An Efficient Content-Based Video Retrieval using Augmented Distance Metrics by Deep Learning

Shubhangini Ugale[1] Dr. Dharamveer Choudhari[2] Dr. Vivek Kapur[3]

(https://orcid.org/0000-0001-8606-9487)

Research Scholar, Guide, Co-guide & Principal and Director

Shubhangini.ugale@raioni.net, dharmaveerc@gmail.com, vivek.kapur@raisoni.net

[1] G.H. Raisoni University, Amravati, MS (India)

[,3] G.H. Raisoni Institute of Engineering and Technology, Nagpur, MS (India)

[2] G.H. Raisoni College of Engineering, Nagpur, MS (India)

Abstract: Deep learning models have paved the way for unsupervised optimization in a wide variety of computer vision applications. Performance of these deep learning models largely depends upon their internal layer design, type of dataset used, and activated augmentation levels. In order to improve this performance for content-based video retrieval (CBVR) applications, researchers have proposed various convolution neural network (CNN) based deep learning models. These models are trained for application specific datasets, and have limited scalability when applied to untrained query video sequences. In such cases, their performance in terms of precision, recall & receiver operating characteristics (ROC) is also limited. To improve this performance, a novel ensemble deep learning model which utilizes VGGNet-16, Dense Net 121, Inception Res Net V2, Mobile V Net, Res Net 101, and Exception Net models is proposed in this paper. The model supports incremental feedback-based learning which is designed using a correlation feature engine. This engine utilizes a novel augmented correlation metric, which combines 18 different distance measures for continuous training set updates. Due to which, the model's performance is incrementally improved after every iterative batch evaluation. The proposed model was tested on UCF101, Open Video, FIVR, Media Graph, IVP, Columbia University Video, and HMDB human video datasets. It was observed that the model is capable of achieving 96.3% precision, 95.8% recall, 98.6% accuracy, and high AUC performance on these datasets. Initial training delay for the model is very large due to use of multiple learning models, which was reduced via use of a novel bio inspired classifier selection model. This model was able to reduce over 34% redundancies, which were introduced during neural network training & classification steps. Due to this redundancy minimization, the training delay was reduced, and was observed to be at par with existing deep learning

models. Moreover, due to use of incremental learning, the proposed model was observed to be highly scalable, and useful for a wide variety of video retrieval applications.

Keywords: Video, retrieval, ensemble, augmentation, deep learning, convolution, variance, distance metric, incremental learning, accuracy

1. Introduction

Modelling of content-based video retrieval (CBVR) applications is a multidomain task, which involves dataset collection, pre-processing, segmentation, feature extraction &selection, clustering, matching, and post-processing. A typical CBVR model is depicted in figure 1, wherein these processes and their typical data flows are visualized. It is observed that input videos are given to a segmentation model, wherein different object-based components and their classification is performed. This classification assists in improving retrieval efficiency via object-to-object mapping& rank evaluation. The detected objects are converted into feature vectors via various feature extraction models, which includes open CNNs, wavelet decomposition, Fourier transforms, etc. These features can be represented using equation 1, and assist in comparing video sequences with reduced complexity via intra-class feature difference minimization [1].

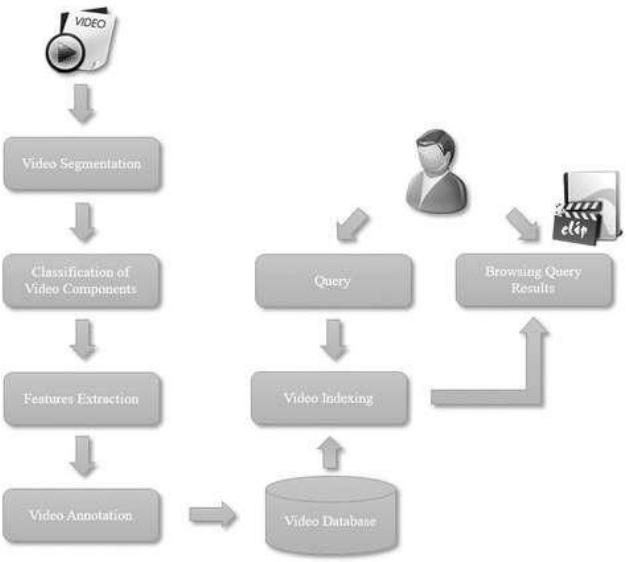

Figure 1. A typical CBVR model

$$F_{out} = \bigcup_{i=1}^{N_{comp}} \left(\sum_{j=1}^{N_{vect}} F_{i,j} * w_j \right) * w_i \ldots (1)$$

Where, $F_{i,j}$ represents feature vectors extracted for feature j of component i in the image, w_i represents weight of the component, w_j represents weight of the extracted vector, N_{comp} represents number of extracted components & objects, while N_{vect}

represents number of vectors extracted for each component. Extracted features are stored in a video database along with an annotated tag, which assists in video identification during retrieval. Query videos are compared with stored videos via object-level feature comparison using deep learning models like CNNs, recurrent neural networks (RNNs), long-short-term-memory (LSTMs), etc. These results are presented to the user, and can also be used for continuous model improvement via feedback learning [2,3]. A wide variety of system models are proposed to perform this task, and each of them have their own nuances, advantages, limitations, and future research scopes. A brief review of these models, and their characteristics is discussed in the next section, which will allow readers to identify different state-of-the-art approaches proposed for CBVR applications. This is followed by section 3, wherein design of the proposed CVRAD2 model, using ensemble deep learning along with augmented distance metrics for improved incremental learning performance is described. This performance is evaluated in section 4, and compared with various state-of-the-art approaches. Finally, this text concludes with some interesting observations about the proposed model, and recommends various fusion & machine learning methods to further improve its performance for different CBVR applications.

2. Literature Review

A wide variety of CBVR models are proposed by researchers over the years. These models vary in terms of computational complexity, accuracy, precision, and other qualitative & quantitative parameters. For instance, the work in [4, 5, 6] proposes deep convolutional neural networks (DCNN) with radial basis loss function (RBF) for low delay CBVR, deep heterogeneous features for facial video retrieval, and ternary hashing-based retrieval model for panoramic video retrieval. These models utilize Deep Neural Network-based designs in order to optimize feature extraction & classification capabilities of CBVR. Performance of these models is further improved via the work in [7], wherein joint video caching with ultra-dense heterogeneous neural networks (JVC UD HNN). The model uses a cascade of NNs to improve feature extraction & selection capabilities, which further assist in improving overall accuracy & precision performance. A similar model is proposed in [8], wherein attention Networks with video hashing (ANVH) are deployed for high-accuracy & moderate delay CBVR. The model uses attention-based networks along with long-short-term Memory Model to improve pattern recognition efficiency, thereby achieving good CBVR performance.

Unsupervised models require lowest deployment complexity, but have moderate CBVR performance. For instance, the work in [9, 10, 11] propose use of deep video hashing with balanced coding, 3D convolutional feature selection with Principal Component Analysis (PCA), and multiple view structure-based action retrieval via deep learning models. These models are highly context-sensitive, and can be applied only to a small-set of applications that support their processing interfaces. Due to which their scalability is limited, but can be improved using the work in [12, 13, 14], wherein researchers have proposed use of different deep learning models like Q-learning, embedded bag-of-features based CNNs, and neighbourhood hash preserving methods are described. These methods utilize recurrent neural networks (RNNs), and similar methods to reduce feature redundancy, and improve accuracy of retrieval. These models have moderate accuracy, but have good recall & precision performance, along with high scalability. To improve their accuracy, application specific models like quadruplet network for face retrieval [15], Motion set network for human Motion retrieval [16], perceptual networks for context-based CBVR [17],

and context-aware online video retrieval [18] are developed. These models allow researchers to deploy high accuracy, application-based CBVR systems, that can be used for real-time scenarios. Similar models are proposed in [19, 20, 21], wherein emotion-based CBVR using reinforcement learning with deep-bidirectional recurrent neural networks (DB RNN RL), multiple streams & multiple modality-based sketch retrieval model, and its extension to incident-based CBVR applications is discussed. These models further assist in identification of various CBVR applications, along with their nuances, advantages, characteristics, and future research scopes when evaluated on different datasets. A study & application of CBVR on such datasets can be observed from [22, 23, 24, 25], wherein distributed CBVR for large-scale videos, high-speed discrete cosine transform-based CBVR for high-density datasets, multimodal video retrieval using low-complexity k means & fuzzy C means for big-datasets, and block truncation coding with edge quantization for high-efficiency CBVR are proposed by researchers. These models are capable of showcasing lower delay, with better throughput, and moderate accuracy CBVR, which makes them suitable for a wide variety of application deployments. Thus, from this discussion it can be observed that a very few CBVR models are suited for large-scale, application-independent scenarios. Motivated by this observation, the next section proposes design of a CBVR model that uses augmented distance metrics & ensemble deep learning for low-delay, and high-accuracy performance. This performance is evaluated in terms of accuracy, precision, recall, &AUC measures, and compared with various state-of-the-art approaches.

3. Proposed content-based video retrieval using augmented distance metrics with ensemble deep learning

From the literature survey it is observed that a wide variety of deep learning models are proposed for high efficiency content-based video retrieval. These models showcase limited accuracy & precision when applied to larger-datasets, which limits their scalability. To improve scalability, continuous learning & train-set trimming layers must be deployed, which would assist in generating optimized training & validation corpuses for different CBVR applications. This section describes design of such a model, which incorporates intensive deep learning (IDL) with augmented distance metrics & uses incremental feedback learning (IFL) for continuous performance enhancement. Overall flow of the proposed model is observed from figure 2, wherein different deep learning layers along with incremental learning & correlation layers are visualized.

The proposed model initially divides entire database into training & validation sets, which is controlled via use of a retraining engine. This engine initially extracts foregrounds from all input frames via Saliency detection, and evaluates their feature sets. These feature sets are clustered via simple k Means model, which assists in set formation. The formed sets are given to a bioinspired model, wherein 6 different deep learning classifiers are used. These classifiers are selected on the basis of their classification capabilities to categorize small, medium and large sequence video features. Results of these classifiers are aggregated and given to a feedback learning layer, which assists in learning rate optimization for the bioinspired model. The retrieved videos from this layer are used for incremental learning, which assists in retrimming input dataset for redundancy reduction & variance maximization, thereby further improving overall efficiency of retrieval. Design of these layers is described in different sections of this text, which will assist readers to implement them in part(s) or as a whole for their own CBVR system designs.

Design of retraining engine

Initially the entire dataset is divided into training & validation sets using a retraining engine. Here, input dataset is given to a Saliency detection model, which assists in high-speed & high-efficiency background separation &foreground extraction from input frames. The input RGB image, is converted into normalized colour vectors via equation 2, 3, and 4 as follows,

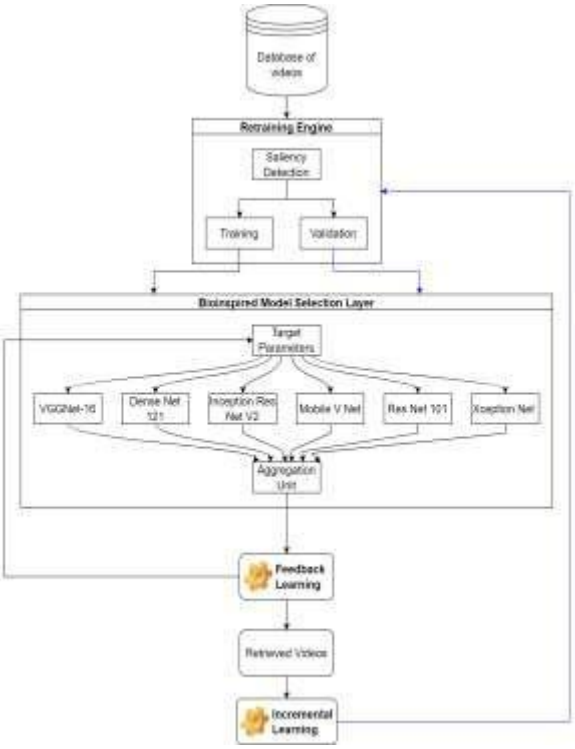

Figure 2. Flow of the proposed CVRAD2 Model for continuous performance improvement

$$R = r - \frac{g+b}{2} \ldots (2)$$

$$G = g - \frac{r+b}{2} \ldots (3)$$

$$B = b - \frac{r+g}{2} \ldots (4)$$

Based, on these values, an intensity component is evaluated via equation 5 as follows,

$$Y = \frac{(r+g)(2-b)}{(2-|r-g|)} \ldots (5)$$

Using these values, relative red & blue components are evaluated via equation 6,

$$RG = R - G, BY = B - Y \ldots (6)$$

These values are used to evaluate quaternions, which assists in energy estimation for input pixels. Quaternions are micro representations of original image pixels, which assist in depicting pixel-to-pixel dependency, thus identifying maximal entropy regions in the input image via equations 7, 8, 9, and 10 as follows,

$$q_0 = \frac{\sqrt{1 + 0.25 * RG + 0.25 * BY + 0.5 * Y}}{2} \dots (7)$$

$$q_1 = \frac{0.25 * (RG - BY)}{2 * Y} \dots (8)$$

$$q_2 = \frac{0.25 * (R + G - B - Y)}{2 * Y} \dots (9)$$

$$q_3 = \frac{0.25 * (B + R + G - Y)}{2 * Y} \dots (10)$$

Where, q_i represents i^{th} quaternion representation of the input image. These representations are given for mean spectrum quality factor (MSQF) evaluation, wherein shifted FFT is used for spectral analysis. The MSQF is evaluated via equation 11 as follows,

$$MSQF_{in} = \sum_{i=1}^{8} \left(ifft \left[fft_{shift} \left(Gaussian \left(\log \left[1 + fft \left(fft \left(q_{in_i} \right) \right) \right] \right) \right) \right] \right)^2 \dots (11)$$

Where, $fft \,\&\, ifft$ represents Fourier and inverse Fourier transform values for input signal. These MSQF values for all quaternions are combined to form entropy values via equation 12,

$$Entropy = \frac{1}{4} \sum_{i=1}^{4} \frac{1}{2 * pi * var(MSQF_i)^2} * \exp \left[\frac{MSQF_i}{2 * var(MSQF_i)^2} \right] \dots (12)$$

Where, varrepresents variance of MSQF data, and is evaluated via equation 13,

$$var(x)^2 = \sum_{i=1}^{N} \frac{\left[x_i - \sum_{j=1}^{N} \frac{x_j}{N} \right]^2}{N} \dots (13)$$

Where, Nrepresents number of elements in input data. Average threshold entropy of all pixels is evaluated via equation 14, and pixels with lower entropy than average threshold are discarded, while others are accepted at the output as foreground pixels.

$$E_{th} = E_{lf} * \frac{\sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{d=1}^{D} Entropy(r, c, d)}{R * C * D} \dots (14)$$

Where, $R, C, \& D$ represents number of rows, number of columns, & number of dimensions for input image, while $E_{lf} \& E_{th}$ represents entropy learning factor, and entropy threshold respectively. Based on this entropy evaluation, the final segmented frame is obtained, which can be observed from figure 3, wherein background regions are removed while foreground regions are retained for better feature extraction.

Figure 3. Results of Saliency detection

Each video frame is passed through these Saliency detection steps, and a segmented image is obtained. These segmented images are given to a feature extraction unit, wherein colour, & shape features are extracted. These features are evaluated only for selection of training & validation sets, which assists in improving efficiency during design of classification engine. The colour map is extracted via equation 15 as follows,

$$C_{map} = \bigcup_{i=1}^{255} \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{d=1}^{D} [I(r,c,d) == i] \dots (15)$$

Where, Irepresents input image, and is also used during edge map extraction via equation 16,

$$E_{map} = \bigcup_{i=1}^{255} \bigcup_{j=1}^{255} \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{d=1}^{D} Canny[[I(r,c,d) = i], [I(r+1,c,d) = j]] == 1 \dots (16)$$

Both these maps are combined to form a super-feature vector (SFV) for each video frame. These vectors are given to a 'N' dimensional k Means model for high-speed clustering into 'k' different clusters. Here, 'N' represents number of features in the SF vector, while 'k' represents total number of categories which are present in the dataset. Main aim of this clustering process is to segregate input feature vectors into multiple non-overlapping groups, and then dividing these groups into training & validation sets for better feature-level computations. The k Means model is controlled via minimization of a distance-based objective function, which is evaluated via equation 17 as follows,

$$D = \sum_{i=1}^{k} \sum_{j=1}^{N} \left| [x_j^i - c_i] \right| \dots (17)$$

Once the algorithm converges, SFVs are clustered into 'k' different clusters, and each of the clusters represents different types of video frames. These clusters are divided into a ratio of 70:30, wherein 70% of frames from each cluster are used for training, while remaining 30% are used for validation & retraining the system model. The training & validation sets are given to a bioinspired model for CNN selection, which is discussed in the next section of this text.

Design of bioinspired model for selection of CNN method

Once the training & validation sets are defined, they are given to a bioinspired model for selection of best performing CNN method. This model uses a modified form of Genetic Algorithm (GA) in order to perform the selection process. The GA is designed using the following process,

- Initialize GA parameters,
  - Number of iterations (N_i)
  - Number of solutions (N_s)
  - Learning rate (L_r)
  - Maximum number of learning models available (Max_{learn})
- Each network is trained using the given training & validation feature sets, and their precision (P), recall (R), accuracy (A), and delay (D) values are evaluated.
- Initially mark all solutions are 'to be changed'
- For each iteration in 1 to N_i
  - For each solution in 1 to N_s

- If the solution is marked as 'not to be changed', then skip it, and go to the next solution
- Else, generate a stochastic solution using the following process,
- Generate a random value for number of classifiers used via equation 18,

$$NC_{used} = rand(2, Max_{learn}) \dots (18)$$

- Select NC_{used} unique random classifiers from pre-trained VGGNet-16, Dense Net 121, Inception Res Net V2, Mobile V Net, Res Net 101, and Xception Net models, and evaluate solution fitness via equation 19,

$$f_i = \frac{\left[\sum_{i=1}^{NC_{used}} \frac{A_i + P_i + R_i}{300} + \frac{Max_{delay}}{Delay_i}\right]}{NC_{used}} \dots (19)$$

- Similarly, find fitness for each solution, and then evaluate fitness threshold via equation 20 as follows,

$$f_{th} = \sum_{i=1}^{N_s} f_i * \frac{L_r}{N_s} \dots (20)$$

  - Mark all solutions with fitness lower than f_{th} as 'to be modified', while pass others to the next iteration
- At the end of final iteration, select the solution with maximum value of fitness, and use the selected classifiers for video retrieval

Each CNN model uses a certain combination of convolutional layers, max pooling layers, and fully connected neural network layers. A typical CNN model architecture can be observed from figure 4, wherein interconnection of these layers is visualized. Feature are extracted using convolutional layers, the outputs of which are controlled via equation 21, wherein a Rectilinear Unit (ReLU) is used for feature activation &analysis.

$$Conv_{out_{i,j}} = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} \sum_{b=-\frac{n}{2}}^{\frac{n}{2}} I(i-a, j-b) * ReLU\left(\frac{m}{2} + a, \frac{n}{2} + b\right) \dots (21)$$

Where, $I, m, n, i, and\ j$ represents input image, rows, columns, current window row, and current window column for the given layer. The number of features generated during each convolution are evaluated via equation 22 as follows,

$$f_{out} = \frac{f_{in} + 2 * p - k}{s} + 1 \dots (22)$$

Where, $f_{in}, f_{out}, p, s, and\ k$ represents number of input features from previous layer, number of output features generated by this layer, convolution padding size, convolution stride size, and kernel size used during convolutions. Extracted features are reduced via Max Pooling operation, wherein variance of feature vectors is evaluated, and based on this variance, similar features are removed from the layers. A threshold is estimated using equation 23, which assists in finding range of features which would be removed. Here average threshold is combined with learning factor to obtain the final threshold level.

$$f_{th} = \left(\frac{1}{X_k} * \sum_{x \in X_k} x^{p_k}\right)^{1/p_k} \dots (23)$$

Where, X_k is size of the image, while p_k is a probability factor, which is tuned in order to modify this threshold value. All feature values more than f_{th} are passed to the next layer, while other values are discarded. This process is repeated multiple times in order to obtain high-level features. These features are given to a fully connected neural network (FCNN) that can classify extracted features into 1 of 'N' classes. The designed network uses a SoftMax activation function in order to perform back propagation-based training, which is controlled via equation 24 as follows,

$$c_{out} = SoftMax\left(\sum_{i=1}^{N_f} f_i * w_i + b\right) \ldots (24)$$

Where, $f_i, w_i, b, and\ N_f$ are values of input feature vector, value of weight, value of bias, and number of features extracted by the convolution layer respectively. All selected classifiers are evaluated based on these operations, and CBVR results are generated.

Based on the retrieval performance of selected classifiers, a feedback learning layer is activated. This layer iteratively improves performance of GA for parameter selection. Design of this layer is discussed in next section of this text.

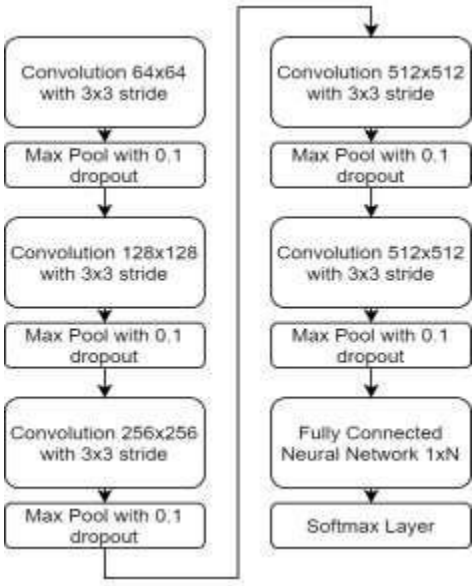

Figure 4. A typical CNN model for CBVR

Design of feedback learning layer

After selection of the best classifier combination for given training & validation datasets, a feedback learning layer is activated. This layer reiterates through all solutions of GA, and identifies optimum value of L_r to achieve higher CBVR performance. The value of L_r is evaluated via equation 25 as follows,

$$New_{L_r} = Old_{L_r}\left(1 + \frac{(New_f - Old_f)}{Min[New_f, Old_f]}\right) \ldots (25)$$

Where, $New_{L_r}, \& Old_{L_r}$ represents new learning rate, and old learning rate, which was obtained during previous learning iteration for the model, while, $New_f, \& Old_f$ represents maximum fitness values obtained during new and old evaluation iterations. Based on these values, the learning rate is continuously modified, and outputs of GA are continuously optimized. These results are given to an incremental learning layer, which assists in re-tuning the dataset in order to obtain better classification performance.

Design of incremental learning layer

Once the GA model is trained, an incremental learning layer is activated. This layer evaluates each retrieved result, and compares it with the query frames using 18 different distance metrics. The reason for selecting multiple metrics for distance evaluation is to get a fine estimate of feature matching between query & retrieved frames. These metrics are evaluated using SFV extracted in section 3.1, and combined via equation 26 as follows,

$$NM(i,j) = \frac{1}{18} * \big(BaC(i,j) + Can(i,j) + Cheb(i,j) + CBlock(i,j) + Corr(i,j) + Cos(i,j) + Euc(i,j) + M(i,j) + Euc^2(i,$$
$$+ Dice(i,j) + Hamming(i,j) + Jac(i,j) + Kul(i,j) + Roger(i,j) + RR(i,j) + SM(i,j) + SS(i,j)$$
$$+ Yule(i,j)\big) \dots (26)$$

Where, $NM(i,j)$ represents the novel metric value between i^{th}retrieved frame, and j^{th}query frame's SFV features,Can indicates Canberra feature Metric,BaC indicates Bray Curtis feature Metric, $Cheb$ indicates Chebyshev feature Metric,$Corr$indicates Correlation feature Metric,$CBlock$ indicates City block feature Metric,Cos indicates Cosine distance feature Metric,M indicates Minkowski feature Metric,Euc indicates Euclidean distance feature Metric,$Dice$ represents dice coefficient feature Metric,Jac indicates Jaccard Distance feature Metric,$Hamming$ indicates Hamming feature Metric,Kul indicates Kulsinski feature Metric,RR indicates Russel Rao feature Metric,$Roger$indicates Rogerstani Moto feature Metric,SMindicates Sokal Michener feature Metric, $Yule$ indicates Yule distance Metric, and SS indicates Sokal Sneath feature Metricvalues. Based on these metrics, a metric threshold is evaluated via equation 27,

$$F_{th} = \partial * \frac{\sum_{i=1}^{N_q}\sum_{j=1}^{N_r} NM(i,j)}{N_S * N_T} \dots (27)$$

Where,$N_q, and\ N_r$ represents number of query images, and number of retrieved images, while,∂represents a tuning factor, which is modified as per temporal performance during CBVR, via equation 28,

$$\partial = \left(1 - \frac{|f_{acc} + f_{prec} + f_{rec}|}{3}\right) \dots (28)$$

Where, $f_{acc}, f_{prec}, f_{rec}$ represents current accuracy, precision, and recall values of CBVR obtained via selected classifier configuration from section 3.2. Frames with $NM > F_{th}$ are discarded, while others are added back to the validation database for continuous performance improvement.The proposed model was evaluated on a wide variety of datasets, and its performance was compared with various reviewed models. This performance can be observed from the next section of this text.

4. Result analysis and comparison

The proposed CVRAD2 model is highly scalable, and can be applied to a wide variety of CBVR datasets. This is because of its incremental learning & feedback learning characteristics. The model was evaluated on the following datasets, and parametric evaluation of precision, recall, accuracy, area under the curve (AUC), and processing delay was compared with DCNN [4], ANVH [8], and DB RNN RL [19],

- UCF101 can be downloaded from, https://www.crcv.ucf.edu/data/UCF101.php
- Open Video dataset from IBM, available at https://research.ibm.com/haifa/projects/imt/video/Video_DataSet.shtml
- FIVR 200K dataset, can be downloaded from, https://github.com/MKLab-ITI/FIVR-200K
- Media Graph database, can be downloaded from https://media.xiph.org/video/
- IVP database, available at, http://ivp.ee.cuhk.edu.hk/research/database/subjective/)
- Columbia University Video dataset, can be downloaded from https://www.ee.columbia.edu/ln/dvmm/columbia374/
- HMDB human video dataset, can be downloaded from https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/

All these sets were combined in order to form a cumulative dataset of 800k video samples, and was converted into 15 fps via video compression. From each video sample, 4 seconds of footage was extracted, to create a dataset of 4.8 million image samples. These samples were used in a ratio of 70:20:10, where 70% of these samples were used for training, 20% for testing, and remaining 10% for validations. Based on this configuration, results of accuracy were evaluated via equation 28 as follows,

$$A = (N_{c_{train}} + N_{c_{test}} + N_{c_{val}}) * \frac{100}{N_t} \dots (28)$$

Where, $N_c \& N_t$ represents number of correctly retrieved& total number of retrieved samples

Due to use of IFL and IDL the proposed model is capable of improving the accuracy by 10% when compared with DCNN [4], 7.6% when compared with ANVH [8], and 7.4% when compared with DB RNN RL [19], thus making the model applicable for a wide variety of high-accuracy CBVR applications. Similarly, results of precision were evaluated via equation 29 as follows,

$$P = t_p * \frac{100}{t_p + f_p} \dots (29)$$

Where, $t_p \& f_p$ represents true positive and false positive rates, here, true positive indicates number of correctly retrieved samples that belong to the same category, while false positive indicates total number of incorrectly retrieved samples. This precision was tabulated w.r.t. number of testing & validation samples (NS) in table 2 as follows,

NS	P (%) DCNN [4]	P (%) ANVH [8]	P (%) DB RNN RL [19]	P (%) CV RAD2
10k	81.29	81.49	84.97	89.76
20k	81.52	81.90	85.34	90.13

Table 2. Precision of CBVR different models

Due to use of IFL, IDL along with bioinspired model for CNN selection, the proposed method is capable of improving the precision by 10.5% when compared with DCNN [4], 7.2% when compared with ANVH [8], and 5.8% when compared with DB RNN RL [19], thus making the model applicable for a wide variety of high-precision CBVR applications. Similarly, results of recall were evaluated via equation 30 as follows,

$$R = t_p * \frac{100}{t_p + f_n} \dots (30)$$

Where, $t_p \& f_n$ represents true positive and false negative rates, here, true positive indicates number of correctly retrieved samples that belong to the same category, while false negative indicates total number of incorrectly retrieved samples with belong to incorrect classes. This recall was tabulated with respect to number of testing & validation samples (NS) in table 3 as follows,

NS	R (%) DCNN [4]	R (%) ANVH [8]	R (%) DB RNN RL [19]	R (%) CV RAD2
10k	82.35	79.70	82.53	89.35
20k	82.60	80.25	82.75	89.72
30k	82.89	81.09	83.00	90.22
40k	83.20	82.11	83.31	90.82

Table 3. Recall of CBVR different models

Due to use of IFL, IDL along with bioinspired model for CNN selection, the proposed method is capable of improving the recall by 8.5% when compared with DCNN [4], 6.8% when compared with ANVH [8], and 9.6% when compared with DB RNN RL [19], thus making the model applicable for a wide variety of high-recall CBVR applications. Similarly, results of AUC were tabulated w.r.t. number of testing & validation samples (NS) in table 4 as follows,

NS	AUC (%) DCNN [4]	AUC (%) ANVH [8]	AUC (%) DB RNN RL [19]	AUC (%) CV RAD2
10k	83.08	81.12	86.01	90.90
20k	83.34	81.73	86.24	91.30
30k	83.63	82.61	86.51	91.82
40k	83.98	83.66	86.83	92.45

Table 4. AUC of CBVR different models

Due to use of IFL, IDL along with bioinspired model for CNN selection, the proposed method is capable of improving the AUC by 9.5% when compared with DCNN [4], 6.8% when compared with ANVH [8], and 8.6% when compared with DB RNN RL [19], thus making the model applicable for a wide variety of high AUC-based CBVR applications. Due to use of bioinspired model that utilizes delay as fitness function, the proposed CVRAD2 model is capable of reducing computational delay by 10% when compared with DCNN [4],

8% when compared with ANVH [8], and 9.5% when compared with DB RNN RL [19], thereby making it useful for a wide variety of high-speed CBVR applications. These improvements are possible due to design of the bioinspired model, along with IFL and IDL methods, which assists in accuracy-aware implementation of CBVR systems.

5. Conclusion and future scope

The proposed model uses a combination of effective dataset clustering, with ensemble deep neural network classifiers & augmented distance metrics to improve efficiency of CBVR for multiple datasets. This model also uses a combination of IFL & IDL in order to incrementally improve its performance w.r.t. number of evaluated video samples. Due to this combination, the proposed CVRAD2 model is capable of achieving 96.3% precision, 95.8% recall, 98.6% accuracy, and 97.08% AUC for different datasets. This performance is incrementally improved with respect to number of tested image samples, which assists in deploying the model for large-scale CBVR applications. The model is also capable of retrieving images with a delay of less than 13 ms per sample, which makes it useful for high-speed CBVR scenarios, thereby further expanding its scalability. This performance was compared with different state-of-the-art models, and it can observed from figure 5,that the proposed model is capable of improving the accuracy by 10% when compared with DCNN [4], 7.6% when compared with ANVH [8], and 7.4% when compared with DB RNN RL [19], while, the proposed method is capable of improving the AUC by 9.5% when compared with DCNN [4], 6.8% when compared with ANVH [8], and 8.6% when compared with DB RNN RL [19], thus suggesting its use in high AUC-based CBVR application scenarios.

Furthermore, the proposed model is capable of reducing computational delay by 10% when compared with DCNN [4], 8% when compared with ANVH [8], and 9.5% when compared with DB RNN RL [19], thereby making it useful for a wide variety of high-speed CBVR applications. Similar observations were made for precision, and recall, which makes the model highly useful for deploying efficient CBVR applications. In future, researchers can extend performance of this model using Q-learning, reinforcement learning, and recurrent Neural Network-based classifier implementations, which can be added with GA to facilitate better retrieval rates during testing & validation phases. Furthermore, the proposed model's delay performance can be further improved via use of feature selection & reduction techniques, which would further assist in incremental accuracy improvement for context-sensitive CBVR scenarios.

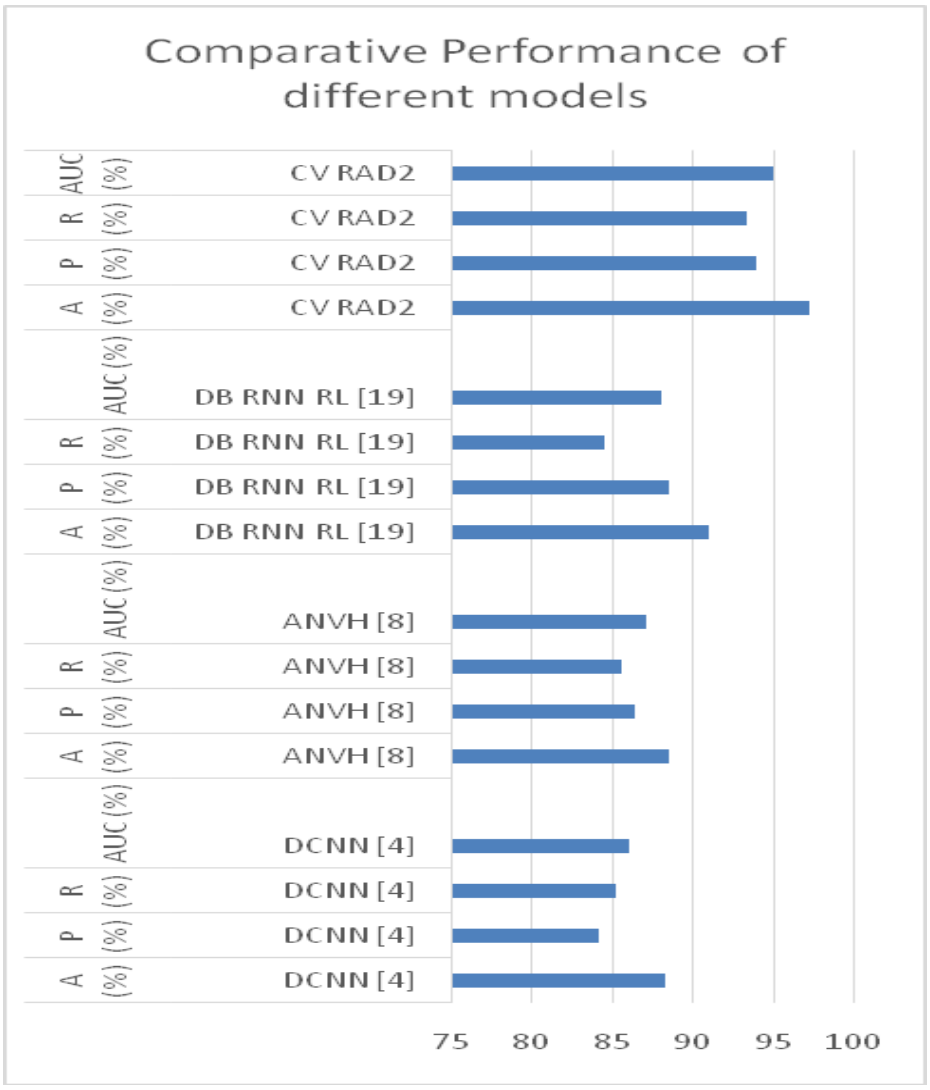

Figure 5. Performance comparison of different models

References

[1] M. Qi, J. Qin, Y. Yang, Y. Wang and J. Luo, "Semantics-Aware Spatial-Temporal Binaries for Cross-Modal Video Retrieval," in IEEE Transactions on Image Processing, vol. 30, pp. 2989-3004, 2021, doi: 10.1109/TIP.2020.3048680.

[2] X. Li, F. Zhou, C. Xu, J. Ji and G. Yang, "SEA: Sentence Encoder Assembly for Video Retrieval by Textual Queries," in IEEE Transactions on Multimedia, vol. 23, pp. 4351-4362, 2021, doi: 10.1109/TMM.2020.3042067.

[3] Z. Dong, J. Wei, X. Chen and P. Zheng, "Face Detection in Security Monitoring Based on Artificial Intelligence Video Retrieval Technology," in IEEE Access, vol. 8, pp. 63421-63433, 2020, doi: 10.1109/ACCESS.2020.2982779.

[4] Y. R. Choi and R. M. Kil, "Face Video Retrieval Based on the Deep CNN With RBF Loss," in IEEE Transactions on Image Processing, vol. 30, pp. 1015-1029, 2021, doi: 10.1109/TIP.2020.3040847.

[5] S. Qiao, R. Wang, S. Shan and X. Chen, "Deep Heterogeneous Hashing for Face Video Retrieval," in IEEE Transactions on Image Processing, vol. 29, pp. 1299-1312, 2020, doi: 10.1109/TIP.2019.2940683.

[6] W. Jing, D. Zhang and H. Song, "An Application of Ternary Hash Retrieval Method for Remote Sensing Images in Panoramic Video," in IEEE Access, vol. 8, pp. 140822-140830, 2020, doi: 10.1109/ACCESS.2020.3006103.

[7] T. Zhang and S. Mao, "Joint Video Caching and Processing for Multi-Bitrate Videos in Ultra-Dense HetNets," in IEEE Open Journal of the Communications Society, vol. 1, pp. 1230-1243, 2020, doi: 10.1109/OJCOMS.2020.3018681.

[8] Y. Wang, X. Nie, Y. Shi, X. Zhou and Y. Yin, "Attention-Based Video Hashing for Large-Scale Video Retrieval," in IEEE Transactions on Cognitive and Developmental Systems, vol. 13, no. 3, pp. 491-502, Sept. 2021, doi: 10.1109/TCDS.2019.2963339.

[9] G. Wu et al., "Unsupervised Deep Video Hashing via Balanced Code for Large-Scale Video Retrieval," in IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 1993-2007, April 2019, doi: 10.1109/TIP.2018.2882155.

[10] 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," in IEEE Access, vol. 8, pp. 196529-196540, 2020, doi: 10.1109/ACCESS.2020.3029834.

[11] K. Zhang, H. Sun, W. Shi, Y. Feng, Z. Jiang and J. Zhao, "A Video Representation Method Based on Multi-View Structure Preserving Embedding for Action Retrieval," in IEEE Access, vol. 7, pp. 50400-50411, 2019, doi: 10.1109/ACCESS.2019.2905641.

[12] L. Rossetto et al., "Interactive Video Retrieval in the Age of Deep Learning – Detailed Evaluation of VBS 2019," in IEEE Transactions on Multimedia, vol. 23, pp. 243-256, 2021, doi: 10.1109/TMM.2020.2980944.

[13] K. Liao et al., "IR Feature Embedded BOF Indexing Method for Near-Duplicate Video Retrieval," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 12, pp. 3743-3753, Dec. 2019, doi: 10.1109/TCSVT.2018.2884941.

[14] S. Li, Z. Chen, J. Lu, X. Li and J. Zhou, "Neighborhood Preserving Hashing for Scalable Video Retrieval," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8211-8220, doi: 10.1109/ICCV.2019.00830.

[15] G. Ren, X. Lu and Y. Li, "Joint Face Retrieval System Based On a New Quadruplet Network in Videos of Multi-Camera," in IEEE Access, vol. 9, pp. 56709-56725, 2021, doi: 10.1109/ACCESS.2021.3072055.

[16] T. Ren, W. Li, Z. Jiang, X. Li, Y. Huang and J. Peng, "Video-Based Human Motion Capture Data Retrieval via MotionSet Network," in IEEE Access, vol. 8, pp. 186212-186221, 2020, doi: 10.1109/ACCESS.2020.3030258.

[17] S. S. Thomas, S. Gupta and V. K. Subramanian, "Context Driven Optimized Perceptual Video Summarization and Retrieval," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 10, pp. 3132-3145, Oct. 2019, doi: 10.1109/TCSVT.2018.2873185.

[18] Y. Feng, P. Zhou, J. Xu, S. Ji and D. Wu, "Video Big Data Retrieval Over Media Cloud: A Context-Aware Online Learning Approach," in IEEE Transactions on Multimedia, vol. 21, no. 7, pp. 1762-1777, July 2019, doi: 10.1109/TMM.2018.2885237.

[19] A. Tripathi, T. S. Ashwin and R. M. R. Guddeti, "EmoWare: A Context-Aware Framework for Personalized Video Recommendation Using Affective Video Sequences," in IEEE Access, vol. 7, pp. 51185-51200, 2019, doi: 10.1109/ACCESS.2019.2911235.

[20] P. Xu et al., "Fine-Grained Instance-Level Sketch-Based Video Retrieval," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 5, pp. 1995-2007, May 2021, doi: 10.1109/TCSVT.2020.3014491.

[21] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras and I. Kompatsiaris, "FIVR: Fine-Grained Incident Video Retrieval," in IEEE Transactions on Multimedia, vol. 21, no. 10, pp. 2638-2652, Oct. 2019, doi: 10.1109/TMM.2019.2905741.

[22] Saoudi, E.M., Jai-Andaloussi, S. A distributed Content-Based Video Retrieval system for large datasets. *J Big Data* **8,** 87 (2021). https://doi.org/10.1186/s40537-021-00479-x

[23] Hamad, Sumaya & Farhan, Ahmeed & Khudhur, Doaa. (2021). Content based video retrieval using discrete cosine transform. Indonesian Journal of Electrical Engineering and Computer Science. 21. 839. 10.11591/ijeecs.v21.i2.pp839-845.

[24] Prathiba, T. & Selva Kumari, R.Shantha. (2021). Content based video retrieval system based on multimodal feature grouping by KFCM clustering algorithm to promote human–computer interaction. Journal of Ambient Intelligence and Humanized Computing. 12. 10.1007/s12652-020-02190-w.

[25] Yan-Hong Chen, Ching-Chun Chang & Cheng-Yi Hsu (2020) Content-based image retrieval using block truncation coding based on edge quantization, Connection Science, 32:4, 431-448, DOI: 10.1080/09540091.2020.1753174