Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **2**, June **2025**

Using graph convolutional networks to improve accuracy of image retrieval with relevance feedback method

Thanh Nguyen Van

Lecturer at Academy of Finance, Hanoi, Vietnam
Ph.D. Student at Faculty of Information Technology, Thuyloi University, Hanoi, Vietnam
nguyenvanthanh@hvtc.edu.vn

Quynh Nguyen Huu*

CMC University, Hanoi, Vietnam
nhquynhml@gmail.com

*Corresponding author: Quynh Nguyen Huu

Received February 5, 2025, revised May 3, 2025, accepted May 7, 2025.

Abstract. *The greatest challenge for content-based image retrieval (CBIR) systems is reducing the semantic gap in image representation to increase retrieval accuracy. Many CBIR methods use information from users to reduce the semantic gap such as relevance feedback methods. However, the accuracy of image retrieval methods with relevance feedback has been limited because the sample set obtained from the feedback process is very small and unbalanced. Recently, deep learning models based on graph neural networks have known as a effective method and have achieved remarkable results. Among them, graph convolutional networks (GCN) models have achieved high efficiency in classification tasks with semi-supervised learning. To address these challenges of the image retrieval methods with relevance feedback, we propose a method using* ***G****raph* ***C****onvolutional* ***N****etworks to improve accuracy of* ***I****mage* ***R****etrieval with* ***R****elevance* ***F****eedback method called* ***GCN_IRRF****. Our proposed method: (1) taking advantage of user's feedback; (2) automatically adding training samples for SVM learning model through graph convolutional neural networks; and (3) leverage the power of deep features in image retrieval. Extensive experiment on the CIFAR100 dataset, Corel-1K dataset show that the accuracy of our method is much better than conventional feedback methods.*
Keywords: content-based image retrieval; graph convolutional networks; relevance feedback; SVM

1. **Introduction.** CBIR is the process of automatically retrieving images by extracting low-level features as: texture, color, shape,... or high-level features as: number and activity of objects, content of event that represented in the image, or any other useful features for retrieval taken from the image itself [1]. Some image retrieval methods combine both image features and image description text for efficient retrieval [2]. The accuracy of CBIR systems depends on the extraction/selection of features to compare the similarity between the features of the query and the features in the database [1]. The biggest challenge of CBIR is to reduce semantic gap in representing images according to their features. This semantic gap often exists due to the discrepancy between the feature representation of image and information perceived from image by the human vision system (HVS).

There are many image complex feature extraction methods are but the retrieval results are only based on the similarity of features without any information about human perception. This leads to the semantic gap problem as presented. To overcome this, the relevance feedback method has been widely studied and applied to narrow the gap between the low-level image features that machine learning can analyze and the high-level perception ability of human vision [3]. In general, the relevance feedback process is applied in many image retrieval systems to improve the query performance and user satisfaction. Relevance feedback algorithms provide interaction between the CBIR system and the user, allowing the user to give feedback on whether the returned results are relevant to the query image or not. Based on these feedback results, the similarity measure is updated to recalculate for the next retrieval result set. However, labeling large datasets is difficult because users often cannot do it manually many times because labeling can take a lot of time and effort, especially labels that require deep expertise. To overcome this limitation, this study proposed a graph-based semi-supervised learning method to automatically increase the number of labeled samples. Our experimental results are performed on the CIFAR-100 dataset and Corel-1K dataset show that the effectiveness of our proposed method is much better than previous relevance feedback-based image retrieval method.

2. **Related Work.**

2.1. **Image retrieval with SVM and relevance feedback.**

SVM is a machine learning method for binary classification with many advantages such as using an overfitting mechanism independent of the number of samples and their effectiveness with sparse data, SVM is used in various fields as object detection, pattern recognition, computer vision,... In computer vision field, many research works using and improving SVM for CBIR problem. In [4], Dongping Tian et al. have comprehensively analyzed the application of SVM for image retrieval. Pighetti et al. [5] combined multi-objective interactive genetic technique and SVM to improve the accuracy of more accurate retrieval of less annotated images by combining image features and user ratings. In [6], active learning SVM and particle swarm optimization (PSO) model is proposed for CBIR, in which PSO uses or does not use features to regularize auxiliary features and parameters in the SVM technique. In addition, the SVM operation is used to actively select most of the image feautures to minimize the results that are not relevant to the query image. In [7], Zhang et al. present a novel CBIR method based on active learning SVM that combines selection model with active learning to accelerate the convergence of the SVM classifier. Wang et al. [8] introduce a novel CBIR using SVM-based active feedback with a set of single-class SVM classifiers are separately pretrained, weight vectors of the sub SVM classifiers are dynamically calculated. Wang et al. later also proposed another CBIR method with SVM [9] using weighted kernel function and probabilistic feature . In [10], Truong-Giang Ngo et al. proposed an SVM-based active learning method for CBIR with relevance feedback, which significantly improved the retrieval accuracy, but the execution time of each iteration was quite large. Related works mainly perform semi-supervised learning independently on datasets with few labeled samples to predict labels for large numbers of unlabeled samples or to perform data classification, no research work has taken advantage of the performance of semi-supervised learning on graphs to supplement automatically training data samples for SVM for image retrieval.

2.2. **Graph-based semi-supervised learning.**

Semi-supervised learning (SSL) is a machine learning method with great practical value because it is capable of exploiting both labeled and unlabeled data samples.

A group of effective SSL methods that work on complex non-linear data, which are represented as graphs to infer label information of unlabeled vertices from other labeled vertices of the graph, are called Graph-based Semi-supervised Learning (GSSL) methods. GSSL methods have recently demonstrated their power and performance in many fields due to their flexible data representation, popularity, and scalability to big data. Gong et al. [11] proposed an SSL method based on Laplacian Deformable Graph (DGL) with the corresponding DGL Label Prediction algorithm (LPDGL). Yang et al. in [12] proposed a random walk-based GSSL method, called Planetoid, in which the representation of a node in the training embedding space is used to predict the class label in a given graph. A popular and effective GNN variant of the basic GNN was proposed by Kipf et al. [13] using regularization techniques that aggregate information from neighboring nodes including the node itself for semi-supervised learning to classify the node, called GCN, and then a large number of GCN variants to improve SSL performance from various aspects were introduced. Li et al. [14] reviewed in detail the achievements and limitations of GCN for SSL tasks. Subsequently, studies extending GCN for SSL began to develop. Giang et al. [15] explored GCN-based graph construction. Yang et al. [16] combined classical graph regularization methods with GCN. Abu et al. [17] presented an N-GCN proposal that combined random walks with GCN, followed by GIL [18] with similar ideas. Some other studys on GCN extensions for semi-supervised learning can be found in [19, 20, 21, 22, 23, 24, 25, 26, 27].

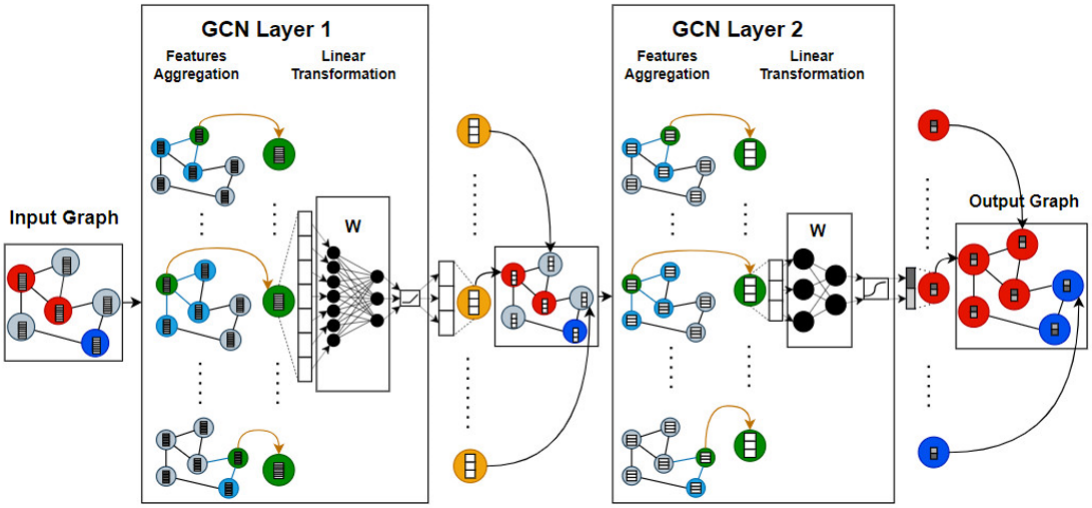

FIGURE 1. Graph convolutional Networks diagram

2.3. **Semi-supervised learning diagram on graph.** Fundamentally, GCN is basically a type of deep neural network specially designed to process graph-structured data, where the information of a vertex depends not only on itself but also on its neighboring vertices in the graph structure.

GCN is designed to combine the features of a vertex with its neighboring vertices to learn a better feature representation. The input to a GCN is a graph constructed with each vertex being a feature vector, two vertices are connected by an edge if there is some relationship. Graph convolutional layers perform feature updates for each vertex by taking the average (or weighted sum) of the features of itself and its neighboring vertices. The output of the GCN is a graph whose features at each vertex have been updated by the convolution process. At each layer, for each node, the information synthesis process is performed in three steps: feature synthesis, linear transformation, and linear activation.

We redraw the detailed general diagram of the semi-supervised learning GCN process on the graph in Figure 1.

3. Proposed method.

3.1. General diagram of proposed method to improve accuracy of image retrieval with relevance feedback.

As shown in Figure 2, we find that the accuracy of image retrieval with relevant feedback using the SVM model (left green border frame) improves as the SVM training process yields a good output model, which depends on the number and balance of labeled samples in each feedback loop.

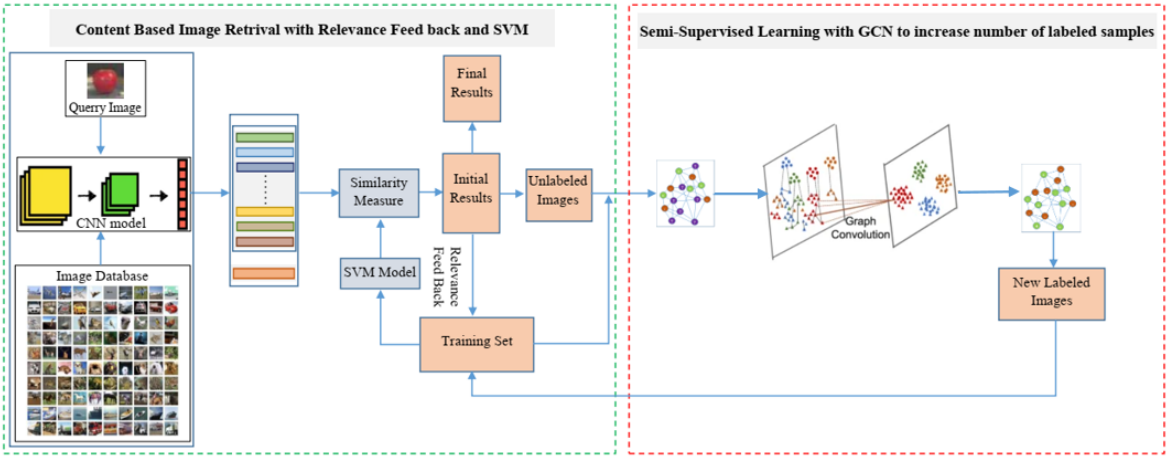

FIGURE 2. Overall architecture of the proposed method

The result of the feedback process is that the input to the SVM training process often has a small number of labeled samples. Therefore, to automatically increase the labeled samples for the input to the SVM training process, we add the right red border, using a graph convolutional neural network to do semi-supervised learning on a dataset consisting of samples labeled by the cumulative feedback process and unlabeled samples taken from the image database. The input graph of the GCN is a graph with a small number of labeled nodes and a large number of unlabeled nodes. The semi-supervised learning process on the graph will predict new labels for the unlabeled nodes to be nodes with positive labels '+' or negative labels '–'. Thus, the GCN will automatically supplement a large number of labeled samples, this set of newly labeled samples can be used to train the SVM. We will present the details of the two frames of the diagram in the following sections 3.2 and section 3.3.

3.2. Updating the retrieval results with relevant feedback and active learning.

The left blue frame in Figure 2, we use the feature vectors extracted by a convolutional neural network model (e.g. LeNet-5, AlexNet, VGGNet, GoogLeNe, ResNet, DenseNet...) then we calculate the distance of the query feauture and the feautures in the database to get the initial result, based on the initial result, the user feedback to indicate the results related (same topic) to the query image is labeled positive (+).

The result images that are not related (not same topic) to the query image are labeled negative (-). The set of features corresponding to the images that have been responded with labels +, - will be used to train the SVM model. After training the SVM model, the labeled samples based on the feedback are divided into two classes and a hyperplane: $f(x) = wx+b$ is determined, this hyperplane is used to update the retrieval results for the next feedback loop. During the feedback process, we improve the active learning method

from [16] to update the retrieval results from unlabeled images in the database, and this updated result will be used in the next feedback loop. First, the active learning stage is presented as follows. Let $\mathcal{L} = (x_1, y_1), \ldots, (x_p, y_p)$ be the labeled data set (relevant or not relevant) during the relevant feedback process; $\mathcal{U} = x_{(p+1)}, \ldots, x_n$ be the data set of the remaining unlabeled data samples, where $x_j \in \mathcal{R}^k$ represents the k-dimensional image feature vector. Let $\mathcal{S}$ be the set of unlabeled data samples and $rsk(f, \mathcal{S}, \mathcal{L}, \mathcal{U})$ is the risk function or evaluation function, which depends on the classification model f. The goal is to find the most useful set of unlabeled data samples $\mathcal{S}^*$ such that the risk function has the smallest value:

$$\mathcal{S}^* = \overset{argmin}{\mathcal{S} \subseteq \mathcal{U} \wedge |\mathcal{S}| = k} rsk(f, \mathcal{S}, \mathcal{L}, \mathcal{U}) \tag{1}$$

The SVM-based active learning method selects unlabeled data samples that are closest to the decision boundary of the SVM model:

$$x^* = \overset{argmin}{x \in \mathcal{U}} |f(x)| \tag{2}$$

In order to increase the retrieval accuracy after the SVM decision boundary is trained based on the initial user feedback, in addition to considering the location of the data point in the SVM classification space, we consider also considering the similarity between the feature vectors of the query image and the images in the database because it is clear that comparing the similarity directly in this way brings greater accuracy than the prediction model. Therefore, in this paper, we combine the confidence level in the SVM and the similarity between the vectors to create an evaluation function and thereby serve as a criterion for selecting the data set $\mathcal{S}$. Let $\mathcal{DS}_j$ be the distance from data point x_j to the decision boundary of the SVM model:

$$\mathcal{DS}(x_j) = |f(x_j)| = |w.x_j + b| \tag{3}$$

where w and b are the weights of the hyperplane in SVM and x_j is the feature vector of image j. Let $\mathcal{DE}_j$ be the Euclidean distance between the query image q and image j in the unlabeled image set. In [16], $\mathcal{DE}_j$ is calculated as the distance between image j and the target image t, where $x_t = \overset{argmax}{i \in U} \mathcal{DS}(x_i)$; however, this calculation will ignore the similarity to the query image. Therefore, we will calculate $\mathcal{DE}_j$ in the following way:

$$\mathcal{DE}(x_j) = \begin{cases} \parallel x_j - x_q \parallel & \text{if } f(x_j) \geq 0 \\ \infty & \text{otherwise} \end{cases} \tag{4}$$

Combined, we have the evaluation function for image j:

$$\mathcal{DSE}(x_j) = \alpha_{\mathrm{x}} \mathcal{DS}(x_j) + \beta_{\mathrm{x}} \mathcal{DE}(x_j) \tag{5}$$

where α and β are parameters to limit the importance of the 2 values $\mathcal{DS}$ and $\mathcal{DE}$. α and β can be constructed by the following formula:

$$\begin{cases} \alpha = \frac{N_{rel}}{N_{rel} + N_{nonrel}} \\ 1 - \frac{N_{rel}}{N_{rel} + N_{nonrel}} \end{cases} \tag{6}$$

where N_{rel} and N_{nonrel} are the number of relevant and not relevant data samples in each loop of the relevance feedback. This formula focuses on controlling the influence of the two values $\mathcal{DS}$ and $\mathcal{DE}$. Specifically, initially the number of relevant images is small, the evaluation function will emphasize the importance of $\mathcal{DE}$ more; but when the number of relevant images is large, the SVM model is improved, the $\mathcal{DS}$ value will be more important. So α and β can be tested many times and declared with a fixed value so that $\beta > \alpha$.

From the above $\mathcal{DSE}$ evaluation function, the algorithm will select unlabeled data samples with the smallest $\mathcal{DSE}$ value to update the retrieval result as the output of the

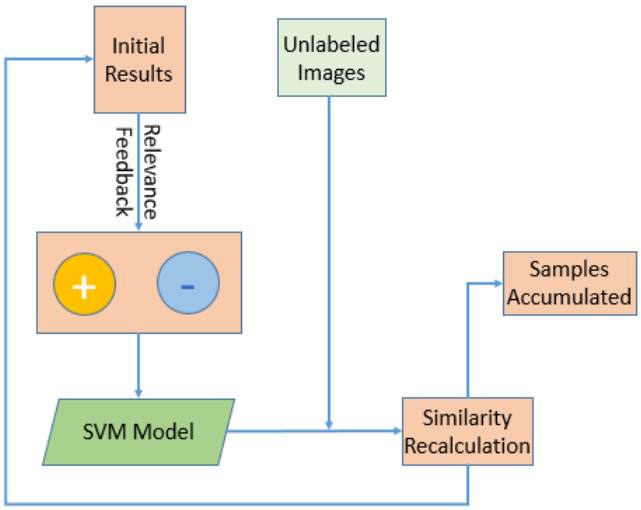

FIGURE 3. Diagram of a feedback loop and sample accumulation

relevance feedback process and cumulatively update the data sample to build the input graph. The process of updating retrieval result is performed in the following steps:

(1) Step 1: Label the data set $\mathcal{S}$ using the SVM model, classifying it into 2 classes: relevant and not elevant. Let $\mathcal{T}$ be the dataset including $\mathcal{N}$ old query result images and m images in data set $\mathcal{S}$.

(2) Step 2: Calculate the $\mathcal{DS}$ and $\mathcal{DE}$ values for the data points in set $\mathcal{T}$.

(3) Step 3: Calculate the $\mathcal{DSE}$ value for the data points in set $\mathcal{T}$.

In this step, if $\mathcal{DSE}$ calculated according to formula (5), the data points that have been evaluated as relevant in the previous retrieval results may be pushed down in rank after the new images. Therefore, the retrieval results may be biased because the relevant data points are not considered. Furthermore, the goal is to display all relevant images first, so it is necessary to multiply the $\mathcal{DSE}$ value by a parameter γ so that the retrieval results are still displayed in the following order: previous relevant images, newly updated relevant images, and other images. To do that, the algorithm will perform:

$$\mathcal{DSE}(x) = \begin{cases} \mathcal{DSE}(x)_{\mathrm{x}}\gamma & \text{if } x \text{ is a relevant in previous results} \\ \mathcal{DSE}(x)_{\mathrm{x}}\frac{1}{\gamma} & \text{otherwise} \end{cases} \tag{7}$$

(4) Step 4: Output a set of $\mathcal{N}$ new retrieval result images based on the $\mathcal{DSE}$ value sorted in ascending order. In other words, the image with the smallest $\mathcal{DSE}$ value is the image with the highest similarity level and will be output first.

(5) Step 5: Update the labeled data set $\mathcal{L}$, the unlabeled data set $\mathcal{U}$ and repeat the relevance feedback process until the retrieval result is satisfied

The general diagram of a feedback loop, recalculating the index and updating the retrieval results, adding samples to the input graph is shown in Figure 3.

3.3. **Using graph convolutional networks to augment training samples.** With the accumulated data set for graph construction in several feedback loops, we use these features to construct the input graph using the mutual $k-NN$ graph approach proposed by Kohei Ozaki et al. [28] as an efficient graph construction for semi-supervised learning.

We construct the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ for semi-supervised learning using graph convolutional networks; where $\mathcal{V} = v_1, v_2, v_3, \ldots, v_{\mathcal{N}}$; $v_i \in \mathcal{R}^F$ x $\mathcal{L}$ is the node set of the graph with F is the dimension of the feature vector, $\mathcal{A} \in \mathcal{R}^{\mathcal{N}\mathrm{x}\mathcal{N}}$ is the adjacency matrix

representing the edges in the mutual $k-NN$ undirected graph, $\mathcal{D} \in \mathcal{R}^{\mathcal{N}x\mathcal{N}}$ is the degree matrix of the graph corresponding to the adjacency matrix $\mathcal{A}$. With the initial data set $\mathcal{S} = (\mathcal{X}, \mathcal{L})$; where $\mathcal{X} \in \mathcal{R}^{\mathcal{N}xF}$ is the feature matrix and $\mathcal{L} \in \{-1; 0; 1\}$ is the label set of the image set with the convention 1 (label +); 0 (label –) and -1 (unlabeled). The graph $\mathcal{G}$ is constructed according to the following steps:

(1) Step 1: Initialize

$\mathcal{V} \leftarrow \varnothing$

$\mathcal{A} \leftarrow \varnothing$

(2) Step 2: Create the graph

for i = 1...$\mathcal{N}$ do:

$v_i = (x_i, l_i)$

$\mathcal{V} \leftarrow v_i$

for j = 1...$\mathcal{N}$ do:

if $(x_j \in k_{NN}(x_i))$ and $(x_i \in k_{NN}(x_j))$:

$\mathcal{A} \leftarrow e(i, j)$

end if

end for

end for

(3) Step 3: Combines the mutual k-NN graph and the maximum spanning tree

The result is the graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$; $\mathcal{V} = \{v_1, v_2, v_3, \ldots, v_{\mathcal{N}}\}$, $v_i \in \mathcal{R}^F x \mathcal{L}$ is the node set of the graph with F is the dimension of the feature vector, $\mathcal{A}$ is the edge set of the graph. Next, we will present a semi-supervised learning method using graph convolutional neural networks to infer the labels of the remaining unlabeled nodes in the graph. After the semi-supervised learning process to infer labels, all vertices of the output graph will have labels 1 (+) or 0 (-), we use this new label set as the training data for the SVM model and conduct lookups on the newly obtained SVM model. In general, each GCN graph convolutional layer performs three operations: feature synthesis, linear transformation, and linear activation. The main difference between the GCN layer and the MLP (multi-layer perceptron) layers lies in the feature synthesis process from neighboring nodes. The representation of the node under consideration can be averaged from the feature representations of its neighbors.

Assume that at the graph convolution layer k^{th}, $H^k \in \mathcal{R}^{\mathcal{N}xF_k}$ is the feature vector representing the node and F_k is the vector length. The initial representation of the node is $H^0 = V$. The feature synthesis process taking place in the GCN layer is performed according to the following formula:

$$H^{(k)} = \hat{D}^{\frac{-1}{2}} \hat{A} \hat{D}^{\frac{-1}{2}} H^{(k-1)} \tag{8}$$

where $\hat{A} = A + I$ denotes the adjacency matrix with the circular edge and $\hat{D}$ is the corresponding degree matrix.

After the feature synthesis step, the remaining two steps of the GCN are linear transformation and nonlinear activation, are performed in the same way as the basic MLP layers. The k^{th} GCN layer consists of a trainable weight matrix $W^k \in \mathcal{R}^{F_{k-1}xF_k}$ and a nonlinear activation function $\sigma(x)$, such as $ReLU(x) = max(0, x)$. The node's feature representation is updated according to the formula:

$$H^{(k)} = \sigma(H^{(k)} W^{(k)}) \tag{9}$$

The weight matrix W is optimized by minimizing the loss function. The process of performing convolution on the graph with labeled +, labeled - and unlabeled vertices to perform the transformation and give the output graph consisting of all labeled vertices. From the output graph we use the learned labels to train the SVM.

4. Experiments.

4.1. **Dataset.** We conducted experiments on two image datasets Cifar100 (Canadian Institute for Advanced Study) and Oxford-IIIT Pet.

With the Cifar100 dataset, we combined the training set of 50,000 images and the test set of 10,000 images to have a common dataset of 60,000 images. In which, we took the query images in the 10,000 images test set to retrieval in the entire 60,000 images experiment dataset. The size of each image in the dataset is 32x32 pixels. With the Oxford-IIIT Pet dataset, there are 7,349 images divided into 37 classes (25 dog breeds, 12 cat breeds), the number of images in each class is not the same and ranges from 180 to 200 images, each image has a different resolution, averaging about 400 × 350 pixels. We preprocess to reduce the images in this dataset to the same size of 224×224 pixels. For each layer, we take each query image and look it up on the entire Oxford-IIIT Pet database.

We used the feature extraction model ViT_b_16 to extract initial features for the images, with the feature vector extracted by ViT_b_16 is 768 dimensions. The nodes of the input graph and the output graph are a set of 2 components $u = \{\text{label, feature vector}\}$, in which the label has a value of 1 corresponding to a positive label, 0 corresponding to a negative label, and -1 corresponding to no defined label. Thus the input graph consists of labeled and unlabeled nodes, the output graph consists of all nodes labeled by semi-supervised learning using GCN.

4.2. **Set parameters.** In the experiment on updating the retrieval results with relevant feedback and active learning in section 3.2 we use fixed values for α, β, γ parameters in formulas (6) and (7). In formula (6) we assign values for parameters as $\alpha = 0.3$ and $\beta = 0.7$ and in formula (7) we set $\gamma = 1/4$

In section 3.3, with GCN semi-supervised learning, we use the Adam optimization method and set the experimental parameters learning_rate to 0.01, weight_decay to 0.0005 and use 3 convolutional layers **Conv1**, **Conv2**, **Conv3** with the following feature vector dimensions:

Table 1. GCN model architecture parameters.

Layer	**Conv1**	**Conv2**	**Conv3**
Dimensions of Input vector	768	126	32
Dimensions of Output vector	126	32	2

The experiment was performed on a Python3 environment with a computer using Linux Operating System, 16GB P100 GPU provided by Kaggle.

4.3. **Results.** We use a simple Euclidean distance function to calculate the similarity between the query image and the images in the database. At each retrieval loop, we display the 100 images closest to the query image, then we use an automatic feedback mechanism to feedback on these 100 result images. After several feedback loops, the number of labeled samples is increased by the relevant feedback and the feedback result accumulation process as presented in section 3.2.

To apply the sample augmentation semi-supervised learning module (red frame on the right), we add randomly 1000 unlabeled images in the database to the accumulated labeled samples to build the input graph for the GCN. Visualize the input graph with Cifar100 dataset for the GCN with 1000 vertices as shown in Figure 4 below.

We divide the data set in the input graph into two train-test sets for semi-supervised learning as follows: the train set consists of images that have been labeled + and – by the

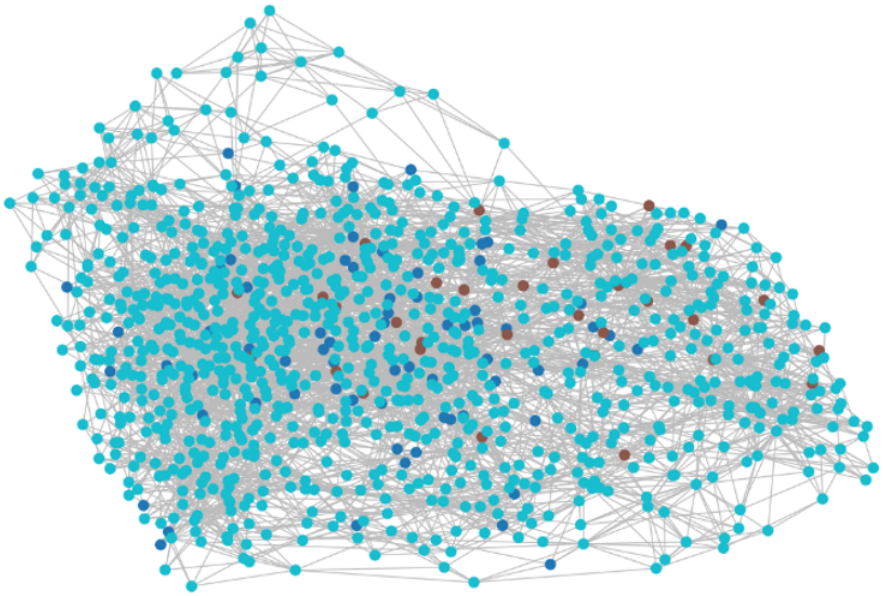

FIGURE 4. Visualize the input graph with Cifar100 dataset for the GCN with 1000 vertices

cumulative feedback process, and the test set consists of unlabeled images. The result of semi-supervised learning on GCN will predict the labels for the nodes of the test set. The visualization of the prediction results of the test sets with the first feedback loop (with Cifar100 dataset) is shown in Figure 5 below:

FIGURE 5. The visualization of the prediction results of the test sets (with Cifar100 dataset)

Visualize comparison of the retrieval results with a randomly selected query image for the cases with and without the semi-supervised learning module applied to augment for training the SVM (with Cifar100 dataset) is shown in Figure 6 in the first feedback loop.

In the first feedback loop (with Cifar100 dataset), figure (a) is the image retrieval result with relevant feedback using SVM active learning without GCN, figure (b) is the image retrieval result with relevant feedback using SVM active learning supplemented with GCN semi-supervised learning module (our proposed GCN_IRRF model). Visually, in figure

FIGURE 6. Visualize comparison of the retrieval results (with Cifar100 dataset)

(a) without applying GCN, the result is 38/100 relevant images, while in figure (b) with GCN module applied, the result is 59/100 relevant images.

TABLE 2. mAP comparison for the two cases applying and not applying semi-supervised GCN learning (with Cifar100 dataset).

Relevance Feedback Loop	Without GCN	With GCN: Our GCN_IRRF
0	0.34	0.52
1	0.43	0.58
2	0.62	0.73
3	0.83	0.87

TABLE 3. mAP comparison for the two cases applying and not applying semi-supervised GCN learning (with Oxford-IIIT Pet dataset).

Relevance Feedback Loop	Without GCN	With GCN: Our GCN_IRRF
0	0.41	0.49
1	0.54	0.63
2	0.72	0.81
3	0.80	0.89

To demonstrate the effectiveness of applying the GCN augmented learning module, we also use the method of evaluating the performance of the retrieval model by experimentally calculating the mAP metric. The mAP measure is an accuracy measure to compare the query performance in the image retrieval problem. The method of determining the mAP value is presented as follows: Average Precision (AP): refers to the coverage area under the precision-recall curve. AP is higher implies a higher precision curve and better retrieval accuracy. AP can be calculated as follows:

$$AP = \frac{\sum_{k=1}^{N} P(k).rel(k)}{R} \tag{10}$$

where R denotes the number of relevant results for the query image from N images. $P(k)$ is the precision of the k images in the returned results and $rel(k)$ is an indicator function that equals 1 if the item in rank k is relevant image and 0 otherwise. The mean average precision (mAP) is applied to evaluate over all query images:

$$mAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q) \tag{11}$$

where Q is the number of query images. The results of mAP comparison for the two cases applying and not applying semi-supervised GCN learning are shown in Table 2 (with Cifar100 dataset) and Table 3 (with Oxford-IIIT Pet dataset) above.

The above results show that when applying the GCN semi-supervised learning module to increase the number of training samples for SVM, the retrieval results are significantly improved. The above analysis and evaluation show that our proposed method has improved the performance of the relevance feedback image retrieval model.

5. **Conclusions.** In this paper, we have presented an efficient proposed model to automatically supplement the SVM training set for image retrieval with relevant feedback and other related tasks. The proposed model has solved the difficult problem of tasks that require user feedback for label information, by automatically supplementing the predicted labels when applying semi-supervised learning based on GCN.

This proposed model is not only applicable to the task of image retrieval with relevant feedback but can be applied to other classes of tasks that require automatic label generation or automatic classification by changing the dimensionality of the output vector in the last convolutional layer, which is changed to be equal to the number of labels or the number of classes of the corresponding task.

Experimental results performed on two image dataset CIFAR-100 and Oxford-IIIT Pet have demonstrated that our proposed model produces highly accurate results. In the next research, the authors plan to continue to study extended models of GCN to solve more complex computer vision tasks with multi-label images. Exploiting GCN and its variants to discover semantic relationships in the label space and local features, object features appearing in multi-label images

REFERENCES

[1] Mahesh Prajapati, Vijay Bhandari, Ritu Shrivastava, "Recent Advances in Content-Based Image Retrieval: Techniques and Applications", *Journal of Computer Technology and Applications*, vol. 16, no. 1, 2025.

[2] Phuc Nguyen, "Enhancing Image Retrieval Efciency through Text Feedback to Improve Search Performance", *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 15, No. 1, pp. 21–35, 2024.

[3] Hamed Qazanfari, Mohammad M. AlyanNezhadi, Zohreh Nozari Khoshdaregi, "Advancements in Content-Based Image Retrieval: A Comprehensive Survey of Relevance Feedback Techniques", *Computer Vision and Pattern Recognition*, 2023, https://arxiv.org/pdf/2312.10089

[4] Dongping Tian, "Support Vector Machine for Content-based Image Retrieval: A Comprehensive Overview", *Journal of Information Hiding and Multimedia Signal Processing*, Vol.9, Number 6, pp. 1464-1478, 2018

[5] R. Pighetti, D. Pallez and F. Precioso, "Hybrid content-based image retrieval combining multiobjective interactive genetic algorithm and SVM", *Proc. of the 21st Int'l Conf. on Pattern Recognition (ICPR'12)*, pp. 2849–2852, 2012.

[6] C. Ma, Q. Dai and S. Liu, "A hybrid PSO and active learning SVM model for relevance feedback in the content-based images retrieval", *Proc. of the Int'l Conf. on Computer Science & Service System (CSSS'12)*, pp. 130–133, 2012

[7] Y. Qi and G. Zhang, "Strategy of active learning support vector machine for image retrieval", *IET Computer Vision*, vol. 10, no. 1, pp. 87–94, 2016.

[8] X. Wang, H. Yang, Y. Li, et al., "A new SVM-based active feedback scheme for image retrieval", *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 43–53, 2015.

[9] X. Wang, L. Liang, W. Li, et al., "A new SVM-based relevance feedback image retrieval using probabilistic feature and weighted kernel function", *Journal of Visual Communication and Image Representation* vol. 38, pp. 256–275, 2016.

[10] Truong-Giang Ngo, Quoc-Tao Ngo, Duc-Dung Nguyen, "Image Retrieval with Relevance Feedback using SVM Active Learning", *International Journal of Electrical and Computer Engineering (IJECE)*, Vol.6, No.6, pp. 3238–3246, 2016.

[11] Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph laplacian for semisupervised learning", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2261–2274, 2015

[12] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting semi-supervised learning with graph embeddings", *International conference on ma22 chine learning. PMLR*, 2016, pp. 40–48

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", *in ICLR (Poster). OpenReview.net*, 2017

[14] Q. Li, Z. Han, and X. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning", *in AAAI. AAAI Press*, 2018, pp. 3538–3545

[15] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 313–11 320.

[16] H. Yang, K. Ma, and J. Cheng, "Rethinking graph regularization for graph neural networks", *arXiv preprint arXiv:2009.02027*, 2020.

[17] S. Abu-El-Haija, A. Kapoor, B. Perozzi, and J. Lee, "N-gcn: Multiscale graph convolution for semi-supervised node classification", *in Uncertainty in Artificial Intelligence. PMLR*, 2020, pp. 841–851.

[18] C. Xu, Z. Cui, X. Hong, T. Zhang, J. Yang, and W. Liu, "Graph inference learning for semi-supervised classification", *in ICLR. OpenReview.net*, 2020.

[19] R. Liao, M. Brockschmidt, D. Tarlow, A. L. Gaunt, R. Urtasun, and R. S. Zemel, "Graph partition neural networks for semi-supervised classification", *in ICLR (Workshop). OpenReview.net*, 2018.

[20] Y. Zhang, S. Pal, M. Coates, and D. Ustebay, "Bayesian graph convolutional neural networks for semi-supervised classification", *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5829–5836.

[21] S. Vashishth, P. Yadav, M. Bhandari, and P. P. Talukdar, "Confidencebased graph convolutional networks for semi-supervised learning", *in AISTATS, ser. Proceedings of Machine Learning Research*, vol. 89. PMLR, 2019, pp. 1792–1801

[22] C. Zhuang and Q. Ma, "Dual graph convolutional networks for graphbased semi-supervised classification", *in WWW. ACM*, 2018, pp. 499–508.

[23] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning", *arXiv preprint arXiv:2009.07111*, 2020.

[24] M. T. Kejani, F. Dornaika, and H. Talebi, "Graph convolution networks with manifold regularization for semi-supervised learning", *Neural Networks*, 2020.

[25] B. Xu, H. Shen, Q. Cao, K. Cen, and X. Cheng, "Graph convolutional networks using heat kernel for semi-supervised learning", *arXiv preprint arXiv:2007.16002*, 2020.

[26] F. Hu, Y. Zhu, S. Wu, L. Wang, and T. Tan, "Hierarchical graph convolutional networks for semi-supervised node classification", *in IJCAI. ijcai.org*, 2019, pp. 4532–4539.

[27] K. Sun, Z. Lin, and Z. Zhu, "Adagcn: Adaboosting graph convolutional networks into deep models," CoRR, vol. abs/1908.05081, 2019

[28] Kohei Ozaki and Masashi Shimbo and Mamoru Komachi and Yuji Matsumoto, "Using the Mutual k-Nearest Neighbor Graphs for Semi-supervised Classification of Natural Language Data", *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 154–162, 23–24 June 2011