Deep Learning of Image Representations with Convolutional Neural Networks Autoencoder for Image Retrieval with Relevance Feedback

An Hong Son[1], Nguyen Huu Quynh[2], Dao Thi Thuy Quynh[3], Cu Viet Dung[2]

[1] Science Management Department, Viet - Hung Industrial University, Hanoi, Vietnam
[2] Faculty of Information Technology, ThuyLoi University, Hanoi, Vietnam
[3] Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

Correspondence: Dao Thi Thuy Quynh (quynhdtt@ptit.edu.vn)
Communication: received 04 Aug 2022, revised 18 Oct 2022, accepted 30 Dec 2022
DOI: 10.32913/mic-ict-research.v2023.n1.1063

***Abstract*: Image retrieval with traditional relevance feedback encounters problems: (1) ability to represent hand-crafted features which is limited, and (2) inefficient with high-dimensional data such as image data. In this paper, we propose a framework based on very deep convolutional neural network autoencoder for image retrieval, called AIR (Autoencoders for Image Retrieval). Our proposed framework allows to learn feature vectors directly from the raw image and in an unsupervised manner. In addition, our framework utilizes a hybrid approach of unsupervised and supervised to improve retrieval performance. The experimental results show that our method gives better results than some existing methods on the CIFAR-100 image set, which consists of 60,000 images.**

Keywords: *image representations, CNNs, autoencoder, relevant feedback.*

I. INTRODUCTION

Content-based image retrieval aims to find similar images through image content analysis. Hence image representations and similarity measurements become important to it. In order that image retrieval is robust to geometrical and visual changes, the similarity between images is calculated based on the content of the images. The content of the images such as color, texture, shape, etc. is represented in the form of a feature descriptor [1]. The similarity between the feature vectors of the corresponding images is referred to as the similarity between the images. Therefore, the performance of any content-based image retrieval method also depends on the representation of the image's features. Any feature representation method is expected to be discriminatory, strong, and low-dimensional. Many feature representation methods have been studied to calculate the similarity between two images for content-based image retrieval. Feature representation using visual cues of manually selected images based on need [2–6]. These approaches are also referred to as hand-crafted feature descriptions. Moreover, in general, these methods are unsupervised learning because they do not need data to design the feature representation method. Hand-crafted features for image retrieval is a very active research area. However, its performance is limited because the hand-designed features cannot represent the image properties in a precise way [7].

For the last decade, we have seen the shift of feature representation from manual design to a learning-based approach, especially with the emergence of deep learning [8, 9]. In this shift, learning based on convolutional neural networks has replaced traditional hand-designed feature representation. Deep learning is a technique for learning abstract features from data that are important for applications and data sets [10]. Depending on the type of data to be processed, different architectures have emerged such as artificial neural networks, multilayer perceptrons [11]. Convolutional Neural Networks (CNNs) for image data [12, 13] and Recurrent Neural Networks (RNNs) for time series data [14]. Progress has been made for image retrieval using the power of deep learning [15, 16].

Deep learning progressions are supervised and would be difficult to apply to a problem with a small number of labeled samples such as an image retrieval problem with relevance feedback. In this problem, the number of user's feedback samples is quite limited. From the perspective of unsupervised deep learning, Hinton and Krizhevsky [17] proposed an autoencoder algorithm with application for image retrieval, which was then used for a number of other tasks such as face alignment [18]. When we train an autoencoder deep neural network, it does not require

labeled samples. An autoencoder can be thought of as a multilayer sparse coding network. Each node in the autoencoder network can be viewed as a prototype of the object image. From the bottom to the top layer, the prototype contains rich semantic information and becomes a better representation. After the autoencoder network is learned, the weights obtained by image reconstruction are based on the prototype, which are used as features for image retrieval and matching. Because the autoencoder can adaptively learn features when training, it can get an extremely good for image retrieval.

However, the image retrieval methods, using the above autoencoder, face the problem of poorer feature discriminability. The reason is models are often trained for classification while image retrieval needs to learn features for matching. Besides, these methods also lose information due to feature quantification [19]. Furthermore, deep neural networks suffer from the problem of vanishing/exploding gradients and computational complexity. Because autoencoders have multiple convolutional and deconvolutional layers, information is lost and performance is degraded when reconstructing images.

To overcome the above limitations, in this paper, we propose a semi-supervised framework based on convolutional neural network autoencoder for image retrieval with relevance feedback (AIR). This framework overcomes two problems: (1) the ability to distinguish the poor features of the previous methods because we integrate the relevance feedback mechanism and ranking via the SVM support vector machine, and (2) solve the problem of vanishing/exploding gradients and computational complexity through the use of shortcut connections in the autoencoder architecture and make it possible to use very deep autoencoders.

This article is organized as follows. We briefly review related work in section II. Section III presents our proposed method. Finally, the experimental results are described in section IV. Conclusions are made in section V.

II. RELATED WORK

Through supervised learning, data is passed from the input to the top layer for prediction. By minimizing the value of the cost function between the target value and the predicted value, the back-propagation algorithm is used to optimize the parameters that connect each pair of layers. Specifically, CNN [12] is a neural network-based transform, which is used to represent features through supervised learning. CNN is often performed in image analysis, speech recognition [20] and text analysis,.... Especially in image analysis, CNN has achieved great success such as face recognition [21], scene analysis [22], cell segmentation [23], and brain injury segmentation [24, 25].

In unsupervised learning approaches, unlabeled data is used to learn features, while a small amount of labeled data is used to tune the parameters, such as the Restricted Boltzmann Machine (RBM) [26], Deep Belief Network (DBN) [27], autoencoders [28] and stacked autoencoders [29]. Kumar et al have proposed an autoencoder approach for unsupervised feature learning [30]. Kalleberg et al. proposed a convolutional autoencoder approach to image analysis [31]. Li et al. have designed an RBM-based approach for classification [32].

Autoencoders were developed to learn efficient features for image content representation. It exploits a neural network to learn the representations of a given sample in order to minimize the reconstruction error. Feature learning with unsupervised learning algorithms that reconstruct input samples based on predefined rules. Autoencoder [33] can learn representative features to reconstruct input samples with minimal reconstruction error. Autoencoders are utilized to combine sounds (audio) and lyrics for musical mood classification [34]. Autoencoders and their variants have also been applied to multimodal representation learning [35, 36]. The authors in [17] proposed an autoencoder network to learn the latent representation between textual and visual content, minimizing the correlation error between the latent representations of the two methods. The authors in [37, 38] took advantage of a denoising autoencoder to learn representative features in an unsupervised way and applied it to train dominant detection models from raw image data. However, these methods face the problem of poorer discriminant ability of features because models are often trained for classification while image retrieval needs to learn features for matching. Besides, these methods also lose information due to feature quantification [19]. In addition, these methods do not take advantage of the very deep architecture of neural networks and their training process converges slowly.

III. PROPOSED METHOD

Figure 1 shows the proposed framework. Our method has three components. The first component is the unsupervised training of a very deep neural network autoencoder on a subset of the image set. The second component is to apply the learning model from the first component encoder to extract low-dimensional features from the database image set. Note here that both the first and second components are taken offline. The third component is to retrieve images that are similar to the query image based on the relevance feedback. The very deep convolutional neural network

model autoencoder is trained on a subset of the database image set. In this case, we use the CIFAR-100 image set.

1. Learning image representations with a very deep convolutional neural network autoencoder

This section describes the very deep convolutional neural network architecture autoencoder and training parameters.

A supervised approach is available for data-driven feature learning, when the connection weights are updated through a back-propagation algorithm. Compared with the supervised learning approach, the unsupervised learning approach can directly receive unlabeled input data, reducing the labor for labeling. Autoencoder extracts output data to reconstruct input data, and compares input data with original input data. After a number of iterations, the value of the cost function reaches the optimal level, which means that the reconstructed input data can approximate the original input data.

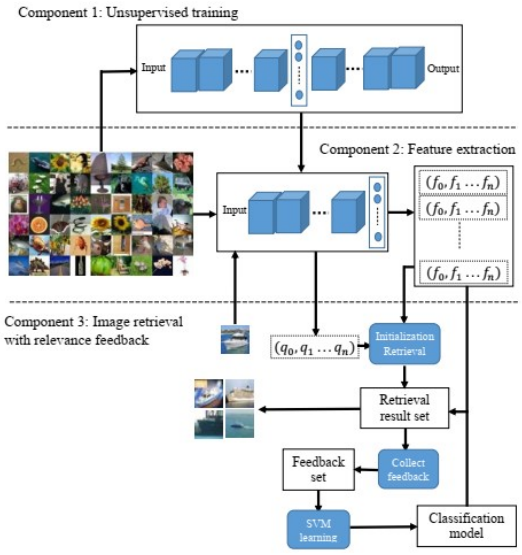

Figure 1. Proposed framework for image retrieval.

- Convolutional neural network autoencoder:

The convolutional neural network autoencoder combines locally connected convolution with the Standard autoencoder, which is a simple operator to add reconstruction input to the convolution operator. The procedure that converts the convolution from the feature map input to the output is called a convolution decoder. Then, the output values are reconstructed through the inverse convolution operator, which is called a convolution encoder. Furthermore, through the unsupervised greedy training autoencoder, the parameters of the encoding and decoding operators can be calculated. In the autoencoder convolution operator, $f(\cdot)$ represents the convolutional encoding operator and $f'(\cdot)$ represents the convolutional decoding operator. The input feature maps $p \in \mathbb{R}^{n\times l\times l}$, which are obtained from the input layer or the previous layer. It contains n feature maps, and the size of each feature is $l \times l$ pixels. The autoencoder convolution operator consists of m convolution kernels, and the output layer generates m feature maps. When input feature maps are generated from the input layer, n represents the number of output feature maps from the previous layer. The size of the convolution kernel is $d \times d$ with $d \le l$.

Let $\theta = \{W, \widehat{W}, b, \hat{b}\}$ represent the parameters of the autoencoder convolution layer, which need to be learned. Where, $W = \{w_j, j = 1, 2, \ldots, m\}$ and $b \in \mathbb{R}^m$ represent the parameters of the convolutional encoder, where $w_j \in \mathbb{R}^{n\times l\times l}$ is defined as a vector $w_j \in \mathbb{R}^{nl^2}$. Besides, $\widehat{W} = \{\hat{w}_j, j = 1, 2, \ldots, m\}$ and $\hat{b}$ represent the parameters of the convolutional decoder, here $\hat{b} \in \mathbb{R}^{nl^2}$, and $w_j \in \mathbb{R}^{1\times nl^2}$.

First, the input image is encoded so that each time a patch of $d\times d$ pixels $p_i, i = 1, 2, \ldots, k$, is selected from the input image, and then the weight w_j of the convolutional j used for convolution calculations. Finally, the neuron value $a_{ij}, j = 1, 2, \ldots, m$, is calculated from the output layer

$$a_{ij} = f(p_i) = \sigma(w_j \cdot p_i + b). \tag{1}$$

In equation (1), σ is a nonlinear activation function, in this paper, we use Rectified Linear Function (RElu)

$$\mathrm{RElu}(p) = \begin{cases} p & \text{if } p \ge 0 \\ 0 & \text{if } p < 0. \end{cases} \tag{2}$$

Then the o_{ij} output from the convolution decoder is encoded that p_i is reconstructed through a_{ij} to produce $\hat{p}_i$

$$\hat{p}_i = f'(a_{ij}) = \phi(w_i \cdot a_{ij} + \hat{b}), \tag{3}$$

$\hat{p}_i$ is generated after each convolution encoding and decoding. We get the patch P obtained from the reconstruction operator. We use the mean square error between the original patch of the input image $p_i, i = 1, 2, \ldots, p$ and the reconstructed patch of the image $\hat{p}_i, i = 1, 2, \ldots, k$ as cost function. In addition, the cost function is described in equation (4), and the reconstruction error is described in equation (5)

$$L(\theta) = \frac{1}{k}\sum_{i=1}^{k} E(p_i, \hat{p}_i), \tag{4}$$

$$E(p_i, \hat{p}_i) = \|p_i - \hat{p}_i\|^2 = \|p_i - \phi(\sigma(p_i))\|^2. \tag{5}$$

- Pooling layer:

Similar to in CNN, the convolution layer is connected to the pooling layer [12]. In the convolutional neural network

architecture autoencoder, the max pooling layer is placed after the convolution layer

$$a_j^i = \max(p_j^i). \tag{6}$$

Each input feature map is divided into n non-intersecting regions according to the size of the pooling region. In equation (6), p_j^i represents the i^{th} region of the j^{th} feature map, and a_j^i represents the i^{th} neuron of the j^{th} feature map. The number of input feature maps is equal to the number of output feature maps in the pooling layer. The neurons in the feature map can be reduced after performing the pooling operator, so the computational complexity is also reduced. In the experiment of this paper, we empirically evaluate the autoencoder architecture including the setting with and without the pooling layer.

- The proposed convolutional network architecture autoencoder:

As we all know, deep neural networks suffer from vanishing/exploding gradients problems and computational complexity. Because autoencoders have multiple convolutional and deconvolutional layers, information is lost and performance is degraded when reconstructing images. Inspiring by the residual neural network consisting of shortcut connections [13], we add shortcut connections to the autoencoder network as shown in Figure 2. These connections make it possible to send feature maps from the first layer of the encoder to several later layers directly. The reason for using sortcut connection is because: first, when the network is too deep, image details can be lost, while feature maps passed through sortcut connections carry a lot of image detail. Second, the sortcut connections make the training process of deep neural networks to converge faster [13]. In addition, using sortcut connections obtains the benefit of feature extraction for image retrieval through increasing the depth of the network.

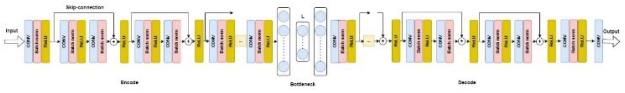

Figure 2. Proposed network architecture autoencoder for feature extraction.

- Training parameters:

Through the stochastic gradient descent (SGD) algorithm, the errors are minimized, and the autoencoder convolution layer is optimized. Finally, the trained parameters are used to generate feature maps that are passed to the next layer.

We use 50,000 unlabeled samples to train the convolutional neural network autoencoder through unsupervised learning at the convolution layer, the gradient is calculated through the cost function in (4), and the parameters are optimized by SGD. Each batch has a minimum of 150 samples, and the number of iterations per batch is 20. The number of channels is set up in equation (2) for the convolutional encoder and equation (3) for the corresponding convolutional decoder.

2. Retrieve images with relevance feedback using support vector machine

- Support Vector Machine:

In this paper, we choose a support vector machine (SVM) [39] for image ranking. The reason for choosing SVM is: first, it is a powerful classifier, especially for binary classifier, and the image retrieval problem with relevance feedback is a two-class problem. Second, through the found optimal hyperplane, we can use the distance from each sample to the optimal hyperplane as the value to rank the images.

- Image retrieval:

As the framework shown in Figure 1, after training the convolutional neural network model autoencoder in Component 1, we proceed to remove the decoder part and keep the encoder part to have the learning model as in Component 2. Use the learning model in Component 2 of the framework for the extraction of low-dimensional feature vectors to obtain a set of n feature vectors $(f_0, f_1, \ldots, f_n)$.

During the retrieval process as in Component 3 of the framework, the user provides a query image q, the feature vector of the query image will be passed the encoder learning model to obtain the feature vector of the query image $(q_0, q_1, \ldots, q_n)$. The initial retrieval process will compare (using Euclidean distance) the vector of the query image with the vector of each database image to obtain the retrieval result set. On this result set, the user selects relevance and unrelevance images to get the feedback set (this feedback set includes samples with negative and positive labels, they are also training samples). SVM learning is applied on the training set to obtain the SVM classification model. Apply the classification model on the feature vector set of the image database: the predicted positively labeled images that have the furthest distance from the optimal hyperplane will be ranked at the number one of the resulting list. Along with that, images that are predicted to have positive labels, which are the second furthest from the optimal hyperplane will be ranked at number two of the resulting list,... This process repeats until the user stops responding.

IV. EXPERIMENT RESULTS

1. Dataset

Image dataset CIFAR-100[1]: This dataset is a subset of 80-million tiny images. It contains 60,000 color images and the images in this set are grouped into 100 classes (600 images per class). The size of each image is 32×32. In our experiment, 10,000 images are taken as a set of query images. This query image set is generated by randomly selecting 100 images from each of the 100 classes. The remaining 50,000 images of CIFAR-100 were used as training set.

2. Evaluation Metrics

Average precision (AP): refers to the coverage below the precision-recall curve. A larger AP implies a higher precision-recall curve and better retrieval precision. AP can be calculated as follows:

$$AP = \frac{\sum_{k=1}^{N} P(k)\text{rel}(k)}{R}, \tag{7}$$

where R represents the number of relevance results for the query image from a total of N images. $P(k)$ is the precision of k retrieved images, and $\text{rel}(k)$ is an index function that has a value of 1 if the k^{th} image in the ranking list is relevant and 0 otherwise. Mean average precision (mAP) is accepted for the evaluation on all query images.

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q), \tag{8}$$

where Q is the number of query images.

3. The results on the image dataset CIFAR-100

To demonstrate the effectiveness of the proposed autoencoder architecture for image retrieval, we experiment with image retrieval by Euclidean distance, called Baseline, as follows:

At the beginning of the retrieval, the Euclidean distance in 128 dimensions is used to rank the images in the database. The reason that we use 128-dimensional feature vector is the appropriate dimension of many image retrieval methods [40, 41]. The results of the initial access (when there is no feedback) according to the different depths of the network on the set CIFAR-100 are shown in Figure 3 below. Figure 3 shows the mAP in five different configurations for the 10, 20, 40, and 60 layers of the network. The first configuration, called `classic`, is an autoencoder that uses the pooling layer and has no sortcut connection. The second configuration, called `Sortcut(con-decon)`, is an autoencoder using the pooling layer, and has a sortcut connection, which is a symmetric connection [42]. The third configuration, called `NoP_sortcut(con-decon)`, is an autoencoder that does not use the pooling layer and has a sortcut connection, and it is also a symmetric connection. The fourth configuration, called `Sortcut`, is an autoencoder that uses the pooling layer and has a sortcut connection but it is not a symmetric connection. The final configuration, called `NoP_sortcut`, is an autoencoder that does not use the pooling layer and has a sortcut connection, and it is also not symmetric (see Figure 2). The reason we experimented with configurations of network architectures with both pooling and without pooling because we wanted to test performance on images as small as those in the CIFAR-100 (32×32).

Looking at Figure 3, we see that the optimal number of layers of the autoencoder network architecture for image retrieval on the CIFAR-100 set (with all architectures) is 40 layers. Also from this Figure 3, we see that the network configuration using the pooling layer is effective on deeper network architectures.

The network architecture in [42] (including the `Sortcut(con-decon)` and `NoP_sortcut(con-decon)` configurations) for precision is lower than that of our network architecture (including `Sortcut` and `NoP_sortcut`). The reason is that although both the `Sortcut(con-decon)` and `NoP_sortcut(con-decon)` configurations of the network architecture in [42] use a sortcut connection, they use a symmetric sortcut connection. Network architectures using symmetric sortcut connections are more suitable for image noise removal than image retrieval. Out of the 5 configurations, two in our network architecture give the best results for all layers. This proves that the integration of asymmetric shortcut connection in autoencoder has effectively generated autoencoder deep networks for image matching.

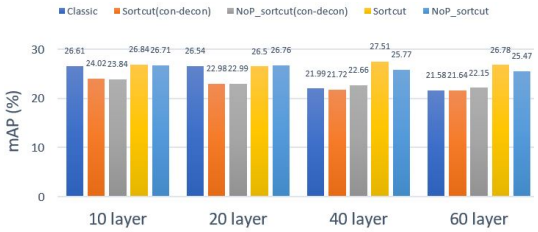

Figure 3. Results of image retrieval at different depths of autoencoder network.

Basing on the above experiment, we see that the optimal number of layers for autoencoder network architecture with

[1]https://www.cs.toronto.edu/~kriz/cifar.html

sortcut connection is 40. To demonstrate the effectiveness of the proposed framework for image retrieval, we test the framework with relevance feedback on this network configuration as follows:

After the user provides relevant feedback, the Baseline, AIR, EDSSCIR (Encoder-Decoder with Symmetric Skip Connection for Image Retrieval) in [42], and SSCAIR (Feature extraction using self-supervised convolutional autoencoder for content) [43] methods are applied to rerank the images in the database. We choose EDSSCIR method for performance comparison, it uses feature learned from convolution autoencoder with symmetric skip connection, because we want to demonstrate the efficiency of feature learned by our method in image retrieval.

Figure 4 shows the Mean average precision of the four methods (including Baseline (Non-RF), AIR, EDSSCIR, and SSCAIR) for the first three feedback iterations. From Figure 4, we see that the Baseline method gives the lowest precision. The reason is the Baseline method has no learning mechanism, it only calculates the Euclidean distance between the feature vector of the query image and that of the database image. Our AIR method gave better results than the other three methods on all iterations. The performance of the AIR method is significantly better than that of Baseline, which indicates that the relevant feedback provided by the user is very helpful in improving retrieval performance. AIR performs better than EDSSCIR because our method obtains a good feature representation.

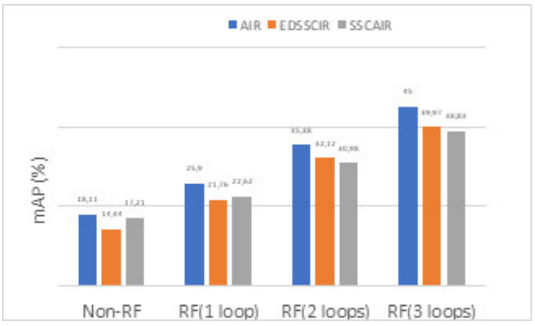

Figure 4. Performance comparison (in mAP) of the three methods for the first three iterations.

V. CONCLUSION

In this paper, we have presented an effective framework for image retrieval. This framework overcomes two problems: first, the poor discriminating ability of the existing methods, and second, mitigating the problem of vanishing/exploding gradients and computational complexity. The very deep convolutional neural network model autoencoder is utilized to learn efficient feature representations for image retrieval through the use of sortcut connections in the autoencoder architecture. This learning model is used to generate feature representations of database images. On the basis of these feature representations, we designed a relevance feedback learning mechanism using a support vector machine to take advantage of labeled samples from user's feedback. The training samples, which were obtained from the relevance feedback mechanism, were fed to the SVM classifier which enhanced the ability to learn the discriminant features that are used for retrieval. As a result of this framework, we have obtained good quality ranked lists, which both overcome the shortage of labeled samples and take advantage of very deep neural networks.

The experimental results performed on the CIFAR-100 set with 60,000 images have proved that our proposed framework produces results with higher accuracy than some current methods.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2020.10.

REFERENCES

[1] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," *Information retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

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

[3] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[4] S. R. Dubey, S. K. Singh, and R. K. Singh, "Rotation and illumination invariant interleaved intensity order-based local descriptor," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5323–5333, 2014.

[5] I. J. Jacob, K. Srinivasagan, and K. Jayapriya, "Local oppugnant color texture pattern for image retrieval system," *Pattern Recognition Letters*, vol. 42, pp. 72–78, 2014.

[6] D. Song and D. Tao, "Biologically inspired feature manifold for scene classification," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 174–184, 2009.

[7] G. Sumbul, J. Kang, and B. Demir, "Deep learning for image search and retrieval in large remote sensing archives," *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, pp. 150–160, 2021.

[8] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[9] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] C. H. Dagli, *Artificial neural networks for intelligent manufacturing*. Springer Science & Business Media, 2012.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 2067–2075.

[15] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 157–166.

[16] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2017.

[17] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval." in *ESANN*, vol. 1. Citeseer, 2011, p. 2.

[18] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European conference on computer vision*. Springer, 2014, pp. 1–16.

[19] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2687–2704, 2021.

[20] M. Chen, P. Zhou, and G. Fortino, "Emotion communication system," *IEEE Access*, vol. 5, pp. 326–337, 2016.

[21] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1891–1898.

[22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2012.

[23] H. Su, Z. Yin, S. Huh, T. Kanade, and J. Zhu, "Interactive cell segmentation based on active and semi-supervised learning," *IEEE transactions on medical imaging*, vol. 35, no. 3, pp. 762–777, 2015.

[24] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[25] A. Patel, S. C. van de Leemput, M. Prokop, B. van Ginneken, and R. Manniesing, "Automatic cerebrospinal fluid segmentation in non-contrast ct images using a 3d convolutional network," in *Medical Imaging 2017: Computer-Aided Diagnosis*, vol. 10134. SPIE, 2017, pp. 522–527.

[26] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[27] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[28] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, 2006.

[29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[30] D. Kumar, A. Wong, and D. A. Clausi, "Lung nodule classification using deep features in ct images," in *2015 12th conference on computer and robot vision*. IEEE, 2015, pp. 133–138.

[31] M. Kallenberg, K. Petersen, M. Nielsen, A. Y. Ng, P. Diao, C. Igel, C. M. Vachon, K. Holland, R. R. Winkel, N. Karssemeijer *et al.*, "Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1322–1331, 2016.

[32] Q. Li, W. Cai, and D. D. Feng, "Lung image patch classification with automatic feature learning," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 6079–6082.

[33] C. C. Tan, "Autoencoder neural networks: A performance study based on image recognition, reconstruction and compression," Ph.D. dissertation, Multimedia University, 2008.

[34] H. Xue, L. Xue, and F. Su, "Multimodal music mood classification by fusion of audio and lyrics," in *International Conference on Multimedia Modeling*. Springer, 2015, pp. 26–37.

[35] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593, 2014.

[36] M. Chen, K. Weinberger, F. Sha, and Y. Bengio, "Marginalized denoising auto-encoders for nonlinear representations," in *International conference on machine learning*. PMLR, 2014, pp. 1476–1484.

[37] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via sdaes," *IEEE transactions on cybernetics*, vol. 46, no. 2, pp. 487–498, 2015.

[38] X. Liu, M. Wang, Z.-J. Zha, and R. Hong, "Cross-modality feature learning via convolutional autoencoder," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1s, pp. 1–20, 2019.

[39] D. Tao, X. Tang, X. Li, and Y. Rui, "Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 716–727, 2006.

[40] Y. Chen, X. Lu, and X. Li, "Supervised deep hashing with a joint deep network," *Pattern Recognition*, vol. 105, p. 107368, 2020.

[41] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, "Unsupervised semantic-preserving adversarial hashing for image search," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4032–4044, 2019.

[42] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, 2016.

[43] I. A. Siradjuddin, W. A. Wardana, and M. K. Sophan, "Feature extraction using self-supervised convolutional autoencoder for content based image retrieval," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, 2019, pp. 1–5.

An Hong Son is a PhD student at the Academy of Science and Technology, Viet Nam Academy of Science and Technology. He works at the Science Management Department, Viet - Hung Industrial University. He graduated with a bachelor's degree in Informatics in 2002, a master's degree in Computer Science in 2008. His research interests include image processing, content-based image retrieval, and machine learning.
Phone: 0912.355219
E-mail: sonanhongvh@gmail.com

Nguyen Huu Quynh received Associate Professor in 2015. Since 2019, he worked at the Faculty of Information Technology, the Thuyloi university. He published over 30 articles in international journals and conferences, which included dozens of published in journals with SCIE indexing.
Research area: learning machine, computer vision, deep neural networks.
Email: quynhnh@tlu.edu.vn

Dao Thi Thuy Quynh received doctor in 2020. Since 2018, she has worked at the Faculty of Information Technology, Posts and Telecommunications Institute of Technology. Research area: learning machines, computer vision, deep neural networks.
Email: quynhdtt@ptit.edu.vn

Cu Viet Dung is a PhD student in Graduate University of Sciences and Technology. He work at the Faculty of Information Technology of Thuyloi University. He graduated from Electric Power of University in 2012 with an engineering degree; Master degree on software engineering from University of Engineering and Technology Ha Noi in 2014. His main research topics include image processing, Content based image retrieval, deep learning.
Email: dungcv@tlu.edu.vn