Class-Specific Variational Auto-Encoder for Content-Based Image Retrieval

Mehdi Rafiei and Alexandros Iosifidis
Department of Electrical and Computer Engineering
Aarhus University, Aarhus, Denmark
{rafiei,ai}@ece.au.dk

Abstract—Using a discriminative representation obtained by supervised deep learning methods showed promising results on diverse Content-Based Image Retrieval (CBIR) problems. However, existing methods exploiting labels during training try to discriminate all available classes, which is not ideal in cases where the retrieval problem focuses on a class of interest. In this paper, we propose a regularized loss for Variational Auto-Encoders (VAEs) forcing the model to focus on a given class of interest. As a result, the model learns to discriminate the data belonging to the class of interest from any other possibility, making the learnt latent space of the VAE suitable for class-specific retrieval tasks. The proposed Class-Specific Variational Auto-Encoder (CS-VAE) is evaluated on three public and one custom datasets, and its performance is compared with that of three related VAE-based methods. Experimental results show that the proposed method outperforms its competition in both in-domain and out-of-domain retrieval problems.

Index Terms—Variational Auto-Encoder, Image Retrieval, Class-Specific Discriminant Learning

I. INTRODUCTION

CONTENT-Based Image Retrieval (CBIR) is the task of searching for images with similar content to that of a query image. Available methods for doing so can be broadly categorized into those based on feature engineering and those relying on deep learning models [1]. Methods belonging to the first category adopt in their first stage explicit feature extraction techniques (e.g. based on color [2], texture [3], and shape [4]), along with keypoint-based image descriptions and representations (e.g. the Scale Invariant Feature Transform (SIFT) [5] combined with Bag of Words (BoW) model [6], or the Fisher Vector (FV) representation [7]).

In recent years, deep learning models outperformed the more traditional methods in diverse computer vision tasks, including CBIR, with convolutional layers being widely used for the feature extraction [8–11]. Such data-driven feature extractors can be part of Convolutional Neural Network (CNN) models which are either pre-trained on large image datasets [10], e.g., like the ImageNet [12], or be fine-tuned on a target dataset usually leading to improved retrieval performance [11].

A type of deep learning models which is well-suited for CBIR is Variational Auto-Encoders (VAEs). Such models are trained to reconstruct their input, and the representation learnt in the latent space (output of the Encoder) can be used for CBIR. Traditional VAEs have limitations when applied in retrieval tasks, as they are not trained to discriminate between different classes forming the retrieval problem. To address

this issue, one can train or fine-tune the model based on other reconstruction losses. However, a balance needs to be kept between generative and discriminative properties of the adopted loss as highly discriminative data representations can lead to losing valuable information which is important for image retrieval, especially when retrieving images based on queries belonging to classes outside of those appearing in the training (out-of-domain retrieval) [13]. To address this problem, a regularized discriminative deep VAE method was proposed in [14] that models the latent generative factors for each of the training classes. Although this method shows good results in in-domain and out-of-domain image retrieval tasks, it is well-suited for multi-class retrieval problems and has limitations when it is applied to class-specific retrieval problems.

In this paper, we consider class-specific retrieval problems where one wants to retrieve images from a database based on a query image which either belongs to a class of interest, or not. In such a problem, the class of interest (called positive class) is usually well populated and all other images belong to multiple classes (forming the negative class of the class-specific problem) which are underpopulated, or their labels may not be available during the training phase. An example application which falls within this problem description is that of face retrieval where a specific person (e.g., the user of a system) is well-represented in the database while the rest of the facial images in the database can belong to other individuals with only a few (or even one) images. Following a multi-class formulation similar to that of [13] would lead to a binary problem where the negative class is modeled to be homogeneous, which generates problems as negative class images can exhibit very large variations.

To address this, we propose a training loss to train class-specific deep VAEs for CBIR. Instead of learning image representations capable to discriminate between different classes, the VAE is trained to discriminate between the class of interest (positive class) and any other available possibility (samples belonging to the negative class). This strategy encourages the model to learn data representations which leads to a homogeneous positive class in the latent space of the VAE and discriminates this class from the data belonging to any other classes forming the negative class. This would also help the model to discriminate the positive class from other classes unseen during training (out-of-domain retrieval). To

Fig. 1. Forces caused by reconstruction, KLD, and repulsive losses in a) RD-VAE, b) binary RD-VAE, and c) CS-VAE.

demonstrate the performance of the proposed method over different data types, it is extensively evaluated on three public and one custom datasets. In addition, for each dataset, two evaluation scenarios (namely in- and out-of-domain retrieval) are considered. The performance of the method is compared with three other VAE-based methods.

The rest of this paper is structured as follows. In Section II, related work on VAE-based image retrieval is briefly discussed. The proposed method is described in III. Experimental evaluations are presented in Section IV. Finally, the paper is concluded in Section V.

II. RELATED WORK

Several supervised and unsupervised VAE methods, such as VAE [15] and regularized discriminative VAE (RD-VAE) [14], have been proposed for CBIR tasks. However, these methods have limitations in class-specific CBIR applications. A VAE is commonly formed by an Encoder which receives an image as input and outputs a representation of that image in a (usually lower-dimensional) latent space, and a Decoder which receives this image representation as input and tries to reconstruct the input to the VAE image on its output. The parameters of both the Encoder and the Decoder are jointly optimized to preserve the maximum information when encoding and to have the minimum reconstruction error when decoding. By considering x_i, $i = 1, \ldots, N$ as the input images in the training set, this optimization is achieved by using the reconstruction loss and the Kullback–Leibler divergence:

$$L_{vae} = MSE_{loss} + \alpha_{\mathrm{KL}} KLD_{unsup}, \qquad (1)$$

$$KLD_{unsup} = \sum_{x_i} \big((x_i - \mu)^2 + \sigma^2 - \log(\sigma) - 1 \big). \qquad (2)$$

The Kullback–Leibler divergence term (KLD_{unsup}) forces the data representations in the latent space to form a Gaussian distribution having mean μ and variance σ. Using such an unsupervised learning process to train the VAE determines a Gaussian distribution that can be considered as the data distribution of all the training data (irrespectively of which class each training image may belong to) in the input of the Decoder. Thus, applying CBIR using image representations

coming from the latent space of a VAE usually leads to low performance.

To solve this limitation, a VAE-based regularized discriminative deep metric learning method (RD-VAE) was proposed in [14]. This method modifies the training loss of the VAE such that samples belonging to the same class form homogeneous clusters which are well-separated by clusters corresponding to other classes. To do that, KLD_{unsup} is replaced with a supervised KLD, and a repulsive term (rep_{loss}) is added to the loss function:

$$L_{\mathrm{rd\text{-}vae}} = MSE_{loss} + \alpha_{\mathrm{KL}} KLD_{sup} + rep_{loss}, \qquad (3)$$

$$KLD_{sup} = \sum_{x_i} \big((x_i - \mu_{l_i})^2 + \sigma_{l_i}^2 - \log(\sigma_{l_i}) - 1 \big), \quad (4)$$

$$rep_{loss} = {}^{1}\!/\!\rho \sum_{x_i} \sum_{x_j \neq x_i} \max \big(0, \rho - \| \mu_{l_i} - \mu_{l_j} \|_2^2 \big)^2, \quad (5)$$

where l_i denotes the class label of x_i, KLD_{sup} determines a Gaussian distribution for each class, and rep_{loss} forces the means of different class distributions away to be in a minimum distance of ρ from each other. Those data representations are also constrained by the MSE_{loss} loss, meaning that they need to preserve adequate input information in order to reconstruct the input image in the output of the Decoder. Figure 1-a shows a schematic 2D representation of RD-VAE at the latent space.

III. PROPOSED CLASS-SPECIFIC VAE

As described above, the class-specific image retrieval task is defined as the task of retrieving images based on a query image which either belongs to a class of interest, or not. One could approach this problem by applying the RD-VAE method described above, i.e., to consider the class of interest as the positive class and form a negative class including all images belonging to all other classes (binary RD-VAE, Figure 1-b). Since all VAE models determine the data representations in the latent space through training, one would assume that (using an adequately high number of network parameters and extensively tuning them) it is possible to force all training images forming the negative class (despite their possibly high variations) to form a homogeneous cluster in the latent space. However, this approach leads to increasing the complexity of

the model and may not be able to generalize well on unseen (test) data.

The Class-Specific VAE (CS-VAE) introduces a new KLD term and a new repulsive term in the training loss of the VAE:

$$L_{\text{cs-vae}} = MSE_{loss} + \alpha_{\text{KL}} KLD_{sup}^{cs} + rep_{loss}^{cs}, \qquad (6)$$

$$KLD_{sup}^{cs} = \sum_{x_i \in l_p} \left((x_i - \mu_{l_p})^2 + \sigma_{l_p}^2 - \log(\sigma_{l_p}) - 1 \right), \qquad (7)$$

$$rep_{loss}^{cs} = 1/\rho \sum_{x_i \notin l_p} \max\left(0, \rho - \| x_i - \mu_{l_p} \|_2^2\right)^2. \qquad (8)$$

Optimizing the loss function in Eq. (6) forces the image representations in the latent space of the positive class to form a Gaussian distribution, defined by mean μ_{l_p} and variance σ_{l_p}. Moreover, the image representations in the latent space of the data belonging to the negative class are forced to be far away (with a minimum distance of ρ) from the mean of the positive class. Those data representations are also constrained by the MSE_{loss} loss, meaning that they need to preserve adequate input information in order to reconstruct the input image in the output of the Decoder.

Figure 1-c shows a schematic 2D representation of CS-VAE at the latent space. As can be seen in that Figure, one would expect that such image representations in the latent space can have some favourable properties for class-specific CBIR. Representations of images belonging to the class of interest (positive class) are learnt to lay close to each other in the latent space (i.e., to be similar to each other, meaning that properties of those images in common are expected to be highlighted). Representations of images not belonging to the class of interest are forced to be well-discriminated in relation to the class of interest and they are not forced to group together, i.e., they are allowed to lay anywhere in the latent space leading to representations of dissimilar input images of the negative class to be far away from each other. This can lead to better-preserving properties of images belonging to the negative class and higher performance.

It should be noted that class-specific optimization criteria have also been used in the past for determining data representations based on linear [16, 17] and kernel-based [18–21] Class-Specific Discriminant Analysis. Those methods commonly optimize a class-specific variant of the Rayleigh Quotient [22, 23] which tries to minimize the in-class scatter to out-of-class scatter ratio in the projection space. Contrary to this approach, the proposed CS-VAE exploits the KLD and the repulsive terms in Eqs. (7) and (8) which, in combination to the reconstruction error at the output of the Decoder, lead to the optimization problem in Eq. (6) which is well-suited for training deep learning models.

IV. EXPERIMENTS

To evaluate the performance of the proposed CS-VAE method, we conducted experiments on four datasets. We used the following three publicly available datasets:

Fig. 2. Example images from the Fashion MNIST dataset.

Fig. 3. Example images from the Cifar-10 dataset.

- *Fashion MNIST* [24]: It includes 70,000 fashion-related gray-scale images with resolution of 28×28 pixels. The images belong to 10 classes, with 7,000 images per class. 60,000 images form the training set and the remaining 10,000 the test set. Figure 2 shows example images from the dataset.
- *Cifar-10* [25]: It includes 60,000 RGB-color images with resolution of 32×32 pixels. The images belong to 10 classes, with 6,000 images per class. 50,000 images form the training set and the remaining 10,000 the test set. Figure 3 shows example images from the dataset.
- *Yale* [26]: It includes 165 gray-scale facial images of 15 different subjects (11 images per subject). The images have 320×243 pixels resolution and are taken under different lighting and facial expressions. Figure 4 shows example images from the dataset.

To also evaluate the performance of the proposed method

Fig. 4. Example images from the Yale dataset.

Fig. 5. Example images from the X-ray dataset.

on a problem coming from an industrial application, we also used an X-ray image dataset of fibrous products. This dataset was collected over an X-ray test on several defective and non-defective fibrous product samples. The dataset contains four classes including a Non-defective (ND) class and the following three defective classes:

- **D1:** the drops of melted raw materials that are not converted to fibers successfully;
- **D2:** binder bulks that are not evenly distributed over the fibers;
- **D3:** a collection of several small shots of molten raw materials close together.

Samples of these four classes are shown in Figure 5. In total, 271 gray-scale images with a resolution of 244×244 pixels are available in this dataset. In our experiments, only the ND class is considered as the class of interest, since this approach resembles a real-life anomaly detection problem where the defects need to be distinguished from the non-defective class in order to retrieve the non-defective products.

A. In-domain CBIR experiments

To evaluate the performance of the proposed method in CBIR based on query images belonging to classes included in the training, we conducted experiments following the in-domain experimental protocol. We compare the performance of CS-VAE with that of three other VAE-based methods, i.e., VAE, RD-VAE, and binary RD-VAE. Since VAE and RD-VAE methods are trained based on unsupervised and multi-class optimization problems, respectively, one model is trained on each dataset and the 11-recall point-based Average Precision (AP) metric [27] is calculated for each class separately, as well as the mean AP (mAP) over all classes. For Binary RD-VAE and CS-VAE, a model is trained for each class-specific problem by considering the corresponding class as the class of interest, and AP is calculated for that class. We also calculated the mAP of all class-specific models. We repeated the experiments five times and we reported the mean and standard deviation of AP values for all experiments.

For hyper-parameter selection on the experiments in Fashion MNIST and Cifar-10 datasets, the training set is randomly split into 80%/20% training/validation subsets, and the values of the hyper-parameters of all models are selected based on their performance on the validation set. Due to the small number of samples per class in Yale and the X-ray datasets, we performed 5-fold cross-validation, where the performance of each model having different hyper-parameter values is evaluated as the average AP over all folds.

Considering Equations (1) - (8), hyper-parameter selection is applied to select values for ρ and α_{KL}, and for determining the model architecture. We use a grid search strategy to select the values of ρ and α_{KL} for each dataset and each model using the ranges $\rho = \{1, \ldots, 10\}$ and $\alpha_{\mathrm{KL}} = \{0.1, \ldots, 10\}$. For instance, the mAP values over all classes on the Fashion MNIST dataset are shown in Figure 6. It can be seen from Figure 6-a, b, and c that there is a linear relation between the hyper-parameters value and the models' performance. However, for CS-VAE, the optimal values of these two hyper-parameters are in the range of the selected intervals.

The selected model parameters, including the model's input size, encoder and decoder architecture, the size of latent space, and the two hyper-parameter values (ρ and α_{KL}), for all datasets and methods are shown in Table I. It can be seen that to create the models' encoders, 2D convolution layers, batch normalization layers, and ReLU activation functions are used, followed by flattening and two parallel linear layers to have means and variances for each dimension of the latent space. For the decoders, after linear and unflattening layers, 2D transposed convolution layers, batch normalization layers, and ReLU activation functions are used, followed by a sigmoid activation function to construct the output image.

Results: The performance (AP%) of each method on each class of Fashion-MNIST, as well as the mAP over all classes, are reported in Table II. From this table, it can be seen that RD-VAE achieved much higher performance in comparison to the VAE model. Such an improvement was expected due to the separate Gaussian distributions defined by RD-VAE for each class and the use of the repulsive loss to push them away from each other in the latent space. The binary RD-VAE reached a slightly lower performance compared to RD-VAE. As it was explained in Section III, such a lower performance is expected as a result of the model forcing all negative image representations to form a homogeneous cluster in the latent space, despite their possibly high variations in the input space. Finally, CS-VAE managed to achieve the highest performance.

Figures 7-a and b illustrate the data representation obtained by applying Principal Component Analysis on the image representations in the latent spaces of the binary RD-VAE and CS-VAE and keeping the top three eigenvectors. It can be seen that for the binary RD-VAE, all samples belonging to the negative class are pushed to cluster one side of the positive class. CS-VAE, as also mentioned in Section III, allows the representations of the negative images in the latent space to freely be arranged, as long as they are adequately far away (parameterized by the value of ρ) from the positive class' mean.

The performance of all competing methods on the Cifar-10 dataset is reported in Table III. As can be seen, similar observations to those made for the results on Fashion MNIST

Fig. 6. Performance (mAP%) on Fashion MNIST (in-domain retrieval) for different values of ρ and α_{KL}: a) VAE, b) RD-VAE, c) binary RD-VAE, and d) CS-VAE

TABLE I
MODEL AND SELECTED HYPERPARAMETER VALUES FOR EACH METHOD AND DATASET.

	Fashion MNIST	CIFAR-10	Yale	X-ray
Model	**Input:** $1\times28\times28$	**Input:** $3\times32\times32$	**Input:** $1\times244\times244$	**Input:** $1\times244\times244$
	Encoder:	**Encoder:**	**Encoder:**	**Encoder:**
	C2d(1, 16, k=3, s=1)	C2d(3, 32, k=3, s=1)	C2d(1, 32, k=3, s=3)	C2d(1, 32, k=3, s=3)
	BN2d(16), ReLU()	BN2d(32), ReLU()	BN2d(32), ReLU()	BN2d(32), ReLU()
	C2d(16, 32, k=3, s=2)	C2d(32, 64, k=3, s=2)	C2d(32, 64, k=3, s=3)	C2d(32, 64, k=3, s=3)
	BN2d(32), ReLU()	BN2d(64), ReLU()	BN2d(64), ReLU()	BN2d(64), ReLU()
	C2d(32, 64, k=3, s=1)	C2d(64, 128, k=3, s=1)	C2d(64, 128, k=3, s=3)	C2d(64, 128, k=3, s=3)
	BN2d(64), ReLU()	BN2d(128), ReLU()	BN2d(128), ReLU()	BN2d(128), ReLU()
	C2d(64, 128, k=3, s=2)	C2d(128, 256, k=3, s=2)	Flatten()	Flatten()
	BN2d(128), ReLU()	BN2d(256), ReLU()	Linear(in=10368, out=256)	Linear(in=10368, out=256)
	Flatten()	Flatten()	BN1d(256), ReLU()	BN1d(256), ReLU()
	Linear(in=2048, out=256)	Linear(in=6400, out=1024)	Linear(in=256, out=30)	Linear(in=256, out=10)
	BN1d(256), ReLU()	BN1d(1024), ReLU()	Linear(in=256, out=30)	Linear(in=256, out=10)
	Linear(in=256, out=30)	Linear(in=1024, out=30)		
	Linear(in=256, out=30)	Linear(in=1024, out=30)		
	Decoder:	**Decoder:**	**Decoder:**	**Decoder:**
	Linear(in=30, out=256)	Linear(in=30, out=1024)	Linear(in=30, out=256)	Linear(in=10, out=256)
	BN1d(256), ReLU()	BN1d(1024), ReLU()	BN1d(256), ReLU()	BN1d(256), ReLU()
	Linear(in=256, out=2048)	Linear(in=1024, out=6400)	Linear(in=256, out=10368)	Linear(in=256, out=10368)
	BN1d(2048),, ReLU()	BN1d(6400), ReLU()	BN1d(10368), ReLU()	BN1d(10368), ReLU()
	UnFlatten()	UnFlatten()	UnFlatten()	UnFlatten()
	CT2d(128, 64, k=3, s=3)	CT2d(256, 128, k=3, s=2)	CT2d(128, 64, k=3, s=3)	CT2d(128, 64, k=3, s=3)
	BN2d(64), ReLU()	BN2d(128), ReLU()	BN2d(64), ReLU()	BN2d(64), ReLU()
	CT2d(64, 32, k=3, s=2)	CT2d(128, 64, k=3, s=1)	CT2d(64, 32, k=3, s=3)	CT2d(64, 32, k=3, s=3)
	BN2d(32), ReLU()	BN2d(64), ReLU()	BN2d(32), ReLU()	BN2d(32), ReLU()
	CT2d(32, 16, k=3, s=1)	CT2d(64, 32, k=3, s=2)	CT2d(32, 1, k=4, s=3)	CT2d(32, 1, k=4, s=3)
	BN2d(16), ReLU()	BN2d(32), ReLU()	Sigmoid()	Sigmoid()
	CT2d(16, 1, k=2, s=1)	CT2d(32, 16, k=3, s=1)		
	Sigmoid()	BN2d(16), ReLU()		
		CT2d(16, 3, k=(4, 4), s=1)		
		Sigmoid()		
VAE	$\alpha_{kl}=10$	$\alpha_{kl}=10$	$\alpha_{kl}=5$	$\alpha_{kl}=2$
RD-VAE	$\rho=10$	$\rho=10$	$\rho=10$	$\rho10=$
	$\alpha_{kl}=10$	$\alpha_{kl}=10$	$\alpha_{kl}=10$	$\alpha_{kl}=10$
Binary RD-VAE	In-domain: $\rho=10$	In-domain: $\rho=10$	In-domain: $\rho=10$	In-domain: $\rho=10$
	Out-of-domain: $\rho=10$	Out-of-domain: $\rho=10$	Out-of-domain: $\rho=10$	Out-of-domain: $\rho=10$
	In-domain: $\alpha_{kl}=10$	In-domain: $\alpha_{kl}=10$	In-domain: $\alpha_{kl}=10$	In-domain: $\alpha_{kl}=10$
	Out-of-domain: $\alpha_{kl}=10$	Out-of-domain: $\alpha_{kl}=10$	Out-of-domain: $\alpha_{kl}=10$	Out-of-domain: $\alpha_{kl}=10$
CS-VAE	In-domain: $\rho=1$	In-domain: $\rho=2$	In-domain: $\rho=10$	In-domain: $\rho=10$
	Out-of-domain: $\rho=1$	Out-of-domain: $\rho=5$	Out-of-domain: $\rho=10$	Out-of-domain: $\rho=5$
	In-domain: $\alpha_{kl}=1$	In-domain: $\alpha_{kl}=10$	In-domain: $\alpha_{kl}=5$	In-domain: $\alpha_{kl}=5$
	Out-of-domain: $\alpha_{kl}=2$	Out-of-domain: $\alpha_{kl}=2$	Out-of-domain: $\alpha_{kl}=5$	Out-of-domain: $\alpha_{kl}=10$

C2d: 2D convolution, CT2d: 2D transposed convolution, BN2d: 2D Batch Normalization, BN1d: 1D Batch Normalization.

| | AP% (mean± std) | | | |
	VAE	RD-VAE	Binary RD-VAE	CS-VAE
C1	48.48±0.33	89.32±0.46	82.77±2.10	**89.94±1.08**
C2	65.14±1.61	96.78±0.19	96.55±2.34	**97.50±0.32**
C3	34.96±0.88	**89.20±0.66**	83.72±1.28	89.14±1.25
C4	42.22±0.42	91.22±0.35	87.87±0.53	**92.72±0.39**
C5	36.20±0.41	87.48±0.18	81.02±0.52	**90.16±0.84**
C6	47.04±2.56	96.86±0.29	97.67±0.23	**98.02±0.11**
C7	28.68±0.22	76.78±0.55	68.15±0.82	**82.22±0.74**
C8	56.50±0.99	96.62±0.30	97.37±0.17	**97.56±0.30**
C9	45.08±3.31	96.82±0.07	**97.50±0.23**	97.44±0.23
C10	54.08±2.51	96.52±0.19	96.45±0.25	**96.76±0.16**
Mean	45.83	91.75	89.02	**93.14**

Fig. 7. Image representations of Fashion MNIST obtained by applying PCA on the latent space (three eigenvectors): a) binary RD-VAE, and b) CS-VAE

can be made here too.

By observing the results reported for the Yale dataset in Table IV it can be seen that, although CS-VAE still achieves the highest performance, the binary RD-VAE outperforms the RD-VAE method. This can be due to lower variations in the images belonging to the negative class.

Experimental results on the X-ray dataset are reported in Table V. As described in Section IV, on this dataset only the ND class is considered as the class of interest and, therefore, only one model is trained for each method and the AP%s values of all methods are reported. Similarly to Fashion MNIST and Cifar-10, CS-VAE achieved the highest performance followed by RD-VAE.

| | AP% (mean± std) | | | |
	VAE	RD-VAE	Binary RD-VAE	CS-VAE
C1	19.10±0.32	71.62±0.55	59.57±2.35	**76.95±3.48**
C2	15.17±0.21	77.77±0.87	68.47±3.85	**83.92±2.98**
C3	17.07±0.16	59.67±0.53	45.77±0.31	**63.07±3.52**
C4	13.50±0.12	53.07±0.72	40.12±1.14	**58.95±0.26**
C5	18.62±0.13	63.05±0.68	49.95±0.81	**69.22±3.01**
C6	14.77±0.19	62.10±0.50	47.42±0.92	**66.57±2.41**
C7	19.90±0.25	74.12±0.10	67.17±1.97	**81.17±2.99**
C8	15.50±0.14	69.57±1.09	61.87±0.86	**77.00±1.70**
C9	19.45±0.27	78.32±0.95	71.07±0.92	**85.75±2.15**
C10	16.22±0.14	75.15±1.34	62.47±3.67	**81.32±1.86**
Mean	16.93	68.44	57.39	**74.39**

| | AP% (mean± std) | | | |
	VAE	RD-VAE	Binary RD-VAE	CS-VAE
C1	72.20±1.75	98.21±1.98	99.12±1.09	**99.20±1.16**
C2	74.40±0.74	98.18±1.39	98.92±0.92	**99.20±0.91**
C3	74.46±1.20	99.52±2.47	**100.0±0.00**	**100.0±0.00**
C4	73.32±1.49	99.33±0.38	**99.94±0.08**	99.84±0.16
C5	73.50±1.11	99.21±1.08	**99.98±0.04**	99.96±0.08
C6	73.45±0.98	99.03±0.84	99.52±0.66	**99.94±0.04**
C7	72.68±0.74	98.97±1.89	**99.56±0.34**	98.48±1.57
C8	74.45±1.97	98.12±0.33	98.90±1.24	**99.64±0.53**
C9	76.62±2.05	98.30±1.60	98.44±1.48	**99.64±0.43**
C10	75.91±1.98	98.52±0.67	99.78±0.34	**99.96±0.08**
C11	77.01±1.40	99.35±0.81	**99.72±0.24**	99.06±1.17
C12	77.21±0.70	99.01±0.32	99.34±1.12	**99.90±0.15**
C13	72.23±0.41	99.15±1.09	**99.82±0.14**	98.70±1.08
C14	74.05±0.76	98.63±0.84	98.10±1.90	**99.44±0.30**
C15	73.05±1.12	99.24±1.60	**99.50±0.63**	99.00±1.42
Mean	74.30	98.85	99.37	**99.46**

| | AP% (mean± std) | | | |
	VAE	RD-VAE	Binary RD-VAE	CS-VAE
ND	39.05±0.54	75.27±3.03	68.30±1.20	**94.50±1.69**

B. Out-of-domain CBIR experiments

To evaluate the performance of the proposed method in retrieving images belonging to classes that are not present in the training phase, we also conducted out-of-domain experiments. To do this, only half of the available classes in the dataset are used to form the negative class in the training phase, which is selected randomly. For testing, we used all available classes in the dataset to form the negative class. Therefore, before splitting the training data into training and validation sets, half of the classes (excluding the class of interest) are randomly discarded. Then, the same experimental protocols and hyper-parameter selection processes as in the in-domain experiments are used. The selected hyper-parameters are shown in Table I.

Results: The performance of binary RD-VAE and CS-VAE on all datasets is reported in Table VI. In all cases, we see an accuracy drop compared to the in-domain experiments. However, this performance drop is higher for the binary RD-VAE. This shows that the proposed CS-VAE is more capable in learning better latent space representations for data belonging in unseen during training classes.

| | mAP% | |
	Binary RD-VAE	CS-VAE
Fashion MNIST	71.39	**80.19**
Cifar-10	54.02	**69.32**
Yale	97.85	**98.59**
X-ray	71.55	**93.85**

V. Conclusion

In this paper, a variant of the VAE was proposed which is suited for class-specific content-based image retrieval. The proposed method models the optimization problem of the VAE to determine a latent space in which the class of interest is well-discriminated by samples belonging to any other class. To do this and for preserving as much information as possible in order to achieve a good reconstruction in its output, the model learns a discriminative data representation using KLD and repulsive losses forcing the data belonging to the class of interest to form a Gaussian distribution, while forcing all other samples far away from the mean of this distribution. This method was extensively evaluated over several public and custom datasets on both in-domain and out-of-domain retrieval tasks. Three related VAE-based methods were used for comparisons and the proposed method outperformed them in all cases.

Acknowledgment

References

[1] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, "Deep learning for instance retrieval: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.

[2] K. M. Alam, N. Siddique, and H. Adeli, "A dynamic ensemble learning algorithm for neural networks," *Neural Computing and Applications*, vol. 32, no. 12, pp. 8675–8690, 2020.

[3] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188–198, 2013.

[4] M. Singha and K. Hemachandran, "Content based image retrieval using color and texture," *Signal & Image Processing: An International Journal*, vol. 3, no. 1, pp. 39–57, 2012.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *IEEE International Conference on Computer Vision*, vol. 2, pp. 1470–1477, 2003.

[7] Y. Uchida, S. Sakazawa, and S. Satoh, "Image retrieval with fisher vectors of binary features," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 4, pp. 326–336, 2016.

[8] J. Pan, X. Zhu, and P. Liu, "Generating adaptive targeted adversarial examples for content-based image retrieval," *International Joint Conference on Neural Networks*, pp. 1–9, 2022.

[9] S. Hamreras, B. Boucheham, M. A. Molina-Cabello, R. Benítez-Rochel, and E. López-Rubio, "Dynamic selection of classifiers for content based image retrieval," *International Joint Conference on Neural Networks*, pp. 1–8, 2021.

[10] S. Pang, J. Ma, J. Xue, J. Zhu, and V. Ordonez, "Deep feature aggregation and image re-ranking with heat diffusion for image retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1513–1523, 2018.

[11] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[13] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[14] N. Passalis, A. Iosifidis, M. Gabbouj, and A. Tefas, "Variance-preserving deep metric learning for content-based image retrieval," *Pattern Recognition Letters*, vol. 131, pp. 8–14, 2020.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] D. T. Tran, M. Gabbouj, and A. Iosifidis, "Multilinear class-specific discriminant analysis," *Pattern Recognition Letters*, vol. 100, pp. 131–136, 2017.

[17] A. Iosifidis, "Probabilistic class-specific discriminant analysis," *IEEE Access*, pp. 183847–183855, 2020.

[18] G. Goudelis, S. Zafeiriou, A. Tefas, and I. Pitas, "Class-specific kernel discriminant analysis for face verification," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 570–587, 2007.

[19] A. Iosifidis, A. Tefas, and I. Pitas, "Class-specific reference discriminant analysis with application in human behavior analysis," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 3, pp. 315–326, 2015.

[20] A. Iosifidis and M. Gabbouj, "Class-specific kernel discriminant analysis revisited: Further analysis and extensions," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4485–4496, 2017.

[21] K. Li and G. Wu, "Randomized approximate class-specific kernel spectral regression analysis for large-scale face verification," *Machine Learning*, vol. 111, no. 6, pp. 2037–2091, 2022.

[22] A. Martinez and A. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[23] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[24] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a

novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical Report, Toronto, ON, Canada*, 2009.

[26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *European Conference on Computer Vision*, vol. 19, no. 7, pp. 43–58, 1996.

[27] H. Schütze, C. D. Manning, and P. Raghavan, "Introduction to information retrieval," *Cambridge University Press*, vol. 39, pp. 234–265, 2008.