [22] to process multispectral images and get a color

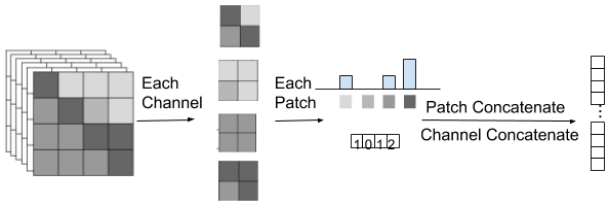

Fig. 2: Illustration of the local color histogram method used to extracte non-RGB features.

histogram feature. In more detail, we use local color histogram method as shown in Fig. 2. Each channel is divided into many small patches, and color histograms are created for these small patches. Color histograms from patches of all channels are concatenated to build the feature vector. The color histogram feature vector and CNN-extracted semantic feature vector are concatenated using (1):

$$feature_{fuse} = (feature_{RGB}, \eta * feature_{MS}) \quad (1)$$

where η is the weight value to balance possible scale difference of these two features. Then a two-stage kNN search process is conducted to generate the estimated similarity matrix. The first kNN search uses the cosine distance of the fused features as the distance metric. The first k_1 neighbors are regarded as similar images. The second kNN search uses the number of the different neighbors as the distance metric. The first k_2 neighbors are regarded as similar images. Two training images are regarded similar when they are similar in both kNN search. Then we get the final estimated similarity matrix for similarity loss.

2.3. Objective Functions

The generator and the discriminator are trained simultaneously. They are both updated at each iteration step. The joint objective functions are composed of four loss functions.

Adversarial Loss: As mentioned before, the backbone of our network is vanilla GAN. Adversarial loss makes the generator generate realistic images and the discriminator accurately distinguish between generated images and real images. The discriminator can learn good representations of the image in this adversarial training process. The adversarial loss functions for the generator and the discriminator are shown in (2) and (3).

$$L_{G_A} = -\mathbf{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(D(G(\mathbf{z})))] \tag{2}$$

$$L_{D_A} = -\mathbf{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] - \mathbf{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))] \tag{3}$$

where $\mathbf{x}$ denotes the real image, $\mathbf{z}$ denotes the random noise vector, $G(\mathbf{z})$ denotes the generated image from the noise vector $\mathbf{z}$, and $D(\cdot)$ denotes the *real or fake* prediction for a input image.

Hash Loss: Even though Tanh activation function in *fc-layer 2* can restrict each entry of the hash codes in the range $(-1, 1)$, some entries may lie in the middle of this range, which can hinder the accuracy of the image retrieval system. Hash loss will push the codes close to -1 and 1. This loss is formulated in (4):

$$L_h = -\mathbf{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}||H(\mathbf{x})||^{\mathbf{2}} \tag{4}$$

where $\mathbf{x}$ denotes the real image and $H(\mathbf{x})$ denotes the hash codes of $\mathbf{x}$. $H(\cdot)$ is made up of part of the discriminator and a hash-specific fully connected layer.

Similarity Loss: The estimated similarity matrix generated from Sec 2.2 is used in the similarity loss, which is formulated in (5). This loss can force the computed hash codes to maintain high-level similarity information in image pairs.

$$L_s = \mathbf{E}_{\mathbf{x}_i\sim p_{data}(\mathbf{x}),\mathbf{x}_j\sim p_{data}(\mathbf{x})}||\frac{H(\mathbf{x}_i) * H(\mathbf{x}_j)}{l} - S_{i,j}||^2 \tag{5}$$

where l denotes the hash code length. $\mathbf{x_i}$ and $\mathbf{x_j}$ denotes two random real images. $H(\mathbf{x_i})$ and $H(\mathbf{x_j})$ denotes the hash codes of $\mathbf{x_i}$ and $\mathbf{x_j}$. $S_{i,j}$ denotes the estimated similarity value of $\mathbf{x_i}$ and $\mathbf{x_j}$ obtained from two-stage kNN search.

Reconstruction Loss: The input of the generator is the latent codes that represent the data distribution in the latent space. From this perspective, the latent codes can be the informative and compact representations of the generated fake images. The aim of image hashing is to find binary representations that are informative and compact. So we can exploit these latent codes to guide the learning process in the discriminator. This idea is named as reconstruction loss and is formulated in (6):

$$L_r = \mathbf{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}||R(G(\mathbf{z})) - \mathbf{z}||^2 \tag{6}$$

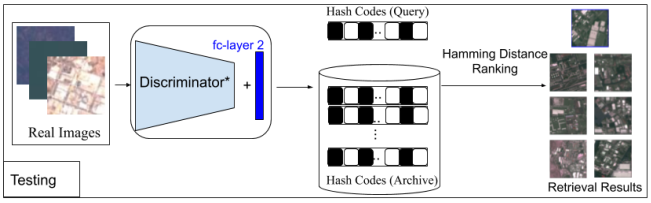

Fig. 3: Flowchart of image retrieval process.

where $R(G(\mathbf{z}))$ denotes the reconstructed codes of the fake image $G(\mathbf{z})$ that is generated from $\mathbf{z}$. $R(\cdot)$ is made up of part of the discriminator and a reconstruction-specific fully connected layer.

Combining all those four loss functions mentioned above, the joint loss functions for the generator and the discriminator are shown in (7) and (8):

$$L_G = L_{G_A} \tag{7}$$

$$L_D = L_{D_A} + w_1 * L_h + w_2 * L_s + w_3 * L_r \tag{8}$$

where w_1, w_2 and w_3 are hyperparameters.

2.4. Image Retrieval

After the whole network is trained, only part of the network is used to compute hash codes. Discriminator* and *fc-layer 2* form a complete hash model which takes images in and gives hash codes out. It's an end-to-end hashing method.

The Hamming distance is taken as the similarity measure between the hash codes of different images. It measures the number of different bits. The smaller distance corresponds to the higher similarity between the original images. This image retrieval process is illustrated in Fig. 3.

3. EXPERIMENTAL RESULTS

Thorough experiments using different methods are conducted on EuroSAT [1], which is a recently published multispectral remote sensing image dataset. The images in this dataset cover 10 land-use classes and have 13 bands with three different spatial resolutions of $10m$, $20m$ and $60m$ per pixel. Each class contains 2000-3000 images. In total, the dataset has 27000 images.

Only 10 bands whose spatial resolution is $10m$ per pixel or $20m$ per pixel are used in our experiments. Other three bands' spatial resolution is too low and they are used for detecting aerosols, water vapor, or cirrus in the atmosphere, which are not related to visual contents of ground objects. Bands with resolution of $20m$ per pixel are upsampled to $10m$ per pixel using cubic-spline interpolation. For the dataset splitting, 100 images are randomly selected from each class to form the query set. The remaining images form the archive set. Meanwhile, 500 images per class are randomly sampled in the archive set to build the training set.

Table 1: Evaluation of estimated similarity matrices using different methods

Method		Accuracy	IoU
RGB	ResNet-152	0.8761	0.0331
	Inception-v3	0.8899	0.1626
MS	Raw Color	0.8884	0.0853
	L. ColorHist.	0.8894	0.1071
Incep.-v3 + L. ColorHist		**0.8948**	**0.1628**

The batch size is 64. The learning rate is 0.0002. The β_1 and β_2 in Adam optimizer are 0.5 and 0.999. The α in Leaky ReLU is 0.2. The w_1 , w_2 and w_3 in joint loss (8) are 0.1, 0.1 and 0.1. The η used in (1) is 6. The k_1 and k_2 used for similarity matrix generation are 500 and 300.

3.1. Evaluation of Estimated Similarity Matrix

The accuracy of the estimated similarity matrix affects the discriminator's ability to generate representative hash codes through similarity loss. Therefore, it is important to generate an accurate estimated similarity matrix.

The estimated similarity matrix is generated only on 5000 training images. It is a binary matrix shaped 5000×5000. Each entry provides the estimated similarity value (1 or 0) of the corresponding training image pair. We evaluate estimated similarity matrices by comparing them with the ground truth using accuracy and IoU, as shown in Table 1.

For RGB bands, Inception-v3 achieves better results than ResNet-152. For multispectral bands, the local color histogram method performs much better than raw color method. When we fuse RGB features extracted from Inception-v3 and MS features obtained from local color histogram method, we can generate a much better similarity matrix than only considering one type of the feature.

3.2. Ablation Study

Three different base models are designed, each of which contains only part of the structure of our proposed network. They are named as CNN_MS, GAN_RGB and GAN_MS. Two extra modules are names as ESM and REC.

RGB or MS denotes using RGB bands or multispectral bands. ESM denotes using the estimated similarity matrix to guide the hash learning. It includes similarity loss and hash loss. REC denotes reconstructing latent codes of the fake image to guide the hash learning. It includes reconstruction loss.

CNN_MS comprises only discriminator* and *fc-layer 2*. It is trained using hash loss and similarity loss. GAN_RGB and GAN_MS both comprise the generator, discriminator* and *fc-layer 1*. An extra *fc-layer* between the 1024-dimension embedding and the final source prediction is added to restrict the penultimate embedding's length to be 32, 64 or 128. These two models are trained using only adversarial loss.

Table 2: MAP@100 of image retrieval using different methods

	Length	32	64	128
CNN_MS	(bin)	0.2752	0.2842	0.2720
GAN_RGB	(non-bin)	0.5184	0.5287	0.5480
GAN_MS	(non-bin)	0.5655	0.5915	0.6002
GAN_MS+REC	(non-bin)	0.5698	0.6169	0.6336
GAN_MS+ESM	(bin)	**0.6470**	0.6556	0.6785
Ours	(bin)	0.6151	**0.6581**	**0.6827**
SKLSH [12]	(bin)	0.1378	0.1435	0.1401
ITQ[13]	(bin)	0.4554	0.5016	0.5050
SpH [14]	(bin)	0.4447	0.4774	0.5118
SP [15]	(bin)	0.4085	0.4371	0.4715

The Mean Average Precision (MAP) of the first 100 retrieved images using different models are shown in Table 3. Comparing the results of CNN_MS with GAN_MS, we can see that the adversarial training can improve the retrieval performance greatly. Comparing the results of GAN_RGB with GAN_MS, we can see that GAN can extract much better codes for retrieval when using multispectral images. Comparing the results of GAN_MS with GAN_MS + ESM and GAN_MS+REC, we can see that the ideas of ESM and REC can both further improve the retrieval performance.

The final method (**Ours**) is actually GAN_MS+ESM+REC. When the code length is 32, REC degrades the final result a little bit. The reason for this is that length 32 is not sufficient to reconstruct the latent codes of fake images. The results of GAN_MS and GAN_MS+REC in length 32 also reflect this. The increase from 0.5655 to 0.5698 is very small.

3.3. Comparison with State-of-the-arts

The results of our method are also compared with other state-of-the-art methods. Four different methods are evaluated on the multispectral image retrieval task: SKLSH [12], ITQ[13], SpH [14] and SP[15]. These 4 methods map high dimensional vectors to binary space and maintain the similarity or distance relationship at the same time. The fused feature vector used for similarity matrix generation (Section 2.2) is the input vector of these four methods. From Table 3, we can see that our proposed method significantly outperforms other state-of-the-arts in all three code lengths.

4. CONCLUSION

In this paper, we addressed unsupervised hashing for multispectral image retrieval with a novel end-to-end method. Thorough experiments on a recently published multispectral image dataset, EuroSAT, proved the effectiveness of several important modules of the proposed method. And our methods shows great superiority over other state-of-the-arts.

5. REFERENCES

[1] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *arXiv preprint arXiv:1709.00029*, 2017.

[2] Pieter Kempeneers and Pierre Soille, "Optimizing sentinel-2 image selection in a big data context," *Big Earth Data*, vol. 1, no. 1-2, pp. 145–158, 2017.

[3] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[4] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[5] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell, "Adversarial feature learning," *arXiv preprint arXiv:1605.09782*, 2016.

[6] Michael J Swain and Dana H Ballard, "Color indexing," *International journal of computer vision*, vol. 7, no. 1, pp. 11–32, 1991.

[7] Greg Pass, Ramin Zabih, and Justin Miller, "Comparing images using color coherence vectors.," in *ACM multimedia*. Citeseer, 1996, vol. 96, pp. 65–73.

[8] Bangalore S Manjunath and Wei-Ying Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 8, pp. 837–842, 1996.

[9] Yong Rui, Alfred C She, and Thomas S Huang, "Modified fourier descriptors for shape representation-a practical approach," in *Proc of First International Workshop on Image Databases and Multi Media Search*. Citeseer, 1996, pp. 22–23.

[10] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus, "Deep image retrieval: Learning global representations for image search," in *European conference on computer vision*. Springer, 2016, pp. 241–257.

[11] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[12] Maxim Raginsky and Svetlana Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Advances in neural information processing systems*, 2009, pp. 1509–1517.

[13] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2012.

[14] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon, "Spherical hashing," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2957–2964.

[15] Yan Xia, Kaiming He, Pushmeet Kohli, and Jian Sun, "Sparse projections for high-dimensional binary codes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3332–3339.

[16] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2475–2483.

[17] Erkun Yang, Cheng Deng, Tongliang Liu, Wei Liu, and Dacheng Tao, "Semantic structure-based unsupervised deep hashing.," in *IJCAI*, 2018, pp. 1064–1070.

[18] Zhaofan Qiu, Yingwei Pan, Ting Yao, and Tao Mei, "Deep semantic hashing with generative adversarial networks," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 225–234.

[19] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi, Yanhua Yang, Cheng Deng, and Heng Huang, "Unsupervised deep generative adversarial hashing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3664–3673.

[20] Cheng Deng, Erkun Yang, Tongliang Liu, Wei Liu, Jie Li, and Dacheng Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Transactions on Image Processing*, 2019.

[21] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen, "Binary generative adversarial networks for image retrieval," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] Osman Emre Dai, Begüm Demir, Bülent Sankur, and Lorenzo Bruzzone, "A novel system for content-based retrieval of single and multi-label high-dimensional remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 7, pp. 2473–2490, 2018.

Supplemental material for a submission to ICIP 2020 with the title:

AN END-TO-END ADVERSARIAL HASHING METHOD FOR UNSUPERVISED MULTISPECTRAL REMOTE SENSING IMAGE RETRIEVAL

Xuelei Chen, Cunyue Lu

Department of Instrument Science and Engineering, Shanghai Jiao Tong University

1. DESCRIPTION

1.1. Network Architecture

The detailed architecture of our network is shown in 1

1.2. Dataset Visualization

Some sample images of EuroSAT dataset are visualized in Fig. 1

1.3. Results Visualization

1.3.1. Estimated Similarity Matrix

The estimated similarity matrices are visualized in Fig. 2. For better visualization, images in the training set are permuted according to their class. Images from the same class are placed in consecutive 500 entries of the training image list.

1.3.2. Image Retrieval

Image retrieval results of a sample query image are shown in Fig. 3

1.3.3. Precision-Recall Curve

Precision-Recall curves of images retrieval using different methods are shown in Fig. 4

1.3.4. Generated Image

In our proposed method, adversarial training is used mainly for unsupervised representation learning. The image generation ability of GAN can be regarded as a by-product. We visualize some generated images in Fig. 5.

Table 1: The detailed network architecture of the proposed method

Operation	Kernel	Strides	#Kernel	BN	Nonlinearity	Output
G - $(50,)$ input						
Linear + Reshape				✓	ReLU	$(4, 4, 512)$
Transposed Convolution	5×5	2×2	256	✓	ReLU	$(8, 8, 256)$
Transposed Convolution	5×5	2×2	128	✓	ReLU	$(16, 16, 128)$
Transposed Convolution	5×5	2×2	64	✓	ReLU	$(32, 32, 64)$
Transposed Convolution	5×5	2×2	c	✗	Tanh	$(64, 64, c)$
D* - $(64, 64, c)$ input						
Convolution	3×3	2×2	32	✓	Leaky ReLU	$(32, 32, 32)$
Convolution	3×3	1×1	64	✓	Leaky ReLU	$(32, 32, 64)$
Convolution	3×3	2×2	128	✓	Leaky ReLU	$(16, 16, 128)$
Convolution	3×3	1×1	256	✓	Leaky ReLU	$(16, 16, 256)$
Convolution	3×3	2×2	256	✓	Leaky ReLU	$(8, 8, 256)$
Flatten						$(8 \times 8 \times 256,)$
Linear				✓	Leaky ReLU	$(1024,)$
fc-layer 1 - Linear				✗	Sigmoid	$(1,)$ **Real/Fake**
fc-layer 2 - Linear				✓	Tanh	$(l,)$ **Hash Codes**
fc-layer 3 - Linear				✓	Tanh	$(50,)$ **Recon. of NV.**

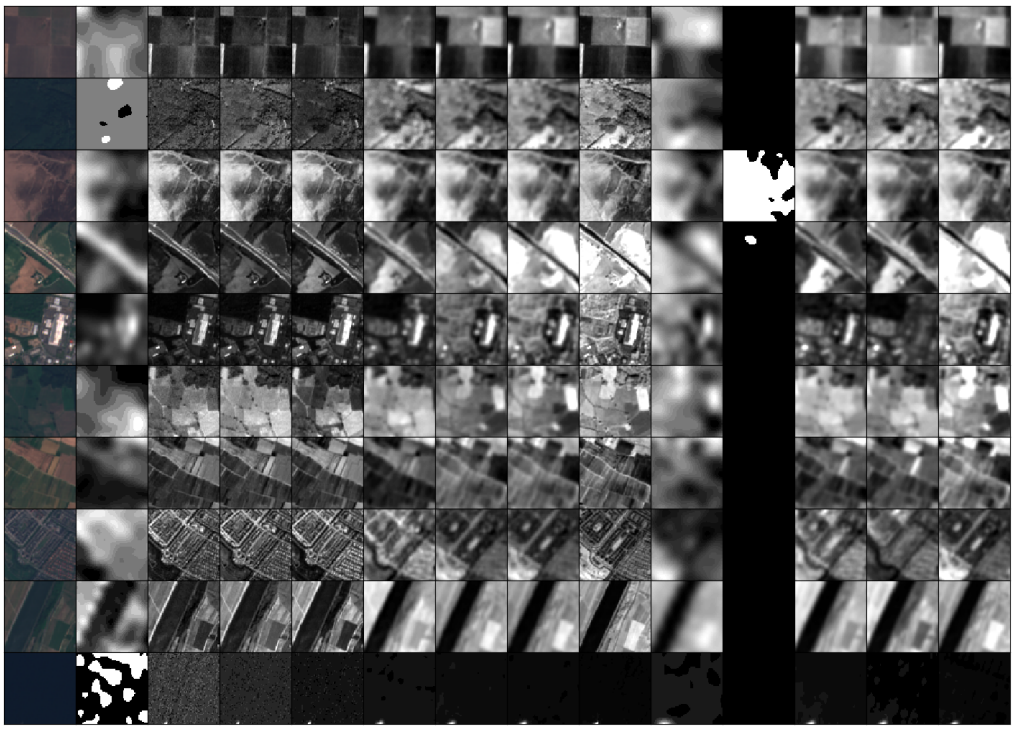

Fig. 1: Visualization of multispectral remote sensing images in EuroSAT. The first column shows RGB images and other columns show single spectral images from different bands. One image from each category is selected and visualized. From top to bottom are Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River and Sea Lake

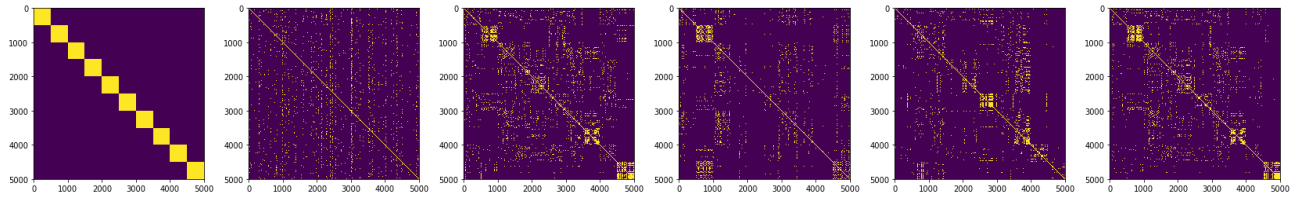

Fig. 2: Visualization of the estimated similarity matrices. The first graph is the ground truth. The second to the sixth graph are results using methods listed in Table 1 of the original paper.

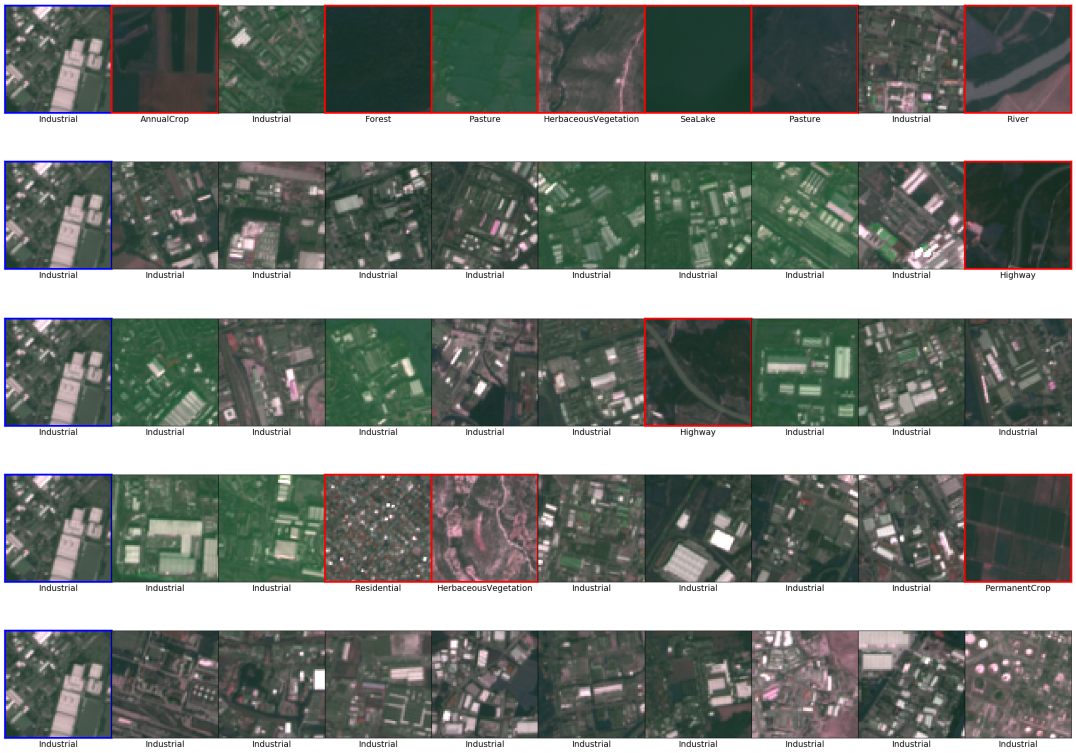

Fig. 3: Visualization of the retrieval results. The 1st, 5th, 10th, 15th, 20th, 25th, 30th, 35th and 40th retrieved images are shown. From top to bottom are results using SKLSH, ITQ, SpH, SP and our method. Code length used here is 64.

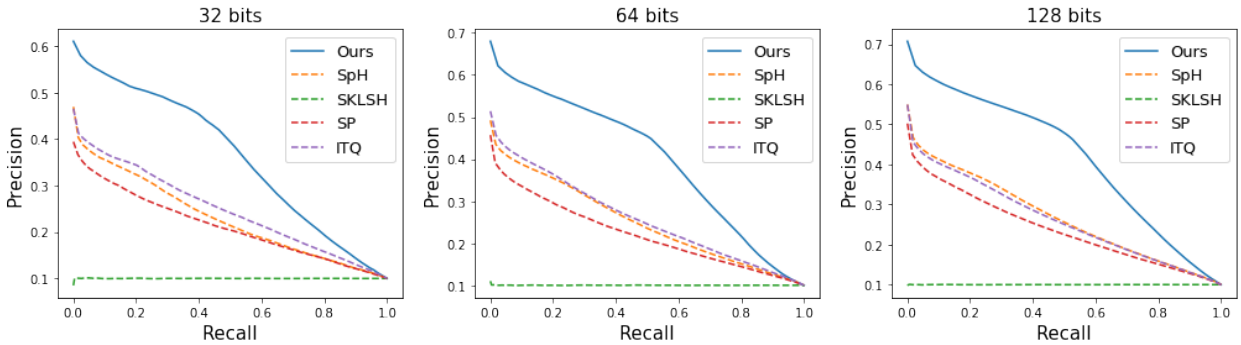

Fig. 4: Precision-Recall Curve of image retrieval using different methods and different code lengths.

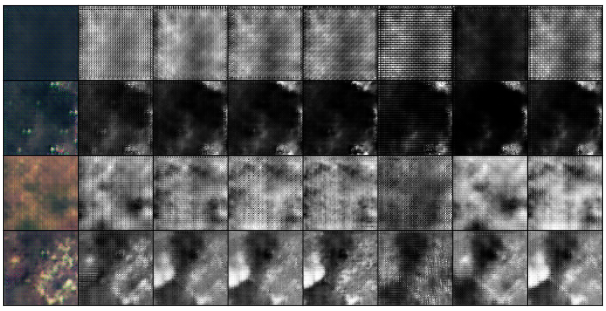

Fig. 5: Examples of generated multispectral images. RGB bands are visualized in the first column. From the second column to the eighth column are B05, B06, B07, B08, B08A, B11, B12.