IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 20, 2022, accepted March 4, 2022, date of publication March 16, 2022, date of current version March 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3160214

Content-Based Video Retrieval With Prototypes of Deep Features

HYEOK YOON AND JI-HYEONG HAN
Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
Corresponding author: Ji-Hyeong Han (jhhan@seoultech.ac.kr)

This work was supported by the Grant from the Institute of Information and Communications Technology Planning and Evaluation (IITP) by the Korean Government through MSIT (Development of autonomous VR and AR content generation technology reflecting the usage environment) under Grant 2020-0-00994.

ABSTRACT The rapid development in the area of information and communication technologies has enabled the transfer of high-resolution, large-sized videos, and video applications have also evolved according to data quality levels. Content-based video retrieval (CBVR) is an essential video application because it can be applied to various domains, such as surveillance, education, sports, and medicine. In this paper, we propose a CBVR method based on prototypical category approximation (PCA-CBVR), which calculates prototypes of deep features for each category to predict the user's query video category without a classifier. We also undertake fine searching to retrieve the video most similar to the user's query video from the predicted category database of videos. The proposed PCA-CBVR approach is efficient in terms of its computational cost and maintains meaningful information of the videos. It does not need to train a classifier even when the database is updated and uses all deep features without any dimension reduction step, such as those in CBVR studies. Moreover, we conduct fine-tuning of the 3D CNN feature extractor based on a few-shot learning approach for better domain adaptation ability and apply salient frame sampling instead of uniform frame sampling to improve the performance of the PCA-CBVR method. We demonstrate the performance capability of the proposed PCA-CBVR approach through experiments on various benchmark video datasets, in this case the UCF101, HMDB51, and ActivityNet datasets.

INDEX TERMS Video retrieval, deep learning, video analytics, prototypes, cross-domain evaluation, few-shot learning.

I. INTRODUCTION

In recent years, the rapid developments of information and communication technologies have enabled faster and easier access to large volumes of data. In March of 2020, due to COVID-19, daily uploads and views of videos at home increased on YouTube by nearly 700% and 210%, respectively, compared to the corresponding levels before the coming of the pandemic [1]. By 2023, the numbers of internet users and 4K TV connections are predicted to be around 5.3 billion and 891 million, respectively [2], meaning that the transmitted video traffic and quality levels will also grow. Therefore, research focusing on with video applications has been active [3]–[6].

Video contains more complicated information compared to a single image, combining motion, audio and text. Accordingly, video application research requires an integrated approach to consider the various types of information. Content-based video retrieval (CBVR) is one area of video application research. The aim of CBVR is to search for videos that are most similar to a query video from a database based on only the video contents without any additional metadata. CBVR has been applied in several domains, such as crime prevention, by identifying suspects through CCTV videos [7], the indexing and retrieving of specific lecture videos for effective education [8], the retrieval of the sports videos such as badminton [9], and the retrieval of surgical procedures similar to an ongoing procedure to ensure an efficient operation [10], among others.

In general, the CBVR process consists of three steps. The first step is frame sampling. To process the video efficiently and enhance the retrieval performance, it is necessary to sample meaningful frames, as many video frames do not assist with an understanding of the video. For example, there

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Mei Zhang.

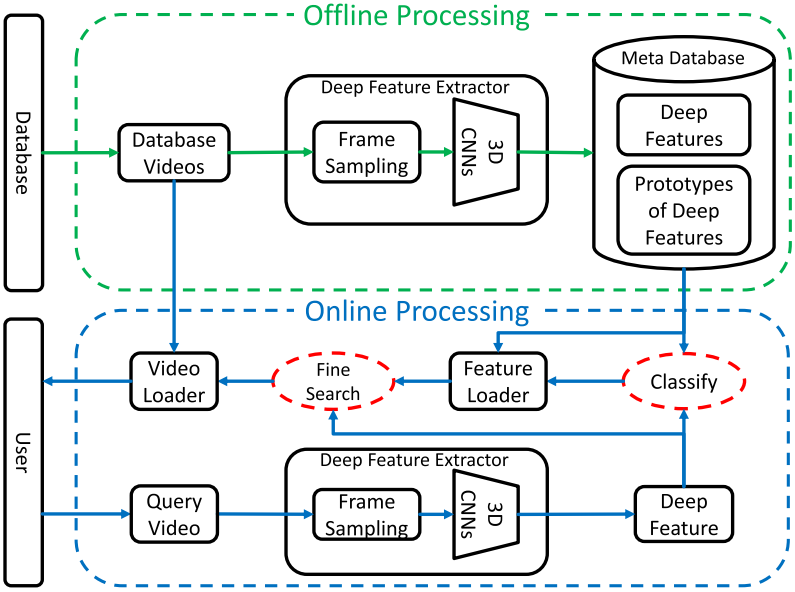


FIGURE 1. The proposed PCA-CBVR is composed of offline (green dashed line) and online (blue dashed line) steps. During offline processing, PCA-CBVR extracts the deep features of database videos using 3D CNNs which exclude the last fully connected layer, and calculates the category prototypes from the extracted deep features. During online processing, PCA-CBVR measures the degree of similarity between the category prototypes and the extracted deep features of the user's query video. Then, the PCA-CBVR method predicts the user's query video category based on the measured similarity scores and fine searches of the predicted category database videos to find the video most similar to the user's query video. Finally, as a result of the fine search, it returns the top-k videos most similar to the user's query video.

TABLE 1. Table of abbreviations.

Abbreviation	Definition
CBVR	Content-Based Video Retrieval
PCA-CBVR	Prototypical Category Approximation Content-Based Video Retrieval
PCA	Principal Component Analysis
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ResNet	Residual Neural Network
R3D	3D Residual Neural Network
R(2+1)D	(2+1)D Residual Neural Network
SGD	Stochastic Gradient Descent
AP	Average Precision
mAP	Mean Average Precision

may be very similar frames in the same context of a certain duration. In this step, we obtain the meaningful key-frames [11], [12] or uniformly sampled frames [8], [13]. The second step is feature extraction. In this step, we extract a color features histogram [14] of the sampled frames using traditional computer vision techniques or deep features [15]–[17] of the sampled frames using a convolutional neural network (CNN). The last step is the distance calculation step which compares extracted features from both the database videos and the query video in terms of the Euclidean distance or cosine similarity metric. Finally, the database video with the shortest distance relative to the query video is retrieved as the video most similar to the query video.

Recently, with the development of deep learning techniques, several deep learning approaches, especially 2D and 3D CNNs, have been widely applied as aspects of CBVR methods. 3D CNNs represent the spatial and temporal features of videos well compared to 2D CNNs [18]. However, 3D CNNs are inefficient compared to 2D CNNs because 3D CNNs require too much video data and long processing times for 3D kernel optimization [19]. To overcome these problems, the recent research has fine-tuned pre-trained 3D CNNs on large-scale video datasets, resulting in a better 3D CNN model that requires less effort than a model trained from scratch [20].

Although the deep learning approach-based CBVR methods have been studied actively, there are still research problems to be addressed. In this paper, we tackle the following four major challenges. The first challenge is the information loss problem because of dimension reduction. Many of CBVR methods applied the dimension reduction to video data to compress the deep features for low computation resources. However, the deep feature information would be lost when trying to reduce the dimensions of these features. The second challenge is re-training the classifier when the database is updated. Many of CBVR methods used the trained classifiers based on the database videos to predict the user's query video category. However, in the real world, tons of new videos are constantly generated, thus the database must be updated. Consequently, the classifiers also must be

re-trained to fit the updated database and this re-training process requires a lot of time and computing resources. The third challenge is the novel domain adaption ability to reduce training cycle, which is a closely related issue with the second challenge. When the database is updated, there is a possibility of adding novel categories not just adding videos of existing categories. As aforementioned, since re-training the classifiers requires a lot of time and resources, if the CBVR method adapts the novel categories and domains without additional re-training, then we could save a lot of time and resources. The last challenge is frame sampling problem for effective video retrieval. Since the lengths of videos are variable and sometimes very long, the CBVR methods need to sample the frames of videos. Thus, the performance of CBVR methods depend on the performance of frame sampling because if we retrieve relevant frames more than meaningless frames then they would contain better information of the videos.

To solve the aforementioned four problems, we propose a CBVR method based on prototypical category approximation (PCA-CBVR) which calculates category prototypes of deep features to predict the user's query video category efficiently without using any dimension reduction algorithms and classifiers. After predicting the user's query video category, the proposed PCA-CBVR utilizes a fine searching step to find videos from the predicted category videos in database, of which are the most similar to the user's query video. Since the PCA-CBVR does not reduce the dimension of deep features and predicts the user's query video category based on similarity measurement without classifiers, it does not have an information loss problem and need to re-train the classifiers when the database is updated. Moreover, we apply fine-tuning with a few-shot learning approach to the PCA-CBVR and verify that it increases novel domain adaptation ability through cross-domain evaluation. Finally, we find that the PCA-CBVR performance depends on the frame sampling methods. When salient frame sampling method is applied shows better performance than just simple uniform frame sampling. Figure 1 and Table 1 present an overview of the proposed PCA-CBVR method and abbreviations used in this paper.

This paper is organized as follows. Section II provides a brief introduction to research related to CBVR methods that use CNNs and the few-shot learning method. In Section III, PCA-CBVR is proposed and the details of PCA-CBVR are explained. Section IV analyzes the proposed PCA-CBVR performance with the uniform and salient frame sampling methods and assess the domain adaptation ability based on several benchmark datasets. Finally, concluding remarks follow in Section V.

II. RELATED WORKS

In this section, we discuss the research related to the proposed PCA-CBVR. In Section II-A, CBVR research based on CNNs for retrieval performance improvements and the unique frame extraction strategy are explained. In Section II-B, few-shot learning methods are described, as we fine-tune the CNNs using the few-shot learning approach for better video retrieval performance on cross-domain video contents. Table 2 summarizes the characteristics and differences between related research and the proposed PCA-CBVR.

A. USE OF CNNs FOR CBVR

CNNs are widely used in computer vision tasks, especially with large-scale image datasets [23] and video datasets [24], [25]. Given the performance improvements of CNNs on computer vision tasks, they have been also applied in CBVR research.

The first approach is to use 2D CNNs to extract the deep features of videos for CBVR. Yu *et al.* [15] extracted deep features using pre-trained 2D CNNs on ImageNet [23] and quantized the extracted deep features for computational efficiency. They showed that deep features were more efficient for CBVR compared to other low-level features. Yu *et al.* [26] also modified the architecture of 2D CNNs and utilized more data to improve the performance capabilities of existing 2D CNNs. Furthermore, they proposed a new feature fusion method to improve both the performance and robustness. Suo *et al.* [27] modified the proposed SimHash [28] algorithm to reduce the dimensions of deep features and increase the retrieval performance efficiency. Moreover, they calculated the distance between two deep features of the frames for precise retrieval. Anuranji and Srimathi [16] proposed stacked heterogeneous multi-kernel 2D CNNs to capture complex deep features, also using, bidirectional LSTM to train the temporal information, with the results of LSTM then passed to a fully connected layer to obtain the binary hash code. Abed *et al.* [12] proposed a new key-frame extraction method based on 2D CNNs and proved that it was efficient on CBVR by integrating it into the CBVR system.

The second approach is to use 3D CNNs for CBVR. The aforementioned studies applied 2D CNNs to video data; however, videos are a sequence of frames that contain temporal information. Thus, recent research applied 3D CNNs that represent video data including temporal information more efficiently. Ullah *et al.* [17] used a pre-trained C3D model [18] on the Sports-1M dataset [24] and reduced the dimensionality by means of PCA to generate the hash code. However, according to work by Hara *et al.* [19], Kataoka *et al.* [20], and Tran *et al.* [29], working with the Sports-1M dataset is not easy because it contains more videos than the Kinetics 400 and Kinetics 700 datasets [30], [31], and their annotations are noisier than those of these Kinetics datasets. They also demonstrated that 3D ResNets (R3D) [32], [33] and R(2+1)D [29] models could outperform C3D and that a model pre-trained on the Kinetics 700 dataset was better than one pre-trained on the Kinetics 400 dataset. Therefore, we utilize the R3D and R(2+1)D models pre-trained on the Kinetics 700 dataset to extract the deep features of videos efficiently.

The aforementioned previous CBVR research attempted to compress the deep features for low computation resources.

TABLE 2. Comparison of deep learning-based CBVR methods based on the four research challenges, which are dimension reduction, retraining classifiers, frame sampling strategy, and cross-domain testing.

Paper	Dimension Reduction	Deep Feature Extractor		Frame Sampling Strategy	Cross-Domain Testing
		Type	Re-training		
Yu et al. [15]	Yes	2D CNNs Pretrained on ImageNet	Unnecessary	-	Not Provided
Yu et al. [26]	Yes	2D CNNs Pretrained on ImageNet	Unnecessary	Key Frames	Not Provided
Suo et al. [27]	Yes	2D CNNs Pretrained on ImageNet	Unnecessary	Key Frames	Not Provided
Anuranji et al. [16]	Yes	2D CNNs Pretrained on ImageNet + BiLSTM	Necessary	Uniform	Not Provided
Abed et al. [12]	No	2D CNNs	Necessary	Key Frames	Not Provided
Ullah et al. [17]	Yes	3D CNNs pretrained on Sports-1M	Unnecessary	Key Frames	Not Provided
The Proposed PCA-CBVR	***No***	***3D CNNs pretrained on Kinetics-700***	***Unnecessary***	***Uniform, Salient***	***Provided***

However, the deep feature information would be lost when trying to reduce the dimensions of these features. Therefore, in this paper we use the proposed category prototypes from Snell *et al.* [34], which are the mean values of the deep features of videos in the same category without the use of any dimension reduction algorithm. These category prototypes help not only to reduce the computation cost but also classify the query video category without classifiers. More details of PCA-CBVR are explained in Section III.

B. CROSS-DOMAIN GENERALIZATION WITH FEW-SHOT LEARNING

To train and optimize the CNNs successfully, a large and well-labeled training dataset is essential. However, a large dataset labeled by humans is difficult to obtain, and when the training dataset is not well-labeled or is insufficiently labeled, the CNNs are over-fitted relative to the dataset. To overcome this problem, few-shot learning methods, which learn from only a few datasets and generalize to different novel classes, have been proposed. Vinyals *et al.* [35] proposed a matching network that utilizes a memory and attention mechanism for rapid learning. They also provided the mini-ImageNet dataset as a few-shot learning method benchmark. Snell *et al.* [34] proposed a prototypical network that generates prototypes by calculating the mean of deep features for each class in support sets and calculating the Euclidean distance between prototypes and the query data to classify the query data. Sung *et al.* [36] proposed a relation network that applied trainable distance calculation model for further generalization instead of using a fixed distance calculation such as the Euclidean distance. Recently, Chen *et al.* [21] and Tseng *et al.* [22] also showed that few-shot learning methods worked well for cross-domain generalization.

The proposed PCA-CBVR method is motivated by prototypes from Snell *et al.* [34]. We consider the database videos as a support set in the few-shot learning approach to classify the query video without additional classifiers. Cross-domain generalization is also important in the CBVR approach in actual applications because users do not send query videos which have the identical domain to the database videos. Therefore, we conduct fine-tuning of the 3D CNN models on the UCF101 [37] dataset with the few-shot learning approach and evaluate these outcomes on the ActivityNet [39] dataset (UCF101 $\rightarrow$ ActivityNet) in an effort to improve the cross-domain generalization ability of PCA-CBVR. More details pertaining to the cross-domain generalization ability of PCA-CBVR are given in Section IV-D.

TABLE 3. Mathematical notations of PCA-CBVR.

Symbol	Description
$V_c = \{v_1, ..., v_{\lvert V_c \rvert}\}$	The set of videos in category c
$\lvert V_c \rvert$	The number of videos in category c
K_c	The number of clusters in category c
c_q	Predicted query video category
v_r	Retrieved videos
$\boldsymbol{df}_c$	The cluster centroids of the deep features in database videos
$\boldsymbol{df}_q$	The deep feature of the query video
$\boldsymbol{df}_v$	The deep feature of the specific video v
$\boldsymbol{df}_{c_q}$	The deep features in predicted category

III. PCA-CBVR

In this section, we explain the proposed PCA-CBVR method in more details. The proposed PCA-CBVR method consists following steps as shown in Figure 1.

- An offline process that calculates and saves the category prototypes of database videos
  1) Sample 16 frames as resized to 112 $\times$ 112 from each database video uniformly.
  2) Extract the deep features from the sampled frames using pre-trained 3D CNNs which exclude the last fully connected layer.
  3) Calculate the category prototypes from the extracted deep features of the database videos in each category.
  4) Save the category prototypes into a meta-database.
- An online process that retrieves the database video most similar to the user's query video
  1) Sample 16 frames as resized to 112 $\times$ 112 from the query video uniformly.
  2) Extract the deep features from the sampled frames using pre-trained 3D CNNs which exclude the last fully connected layer.
  3) Predict the query video category based on category prototypes in the meta-database without classifiers.
  4) Finely search for database videos in the predicted category most similar to the user's query video.

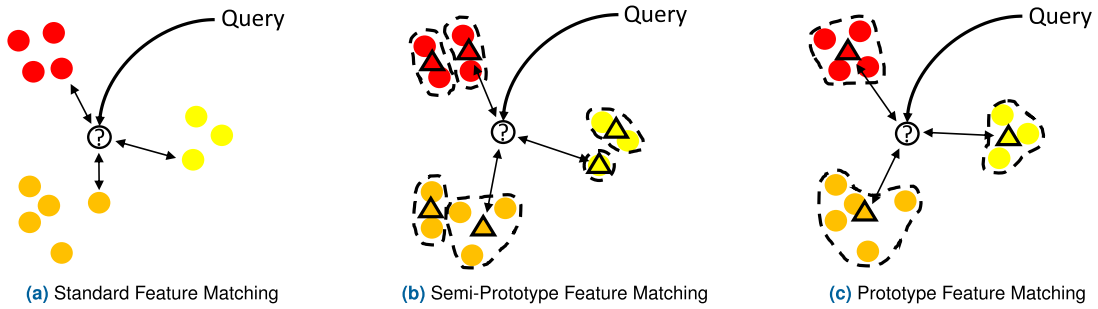


FIGURE 2. Examples of query video category prediction without a classifier. The red, orange, and yellow circles are deep features of videos in the database and the colors represent the categories in the database; i.e., in this example, there are three different categories in the database, and the red, orange, and yellow categories contain four, five, and three videos, respectively. The circle with the question mark is a deep feature of the query video. The red, orange, and yellow triangles in (b) and (c) are the centroids of the clusters and the dashed lines represent each cluster. (a) Standard feature matching to predict the query video category by comparing the deep feature of the query video with the deep features of every video in the database. (b) Semi-prototype feature matching to predict the query video category by comparing the deep features of the query video with the deep features of semi-prototypes in the database. The deep features of the semi-prototypes are calculated according to the centroids of the clusters. In this example, the numbers of clusters for each category, K_{red}, K_{orange}, K_{yellow}, are set to 2, which is in the range of $(1, |V_c|)$. We refer to this as a semi-prototype because there are several deep features for the one category. (c) Prototype feature matching to predict the query video category by comparing the deep feature of the query video with the deep features of the prototypes in the database. The deep features of the prototypes are calculated according to the centroids of clusters and the numbers of clusters for each category, K_{red}, K_{orange}, K_{yellow}, are set to 1. We refer to these as prototypes because there is one deep feature for one category.

There are two main parts of the proposed PCA-CBVR. The first part is category classification of the user's query video, which is done by measuring the similarity between the category prototypes and the deep features of the query video. The other part is a fine search based on the query video category predicted in the first step to obtain the videos most similar to the query video. In the following subsections, the details of each step and organized mathematical notations in Table 3 are explained.

A. USERS' QUERY VIDEO CATEGORY PREDICTIONS WITH PROTOTYPES

We still need to re-train the classifier when the database is updated. For example, if some novel categories of videos are added to the database, the former trained classifier cannot then recognize the added categories. Therefore, in this paper, we use the proposed prototypical category approximation technique from Snell *et al.* [34] to classify the query video without a classifier.

We usually predict the query video category by comparing the query video with every video in the database, as shown in Figure 2(a). However, this approach requires a considerable computation cost, long times, and much memory. Instead, we utilize the K-means clustering algorithm to reduce deep feature matching points, as shown in Figures 2 (b) and (c). For clarification, we redefine the K-means clustering terms as shown in Table 4.

Equation (1) is a generalized form to predict the query video category without a classifier, as follows:

$$c_q = category\left(\underset{\boldsymbol{df}_c}{\text{argmax}}\left(\frac{\boldsymbol{df}_q \cdot \boldsymbol{df}_c}{\|\boldsymbol{df}_q\| \times \|\boldsymbol{df}_c\|}\right)\right) \tag{1}$$

TABLE 4. Redefined terms depending on the range of the hyperparameter K of the K-means clustering algorithm, where K_c is the number of clusters in category c and $|V_c|$ is the number of videos of set $V_c = \{v_1, \ldots, v_{|V_c|}\}$ in specific category c.

Standard Feature Matching	Semi-Prototype Feature Matching	Prototype Feature Matching
$K_c = \|V_c\|$	$1 < K_c < \|V_c\|$	$K_c = 1$

where c_q is the query video category that we want to predict, and $\boldsymbol{df}_q$ and $\boldsymbol{df}_c$ denote the deep features of the query video and the cluster centroids of the database videos, respectively. Equation (1) is intended to compare the deep features of the query video and the cluster centroids in the database based on the cosine similarity measurement and then predict the query video category as the category of the most similar cluster centroid with the greatest degree of cosine similarity. In contrast to Snell *et al.* [34], we employ cosine similarity in Equation (1) to increase the performance. The performance comparison between using Euclidean distance and cosine similarity is presented in Section IV-G and Table 8. If the number of clusters in each category is set to 1, then it is the proposed PCA-CBVR and there is no need to apply the K-means clustering algorithm because the deep features of each category, $\boldsymbol{df}_c$, are calculated as follows:

$$\boldsymbol{df}_c = \frac{\sum_{v \in V_c} \boldsymbol{df}_v}{|V_c|} \tag{2}$$

In this equation, $\boldsymbol{df}_v$ represents the deep feature of v, which is a video from set $V_c = \{v_1, \ldots, v_{|V_c|}\}$ in category c, and $|V_c|$ is the number of videos that correspond to category c.

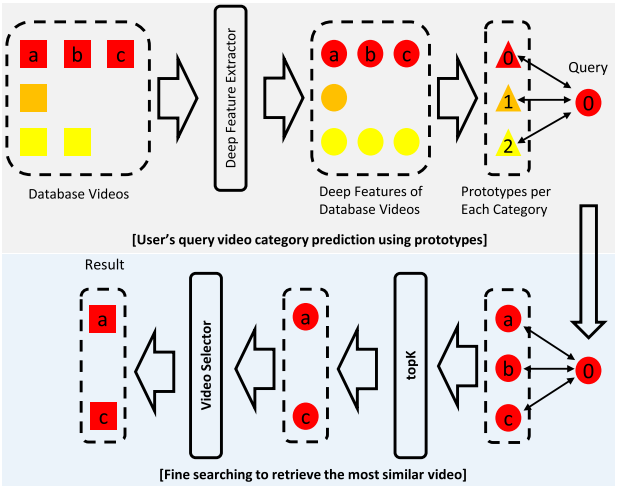


FIGURE 3. The process of the PCA-CBVR method. The squares, circles, and triangles indicate the raw videos, deep features of the videos, and the category prototypes, respectively. In this example, there are three categories, shown in red, orange, and yellow, and the user's query video category is red. First, PCA-CBVR extracts the deep features of database videos and calculates the prototypes of each category. Then, it calculates the cosine similarities between the deep features of the user's query video and those of each prototype and predicts the user's query video category as the prototype category which has the highest degree of cosine similarity. Finally, PCA-CBVR undertakes fine searching to retrieve the most similar video from the database of videos in the predicted category. During the fine search process, it calculates the cosine similarities between the deep features of the database videos in the predicted category and the user's query video and retrieves the top K most similar video with the top K highest degree of cosine similarity. In this example, K is set to 2.

B. FINE SEARCHING ON THE SELECTED EMBEDDING SPACE

In section III-A, we predict a query video's category efficiently using the aforementioned prototypes. After predicting the query video's category, it then becomes necessary to retrieve the video most similar to the user's query video among the database videos in the predicted category. Therefore, the second step of PCA-CBVR is a fine search, which proceeds as follows:

$$v_r = video\left(\underset{\boldsymbol{df}_{c_q}}{\text{argtopK}}\left(\frac{\boldsymbol{df}_q \cdot \boldsymbol{df}_{c_q}}{\|\boldsymbol{df}_q\| \times \|\boldsymbol{df}_{c_q}\|}\right)\right) \tag{3}$$

where v_r is the retrieved video, argtopK returns the arguments which have top K rank, and $\boldsymbol{df}_q$ and $\boldsymbol{df}_{c_q}$ are the deep features of the query video and database videos in the predicted query video's category, respectively. Equation (3) is used to compare the deep features of the query video and database videos in the predicted query video's category based on the cosine similarity measurement and to retrieve the video most similar to the query video. Figure 3 explains the overall process of PCA-CBVR, including the category prediction of the user's query video and the fine search process to retrieve the most similar video.

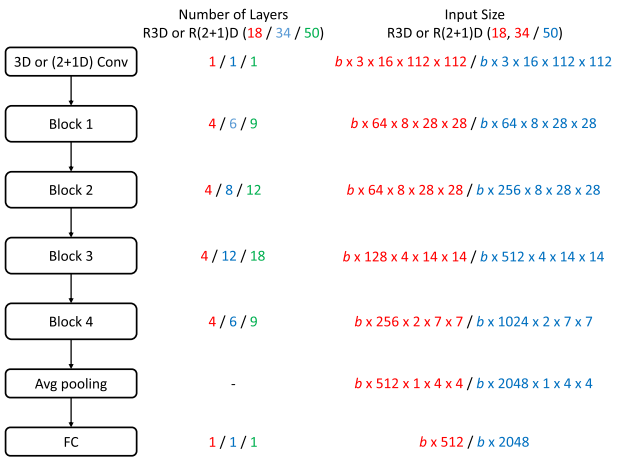


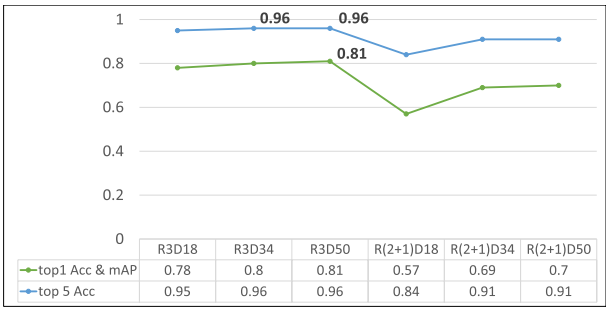
FIGURE 4. The summarized 3D CNN architecture information which is used in this paper. Input size has five dimensions which are (batch size × channel size × sequence size × height × width). The R3D and R(2+1)D with 18 and 34 layers are composed of basic blocks, and ones with 50 layers are composed of bottleneck blocks. To extract deep features, we exclude the last fully connected layer, thus the output shape of deep features is (batch size × 512) for 3D CNNs with 18 and 34 layers and (batch size × 2048) for 3D CNNs with 50 layers, respectively.

IV. EXPERIMENTS AND RESULTS

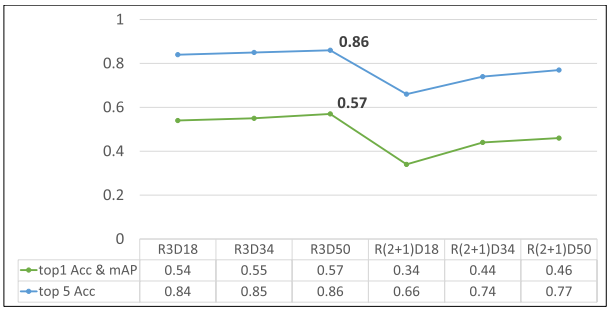
In this section, we verify the performance outcomes of the proposed PCA-CBVR in different situations. First, we evaluated the proposed PCA-CBVR performance based on different combinations of 3D CNNs and depths to determine the best feature extractor. Then, we verified that the proposed PCA-CBVR with the selected feature extractor 3D CNN outperforms other video retrieval methods on certain datasets. Second, we conducted experiments to demonstrate the domain adaptation ability of the proposed PCA-CBVR by fine-tuning based on the few-shot learning approach. Third, we showed the benefit of fine searching compared to random selection to return the best recommending results. Fourth, we applied different frame sampling approaches to the proposed PCA-CBVR, specifically uniform frame sampling and salient frame sampling based on the prototypes, and showed that the salient frame sampling approach based on the prototypes outperforms the uniform sampling approach. Fifth, we showed that cosine similarity boosts PCA-CBVR performance compared to Euclidean distance. Finally, we analyzed video retrieval time to discuss the computational complexity of PCA-CBVR.

Figure 4 shows summarized 3D CNNs architecture which was applied in the experiments along with the number of layers and the input size used in this paper. In the experiments, each category was clustered as one cluster. It means that videos in the same category have the same category prototype which is a mean vector of video features of that category. Moreover, we provide the codes used in this paper, which are available on GitHub.[1] Details of the datasets, the performance

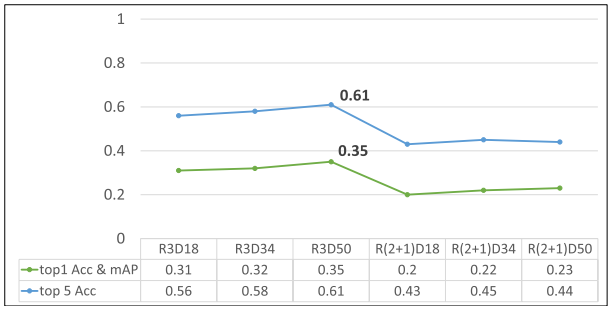
[1] https://github.com/titania7777/PCA-CBVR



(a) PCA-CBVR Results on UCF101



(b) PCA-CBVR Results on HMDB51



(c) PCA-CBVR Results on ActivityNet

FIGURE 5. Accuracy and mAP results of PCA-CBVR on the (a) UCF101, (b) HMDB51, and (c) ActivityNet datasets depending on the 3D CNN feature extractors, which are R3D and R(2+1)D models with different depths. We applied 3D CNNs that were pre-trained on kinetics 700 without any fine-tuning.

evaluation metrics, and the experimental results are explained in the following subsections.

A. DATASETS

We used the UCF101 [37], HMDB51 [38], and ActivityNet [39] datasets, which are representative video datasets widely used in human activity analyses. The UCF101 dataset is a trimmed dataset with 13,320 YouTube videos in 101 action categories, and the numbers of videos for training and testing are 9,537 and 3,738, respectively. The HMDB51 dataset is also a trimmed dataset with 6,766 videos from YouTube, movies, and web in 51 action categories, and the numbers of videos for training, validation and testing are 3,570, 1,666 and 1,530, respectively. On the other hand, ActivityNet is an untrimmed dataset with 19,994 web videos in 200 action categories, and the numbers of videos for training, validation and testing are 10,024, 4,926 and 5,044, respectively. The above three datasets help to confirm retrieval performances on each trimmed and untrimmed video, and (trimmed $\rightarrow$ untrimmed) videos cross-domain adaptation ability. We consider the training data and test data in the UCF101 and HMDB51 datasets as the database videos and the query videos in the video retrieval task, respectively. The ActivityNet dataset does not provide true labels for test data; thus, we consider the validation data as query videos in the video retrieval task. For video data preprocessing, we applied uniform frame sampling to sample 16 frames in each video and resized the sampled frames to 112×112.

B. EVALUATION METRICS

To evaluate the video retrieval performance, we used the top1 and top5 accuracy and mAP for information retrieval as evaluation metrics. The accuracy metric is used here because the PCA-CBVR performance depends on the query video category prediction ability. The mAP for information retrieval is different from that for classification. We used Equation (4) to measure the mAP in the information retrieval context.

$$\text{mAP} = \frac{\sum_{q=1}^{Q} \text{AP}\,(q)}{\text{Q}} \tag{4}$$

where AP is the average precision function and Q is the number of queries. AP is defined as follows:

$$\text{AP} = \frac{\sum_{k=1}^{n} (\text{P}\,(k) \times \text{rel}\,(k))}{\text{number of relevant videos}} \tag{5}$$

where P is the precision function that returns the cut-off k precision, rel is a masking function that returns 1 if the video at k is relevant or 0 otherwise, and n is the number of retrieved videos. In the PCA-CBVR method, the AP result would be 1 or 0; therefore, the mAP of PCA-CBVR is identical to the top1 accuracy.

C. PERFORMANCE ANALYSIS DEPENDING ON FEATURE EXTRACTORS USING DIFFERENT COMBINATIONS OF 3D CNNs AND DEPTHS

To select the best 3D CNN feature extractor, we evaluated the PCA-CBVR performance with different feature extractors which were R3D and R(2+1)D models pre-trained on the Kinetics 700 dataset with different depths. Figure 5 shows the PCA-CBVR results with the different combinations of feature extractors. As shown in Figure 5, when we applied the R3D model as a feature extractor, it outperformed compared to when we applied the R(2+1)D model, and the deeper

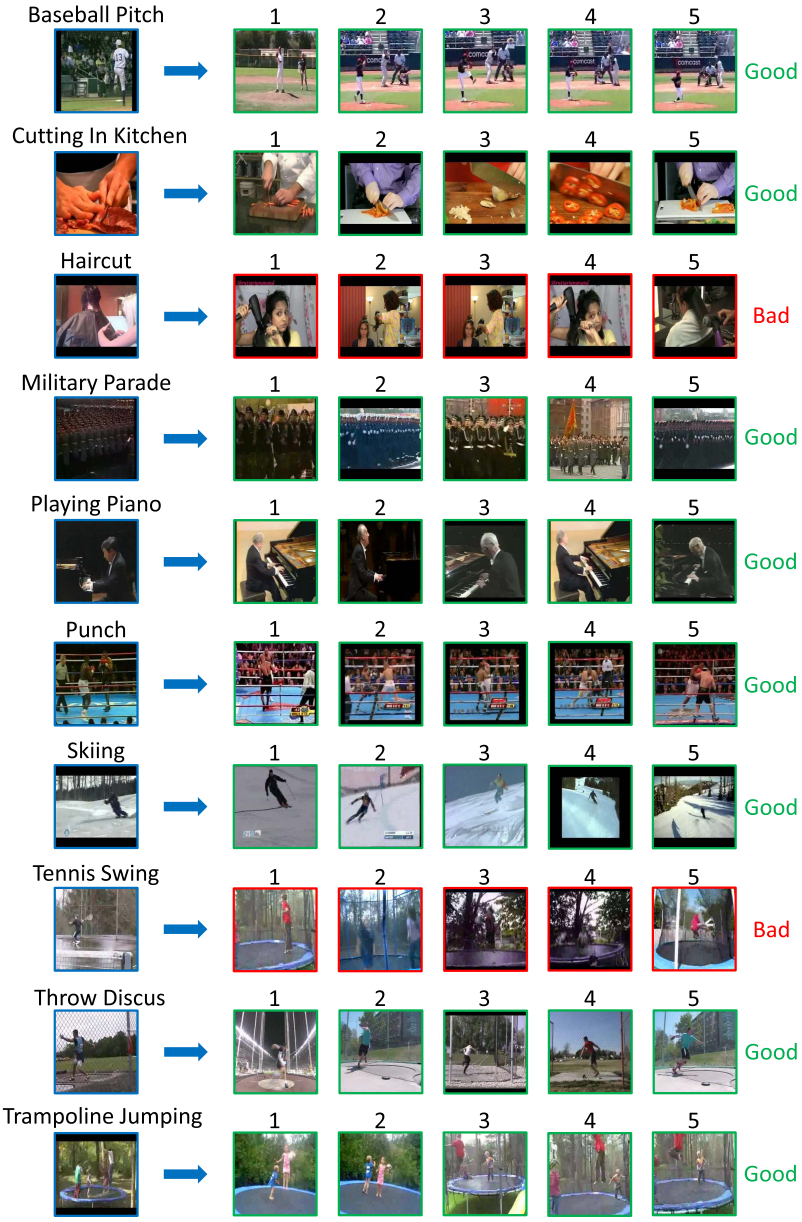


FIGURE 6. The five retrieval video results of PCA-CBVR on each of ten example user query videos from the UCF101 dataset. The blue boxes (left side) are the user query videos, the green boxes are successful retrieval results, and the red boxes are failed retrieval result. The PCA-CBVR results depend on the classification performance; thus, if PCA-CBVR fails to find the user's query video category, it would then retrieve incorrect videos.

networks showed better performance as well. The best performance overall was achieved when we applied R3D50 (R3D model with 50 depth layers) as a feature extractor. The overall performance of PCA-CBVR on the ActivityNet dataset was not as good compared to the outcomes on the UCF101 and HMDB51 datasets, because the ActivityNet dataset is an untrimmed dataset, on the other hand, the UCF101 and HMDB51 datasets are trimmed datasets. In other words, the ActivityNet dataset has much noisier frames which are not closely related to the video context compared to the UCF101 and HMDB51 datasets.

TABLE 5. The mAP comparison results of PCA-CBVR with the R3D50 feature extractor without any fine-tuning and state-of-the-art CBVR methods.

Methods	UCF101	HMDB51	ActivtyNet
Event-Oriented Video Retrieval [17]	**0.83**	**0.75**	-
Stacked HetConv-MK-BiDLSTM [16]	-	-	0.17
PCA-CBVR(without fine-tuning)	0.81	0.57	**0.35**

Table 5 shows the mAP results of the proposed PCA-CBVR with the R3D50 feature extractor, which shows the best performance as shown in Figure 5, and other

TABLE 6. The PCA-CBVR results, which are mAP (top5 accuracy), from the cross-domain (trained on UCF101 and evaluated on ActivityNet) evaluation depending on the number of fine-tuned residual blocks with specific models and learning algorithms. The red and blue values indicate the best mAP and top5 accuracy for each model, respectively.

Models	Learning Algorithm	Number of Residual Blocks			
		1 Block	2 Blocks	3 Blocks	4 Blocks
R3D18	Categorical learning	0.296(0.562)	0.272(0.525)	0.264(0.523)	0.269(0.524)
	Few-Shot learning (5 Way 1 Shot)	0.322(0.58)	0.325(0.576)	0.33(0.587)	0.328(0.589)
	Few-Shot learning (5 Way 5 Shot)	0.323(0.582)	0.33(0.59)	0.327(0.576)	0.336(0.596)
	Few-Shot learning (5 Way 10 Shot)	0.319(0.581)	0.332(0.589)	**0.337**(**0.601**)	0.333(0.584)
R3D34	Categorical learning	0.311(0.57)	0.274(0.535)	0.258(0.512)	0.279(0.529)
	Few-Shot learning (5 Way 1 Shot)	0.34(0.598)	0.34(0.609)	0.35(0.608)	0.34(0.598)
	Few-Shot learning (5 Way 5 Shot)	0.339(0.605)	0.338(0.602)	0.343(0.599)	0.351(0.613)
	Few-Shot learning (5 Way 10 Shot)	0.336(0.6)	0.343(**0.608**)	**0.352**(0.607)	0.336(0.587)
R3D50	Categorical learning	0.315(0.592)	0.322(0.577)	0.294(0.565)	0.315(0.577)
	Few-Shot learning (5 Way 1 Shot)	0.362(0.627)	0.362(0.622)	0.348(0.606)	0.367(0.634)
	Few-Shot learning (5 Way 5 Shot)	0.368(0.634)	0.359(0.62)	0.367(0.631)	**0.37**(**0.637**)
	Few-Shot learning (5 Way 10 Shot)	0.365(0.633)	**0.37**(**0.637**)	0.366(0.629)	0.358(0.619)
R(2+1)D18	Categorical learning	0.231(0.474)	0.241(0.485)	0.272(0.538)	0.297(0.565)
	Few-Shot learning (5 Way 1 Shot)	0.249(0.492)	0.28(0.533)	0.308(0.565)	0.325(0.59)
	Few-Shot learning (5 Way 5 Shot)	0.248(0.501)	0.281(0.541)	0.323(0.586)	0.333(0.596)
	Few-Shot learning (5 Way 10 Shot)	0.246(0.494)	0.292(0.541)	0.314(0.572)	**0.337**(**0.599**)
R(2+1)D34	Categorical learning	0.255(0.511)	0.264(0.519)	0.272(0.52)	0.3(0.562)
	Few-Shot learning (5 Way 1 Shot)	0.291(0.541)	0.314(0.577)	0.344(0.605)	0.38(0.65)
	Few-Shot learning (5 Way 5 Shot)	0.286(0.542)	0.315(0.575)	0.356(0.618)	0.388(0.655)
	Few-Shot learning (5 Way 10 Shot)	0.288(0.542)	0.321(0.583)	0.357(0.622)	**0.391**(**0.658**)
R(2+1)D50	Categorical learning	0.268(0.531)	0.325(0.6)	0.354(0.625)	0.346(0.617)
	Few-Shot learning (5 Way 1 Shot)	0.305(0.551)	0.339(0.597)	0.361(0.623)	**0.408**(**0.679**)
	Few-Shot learning (5 Way 5 Shot)	0.312(0.558)	0.368(0.628)	0.394(0.658)	0.402(0.667)
	Few-Shot learning (5 Way 10 Shot)	0.311(0.558)	0.355(0.616)	0.391(0.648)	0.399(0.668)

video retrieval methods on different datasets. As shown in Table 5, the PCA-CBVR without any fine-tuning showed a poorer mAP outcome on the trimmed datasets compared to Ullah *et al.* [17] and showed a better mAP outcome on the untrimmed dataset compared to Anuranji and Srimathi [16]. In general, users not only send trimmed videos but also send untrimmed videos as query videos; thus, good performance on an untrimmed dataset such as ActivityNet is an the important point in video retrieval tasks, and the proposed PCA-CBVR showed better performance on an untrimmed dataset. Moreover, when we fine-tuned R3D50 and R(2+1)D50 on the UCF101 dataset and applied the fine-tuned feature extractor to the proposed PCA-CBVR, the corresponding mAP results on the UCF101 dataset were 0.86 and 0.89, respectively, outcomes higher than those in Ullah *et al.* [17]. Thus, if we fine-tuned the 3D CNN feature extractor on the particular dataset and apply it to a video retrieval task on a dataset in the same domain, the retrieval performance would increase. However, the user's query does not always involve the same domain as the database videos. Accordingly, evaluation results from the same domain have less meaning than the video retrieval performance in different domain videos. To solve the cross-domain problem, we conducted more experiments with fine-tuning based on the few-shot learning approach. These results are discussed in Section IV-D.

Another possible reason why the proposed PCA-CBVR without fine-tuning showed poorer mAP results on trimmed datasets compared to Ullah *et al.* [17] is that Ullah *et al.* used a novel deep feature selection mechanism to choose the valuable features or frames. On the other hand, we applied simple uniform frame sampling in this experiment. Thus, we conducted more experiments to increase the performance of PCA-CBVR by applying the salient frame sampling approach instead of uniform frame sampling. These results are discussed in Section IV-F.

Figure 6 shows examples of the success and failure of PCA-CBVR on the UCF101 dataset with ten different query videos. The PCA-CBVR approach retrieved videos similar to the user's query video successfully at a rate of 80%. As shown in Figure 6, failures, i.e., the Haircut and Tennis Swing query videos, occurred when the query video and the retrieved videos have similar backgrounds or activity levels, as the proposed PCA-CBVR classifies the user's query video category only based on the generalized deep features of the videos. These outcomes verify that the proposed PCA-CBVR is easily governed by salient frames or by its video representation ability. Therefore, we applied the salient frame sampling approach and discuss the results in Section IV-F, as mentioned earlier.

D. CROSS-DOMAIN EVALUATION FOR THE PCA-CBVR DOMAIN ADAPTATION ABILITY

The domain of the user's query video is not always identical to that of the database videos. Therefore, we need to solve the cross-domain problem in the video retrieval task by increasing the domain adaptation ability of CBVR methods. In the proposed PCA-CBVR method, we fine-tuned a few 3D CNN feature extractors based on the few-shot learning approach, as the fine-tuning based on few-shot learning is more appropriate to resolve the cross-domain problem compared to that based on categorical learning. We conducted fine-tuning based on categorical learning and few-shot

FIGURE 7. Video retrieval results of the fine search and random selection. Three example queries (Mixing, Shaving, Writing) and five retrieval results from the UCF101 dataset are shown. The blue boxes (left side) are the user's query videos, the green boxes are the fine search retrieval results, and the red boxes are the random selection retrieval results. Fine searching catches more semantic information successfully than the random selection method.

learning on all models up to 100 epochs and 3000 episodes, respectively. Also, we utilized the stochastic gradient descent (SGD) optimizer and the cross-entropy loss function in the training phase; the learning rate, momentum, and weight decay were 1e-3, 0.9, and 1e-3 for categorical learning and 1e-4, 0.9, and 1e-3 for few-shot learning respectively.

To produce the cross-domain problem in the video retrieval task, we assumed that untrimmed dataset was the user's query videos and that the trimmed small-size dataset as the database, meaning that the user's query videos have a different domain from the database videos and that the database videos are not sufficient to train the model. To investigate the video retrieval performance in the cross-domain problem, we fine-tuned the model on the UCF101 dataset, which was considered as containing small-sized database videos, and evaluated it on the ActivityNet dataset, which was considered as containing untrimmed user's query videos. We used 64 batch sizes for the categorical learning algorithm and a 5-way 1-shot, 5-way 5-shot, and 5-way 10-shot scenarios for the few-shot learning algorithm. For few-shot learning, we used the prototypical few-shot learning algorithm proposed by Snell *et al.* [34]. Despite the fact that the categorical and few-shot training strategies are different, we still assign certain constraints to the few-shot learning strategies. In summary, fine-tuning based on categorical learning used 9,537 videos in 101 categories, and fine-tuning based on few-shot learning used 9,283 videos in 71 categories for training. The remaining videos in 30 categories can never be used with fine-tuning based on few-shot learning.

Table 6 shows the PCA-CBVR results (mAP and top5 accuracy) on the cross-domain task, referring to the training of the models on the UCF101 dataset and their evaluation on a different dataset, the ActivityNet dataset in this case. We fine-tuned each model based on the categorical learning and few-shot learning approaches to show that the few-shot learning approach outperforms on the domain adaptation task. We trained different numbers of residual blocks of R3D and R(2+1)D from the bottom of the model to verify how many blocks must be fine-tuned for the best domain adaptation ability. As shown in Table 6, the few-shot learning approach showed better performance than the categorical learning approach, and when we fine-tuned more blocks, the performance increased. In this cross-domain experiment, the overall best performance was achieved when we applied the R(2+1)D50 model with fine-tuning of four blocks with the few-shot learning approach using the 5-way 1-shot scenario.

E. RANDOM SELECTION VS. FINE SEARCHING

After the proposed PCA-CBVR predicts the user's query video category, there is one remained step in the video retrieval task; to return the video most similar to the user's

TABLE 7. The PCA-CBVR performance comparison results between uniform and salient frame sampling, which are mAP (top5 accuracy) outcomes. The red and blue values indicate the best mAP and top5 accuracy outcomes for each model, respectively. For this, we applied 3D CNN models that were pre-trained on kinetics 700 without any fine-tuning.

Datasets	Models	Uniform Frame Sampling	Salient Frame Sampling
UCF101	R3D18	0.777(0.947)	**0.801**(**0.961**)
	R3D34	0.797(0.961)	**0.824**(**0.966**)
	R3D50	0.805(0.96)	**0.826**(**0.97**)
	R(2+1)D18	**0.574**(**0.841**)	0.551(0.826)
	R(2+1)D34	0.687(0.911)	**0.702**(**0.918**)
	R(2+1)D50	**0.698**(**0.913**)	0.683(0.905)
HMDB51	R3D18	0.538(0.839)	**0.62**(**0.89**)
	R3D34	0.547(0.848)	**0.636**(**0.897**)
	R3D50	0.569(0.856)	**0.65**(**0.906**)
	R(2+1)D18	0.344(**0.657**)	**0.383**(0.652)
	R(2+1)D34	0.439(0.741)	**0.496**(**0.786**)
	R(2+1)D50	0.459(0.765)	**0.525**(**0.797**)
ActivityNet	R3D18	0.308(0.558)	**0.419**(**0.683**)
	R3D34	0.322(0.583)	**0.446**(**0.707**)
	R3D50	0.349(0.608)	**0.451**(**0.719**)
	R(2+1)D18	0.202(0.425)	**0.231**(**0.463**)
	R(2+1)D34	0.224(0.449)	**0.319**(**0.563**)
	R(2+1)D50	0.229(0.438)	**0.303**(**0.541**)

TABLE 8. The PCA-CBVR performance comparison results between Euclidean distance and cosine similarity metric, which are mAP (top5 accuracy) outcomes. The red and blue values indicate the best mAP and top5 accuracy outcomes for each model, respectively. For this, we applied 3D CNN models that were pre-trained on kinetics 700 without any fine-tuning.

Datasets	Models	Euclidean Distance	Cosine Similarity
UCF101	R3D18	0.768(0.938)	**0.777**(**0.947**)
	R3D34	0.788(0.957)	**0.797**(**0.961**)
	R3D50	0.803(0.957)	**0.805**(**0.96**)
	R(2+1)D18	0.573(**0.844**)	**0.574**(0.841)
	R(2+1)D34	0.685(0.906)	**0.687**(**0.911**)
	R(2+1)D50	0.688(0.91)	**0.698**(**0.913**)
HMDB51	R3D18	0.523(0.831)	**0.538**(**0.839**)
	R3D34	0.537(0.833)	**0.547**(**0.848**)
	R3D50	0.552(0.846)	**0.569**(**0.856**)
	R(2+1)D18	**0.344**(0.653)	**0.344**(**0.657**)
	R(2+1)D34	0.424(0.727)	**0.439**(**0.741**)
	R(2+1)D50	0.453(0.749)	**0.459**(**0.765**)
ActivityNet	R3D18	0.29(0.539)	**0.308**(**0.558**)
	R3D34	0.307(0.561)	**0.322**(**0.583**)
	R3D50	0.333(0.589)	**0.349**(**0.608**)
	R(2+1)D18	0.191(0.402)	**0.202**(**0.425**)
	R(2+1)D34	0.207(0.427)	**0.224**(**0.449**)
	R(2+1)D50	0.216(0.419)	**0.229**(**0.438**)

query video from the predicted category database videos. There are two possible ways to do this: random selection and fine searching. Random selection retrieves random videos from the predicted category database videos; however, even if they are the videos from the same category, the detailed context of each video can differ. For example, videos in the basketball category are taken from different places, i.e., a street, an indoor court, and an outdoor court. Thus, to retrieve the video most similar to the user's query video, random selection from the predicted category is not feasible.

To retrieve the video most similar to the user's query video more accurately, we apply a fine searching step after category prediction by PCA-CBVR, with a fine search also done based on the deep features calculated from PCA-CBVR. Figure 7 shows typical results of video retrieval based on random selection and fine searching. As shown in Figure 7, fine searching retrieved a video more similar to the user's query video compared to random selection by considering the detailed semantic information. For example, when the user's query video category was predicted as ''Mixing,'' the fine searching approach retrieved videos in the same context, including those with the mixing ingredients, the mixing bowl, and the whisk. On the other hand, the random selection approach retrieved videos from the same category, but they were different in terms of the detailed context, such as different mixing ingredients with different cooking tools. Another example is the ''Shaving'' video. The fine searching approach was able to retrieve videos from the same context, showing a man shaving while using shaving cream. On the other hand, the retrieved videos when using the random selection approach included different context videos, in this case showing a man using an electric razor. The last example is the ''Writing'' video. In this example, the fine search approach retrieved videos in the same context, in which a person writes on a whiteboard. On the other hand, the videos retrieved by the random selection method included those in different contexts, where a person was writing on a blackboard.

F. UNIFORM FRAME SAMPLING VS. SALIENT FRAME SAMPLING

The proposed PCA-CBVR method utilizes prototypes to predict the user's query video category, and the prototypes are generalized features of videos' deep features devised by taking the corresponding mean values. Thus, if there are many outliers in the deep features, this will affect the representation ability of the prototypes for the category. This problem is moderated by the frame sampling method.

To determine the capabilities of the frame sampling method, we conducted experiments to compare the uniform frame sampling with salient frame sampling [40]. Yoon *et al.* [40] proposed the salient frame sampling method; the proposed salient frame sampling method eliminated meaningless and outlier frames from the video by using the mean of all deep features of the frames in the video. Table 7 shows the mAP and top5 accuracy results when applying uniform frame sampling and salient frame sampling to the proposed PCA-CBVR. As shown in this Table 7, the salient frame sampling method [40] shows better results than uniform frame sampling in most cases.

G. EUCLIDEAN DISTANCE VS. COSINE SIMILARITY

To calculate the ranking score, we can consider two simple metrics which are Euclidean distance and cosine similarity. To decide which metric is better, we conducted experiments to compare the PCA-CBVR performance based on Euclidean distance and cosine similarity and Table 8 shows the comparison results. As shown in Table 8, cosine similarity helps boost

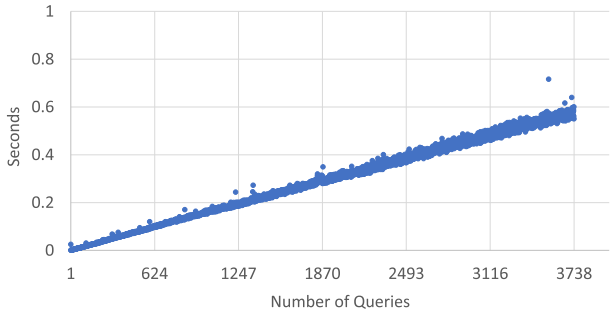


FIGURE 8. Video retrieval time results on UCF101 depending on the number of queries. We excluded additional processes like video load, deep feature extract, and feature load, to analyze pure PCA-CBVR computational cost.

up PCA-CBVR performance compared to Euclidean distance in most case. This means the similarity factor is more appropriate in the proposed PCA-CBVR than the distance factor, thus we applied the cosine similarity instead of Euclidean distance.

H. RETRIEVAL TIME ANALYSIS DEPENDING ON THE NUMBER OF QUERIES

In this subsection, we discuss the proposed PCA-CBVR retrieval time. In these experiments, we excluded irrelevant components to know pure PCA-CBVR performance, such as video load, deep feature extract time, and feature load time. We only included subsections III-A and III-B processing time. We used Intel Xeon Silver 4215R 3.2GHz CPU, Samsung (16GB $\times$ 3) 2,666MHz RAM and Samsung 870 EVO 2TB SSD for experiments. According to Figure 8, the proposed PCA-CBVR computation time is almost linear with a 1/6230 slope depending on the number of queries. Moreover, for the video retrieval process, we utilized a 2k byte (32 bits $\times$ 512) array for 18 and 34 layers 3D CNN models output and an 8k byte (32 bits $\times$ 2048) array for 50 layers 3D CNN models output per each video retrieval in theoretically.

V. CONCLUSION

This paper proposed what is termed the PCA-CBVR method to retrieve videos most similar to users' query videos based on the videos' contexts without any additional information such as tags, among other types. The proposed PCA-CBVR method consists of two main steps: category prediction of the user's query video and fine searching to retrieve the video most similar to each user's query video. To reduce the computational cost while maintaining meaningful information of the videos for the user query video category prediction step, the PCA-CBVR calculates prototypes of the deep features for each category instead of using a dimension reduction strategy or generating binary hash codes as in previous CBVR research. Video category prediction of the user's query based on the prototypes was efficient because there is no need to train the classifier, even when the database is updated. The experimental results here showed that the proposed PCA-CBVR performed better with an untrimmed dataset compared to the outcome state-of-the-art CBVR research, with fine searching based on deep features retrieving the videos most similar to the user's query video by considering the detailed context information. Moreover, to solve the cross-domain problem associated with the CBVR task, we fine-tuned the 3D CNN feature extractor based on the few-shot learning approach, and the PCA-CBVR with fine-tuned feature extractors showed better domain adaptation ability. To improve the performance of the PCA-CBVR, we also applied salient frame sampling to PCA-CBVR instead of uniform frame sampling. As a result, the mAP and top5 accuracy rates were improved. As a future work, we would improve the proposed PCA-CBVR by analyzing 3D CNNs architecture and prototypes property using explainable AI (XAI) techniques and also by utilizing concatenated low level features such as color and texture from frames [41] and trajectory features [10], [42]. Moreover, we would apply the proposed PCA-CBVR to augmented reality (AR) and virtual reality (VR) applications to recommend the proper videos to add and edit the contents based on the user's current situation in real-time.

REFERENCES

[1] *YouTube During COVID-19*. Accessed: Apr. 2, 2021. [Online]. Available: https://www.youtube.com/trends/articles/what-it-means-to-stayhome-on-youtube/

[2] *Cisco Annual Internet Report (2018–2023) White Paper*. Accessed: Apr. 2, 2021. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 41, no. 6, pp. 797–819, Nov. 2011, doi: 10.1109/TSMCC.2011.2109710.

[4] N. Spolaor, H. D. Lee, W. S. R. Takaki, L. A. Ensina, C. S. R. Coy, and F. C. Wu, "A systematic review on content-based video retrieval," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103557, doi: 10.1016/j.engappai.2020.103557.

[5] D. Jain, S. Agrawal, S. Sengupta, P. De, B. Mitra, and S. Chakraborty, "Prediction of quality degradation for mobile video streaming apps: A case study using YouTube," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2016, pp. 1–2.

[6] G. Aceto, G. Bovenzi, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescape, "Characterization and prediction of mobile-app traffic using Markov modeling," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 907–925, Mar. 2021, doi: 10.1109/TNSM.2021.3051381.

[7] Y. Yang, B. C. Lovell, and F. Dadgostar, "Content-based video retrieval (CBVR) system for CCTV surveillance videos," in *Proc. Digital Image Comput., Techn. Appl.*, Dec. 2009, pp. 183–187.

[8] D. Patil and M. A. Potey, "Survey of content based lecture video retrieval," *Int. J. Comput. Trends Technol.*, vol. 19, no. 1, pp. 5–8, Jan. 2015, doi: 10.14445/22312803/ijctt-v19p102.

[9] X. Li, W. Li, and X. Qi, "Research and develope of badminton sports video retrieval system," in *Proc. Int. Conf. Comput. Sci. Netw. Technol.*, Dec. 2011, pp. 1855–1858.

[10] Z. Droueche, M. Lamard, G. Cazuguel, G. Quellec, C. Roux, and B. Cochener, "Content-based medical video retrieval based on region motion trajectories," in *Proc. 5th Eur. Conf. Int. Fed. Med. Biol. Eng.*, Sep. 2011, pp. 622–625.

[11] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Video shot detection and condensed representation. A review," *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 28–37, Mar. 2006, doi: 10.1109/MSP.2006.1621446.

[12] R. Abed, S. Bahroun, and E. Zagrouba, "Face retrieval in videos using face quality assessment and convolution neural networks," in *Proc. IEEE 16th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2020, pp. 399–405.

[13] C. L. Chou, H. T. Chen, and S. Y. Lee, "Pattern-based near-duplicate video retrieval and localization on web-scale videos," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 382–395, Mar. 2015, doi: 10.1109/TMM.2015.2391674.
[14] A. Amir, M. Berg, S.-F. Chang, W. Hsu, G. Iyengar, C.-Y. Lin, M. R. Naphade, A. Natsev, C. Neti, H. Nock, J. R. Smith, B. Tseng, Y. Wu, and D. Zhang, "IBM research TRECVID-2003 video retrieval system," in *Proc. TREC Video Retr. Eval.*, Nov. 2003, pp. 1–18.
[15] S.-I. Yu, L. Jiang, Z. Xu, Y. Yang, and A. G. Hauptmann, "Content-based video search over 1 million videos with 1 core in 1 second," in *Proc. 5th ACM Int. Conf. Multimedia Retr.*, Jun. 2015, pp. 419–426.
[16] R. Anuranji and H. Srimathi, "A supervised deep convolutional based bidirectional long short term memory video hashing for large scale video retrieval applications," *Digit. Signal Process.*, vol. 102, Jul. 2020, Art. no. 102729, doi: 10.1016/j.dsp.2020.102729.
[17] A. Ullah, K. Muhammad, T. Hussain, S. W. Baik, and V. H. C. De Albuquerque, "Event-oriented 3D convolutional features selection and hash codes generation using PCA for video retrieval," *IEEE Access*, vol. 8, pp. 196529–196540, 2020, doi: 10.1109/ACCESS.2020.3029834.
[18] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[19] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.
[20] H. Kataoka, T. Wakamiya, K. Hara, and Y. Satoh, "Would mega-scale datasets further enhance spatiotemporal 3D CNNs?" 2020, *arXiv:2004.04968*.
[21] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2019.
[22] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang, "Cross-domain few-shot classification via learned feature-wise transformation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2020.
[23] O. Russakovsky, J. Deng, H. Su, and J. Krause, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[25] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," 2016, *arXiv:1609.08675*.
[26] S.-I. Yu, Y. Yang, Z. Xu, S. Xu, D. Meng, Z. Mao, Z. Ma, M. Lin, X. Li, H. Li, Z. Lan, L. Jiang, A. G. Hauptmann, C. Gan, X. Du, and X. Chang, "Strategies for searching video content with text queries or video examples features, semantic detectors, fusion, efficient search and reranking," *ITE Trans. Media Technol. Appl.*, vol. 4, no. 3, pp. 227–238, Jan. 2016, doi: 10.3169/mta.4.227.
[27] Y. Suo, C. Zhang, X. Xi, X. Wang, and Z. Zou, "Video data hierarchical retrieval via deep hash method," in *Proc. IEEE 11th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2019, pp. 709–714.
[28] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proc. 34th Annu. ACM Symp. Theory Comput. (STOC)*, 2002, pp. 380–388.
[29] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
[30] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.
[31] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," 2019, *arXiv:1907.06987*.
[32] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," 2017, *arXiv:1708.05038*.
[33] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3D residual networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 3154–3160.
[34] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 4077–4087.
[35] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 3630–3638.
[36] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.
[37] K. Soomro, A. Roshan Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
[39] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 961–970.
[40] H. Yoon, Y.-G. Kim, and J.-H. Han, "Salient video frames sampling method using the mean of deep features for efficient model training," in *Proc. Conf. Korean Inst. Broadcast Media Eng.*, Jun. 2021, pp. 318–321.
[41] N. Kayhan and S. Fekri-Ershad, "Content based image retrieval based on weighted fusion of texture and color features derived from modified local binary patterns and local neighborhood difference patterns," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 32763–32790, Aug. 2021, doi: 10.1007/s11042-021-11217-z.
[42] M. Broilo, N. Piotto, G. Boato, N. Conci, and F. G. B. De Natale, "Object trajectory analysis in video indexing and retrieval applications," in *Video Search and Mining*, vol. 287. Berlin, Germany: Springer, 2010, pp. 3–32, doi: 10.1007/978-3-642-12900-1_1.

HYEOK YOON received the B.S. and M.S. degrees in computer engineering from the Korean Academic Credit Bank System and Seoul National University of Science and Technology, Seoul, South Korea, in 2019 and 2021, respectively. Since 2021, he has been working at Elroilab which develops hyperspectral system based on artificial intelligence, where he is currently a Software Engineer. His research interests include machine learning and computer vision.

JI-HYEONG HAN received the B.S. and Ph.D. degrees in electrical engineering from KAIST, Daejeon, South Korea, in 2008 and 2015, respectively. From 2015 to 2017, she was a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon. Since 2017, she has been with the Seoul National University of Science and Technology, Seoul, South Korea, where she is currently an Assistant Professor. Her research interests include machine learning, human-centered intelligent robotics, and human–robot interaction.