sensors

MDPI

Article

A Supervised Video Hashing Method Based on a Deep 3D Convolutional Neural Network for Large-Scale Video Retrieval

Hanqing Chen [1,†], Chunyan Hu [2,†], Feifei Lee [1,*], Chaowei Lin [1], Wei Yao [1], Lu Chen [1] and Qiu Chen [3,*]

[1] Shanghai Engineering Research Center of Assistive Devices, School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; hqchen0503@163.com (H.C.); cwlin2019@gmail.com (C.L.); weiyao0721@163.com (W.Y.); lchenshijuesh@163.com (L.C.)

[2] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; hhuchy@163.com

[3] Major of Electrical Engineering and Electronics, Graduate School of Engineering, Kogakuin University, Tokyo 163-8677, Japan

[*] Correspondence: feifeilee@ieee.org (F.L.); q.chen@ieee.org (Q.C.)

[†] Both authors contributed equally to this work.

Abstract: Recently, with the popularization of camera tools such as mobile phones and the rise of various short video platforms, a lot of videos are being uploaded to the Internet at all times, for which a video retrieval system with fast retrieval speed and high precision is very necessary. Therefore, content-based video retrieval (CBVR) has aroused the interest of many researchers. A typical CBVR system mainly contains the following two essential parts: video feature extraction and similarity comparison. Feature extraction of video is very challenging, previous video retrieval methods are mostly based on extracting features from single video frames, while resulting the loss of temporal information in the videos. Hashing methods are extensively used in multimedia information retrieval due to its retrieval efficiency, but most of them are currently only applied to image retrieval. In order to solve these problems in video retrieval, we build an end-to-end framework called deep supervised video hashing (DSVH), which employs a 3D convolutional neural network (CNN) to obtain spatial-temporal features of videos, then train a set of hash functions by supervised hashing to transfer the video features into binary space and get the compact binary codes of videos. Finally, we use triplet loss for network training. We conduct a lot of experiments on three public video datasets UCF-101, JHMDB and HMDB-51, and the results show that the proposed method has advantages over many state-of-the-art video retrieval methods. Compared with the DVH method, the mAP value of UCF-101 dataset is improved by 9.3%, and the minimum improvement on JHMDB dataset is also increased by 0.3%. At the same time, we also demonstrate the stability of the algorithm in the HMDB-51 dataset.

Keywords: video retrieval; 3D CNN; supervised hashing; triplet loss

check for updates

Citation: Chen, H.; Hu, C.; Lee, F.; Lin, C.; Yao, W.; Chen, L.; Chen, Q. A Supervised Video Hashing Method Based on a Deep 3D Convolutional Neural Network for Large-Scale Video Retrieval. *Sensors* **2021**, *21*, 3094. https://doi.org/10.3390/s21093094

Academic Editors: Yong Ju Jung, Joohyung Lee, Giorgio Fumera and S. H. Shah Newaz

Received: 15 March 2021
Accepted: 26 April 2021
Published: 29 April 2021

1. Introduction

In the past several years, video information has been widely used because of its richer content and it is easier to understand compared with other media. Due to the rise of various short video platforms and video sharing websites, a large amount of video information is uploaded to the Internet every day, which is widely shared through various social media and news platforms. People can also use various editing software to edit the video while browsing, such as inserting icons, changing the brightness, resizing, clipping video, and so on, which also increases the amount of video data. This not only leads to copyright infringement, but also makes video retrieval difficult, that is, people can hardly find the best matching videos. Usually, the results retrieved by computer need to be manually selected to find the most appropriate ones, which will increase the workload of users and affect the user experience.

As a result, efficient video retrieval algorithms have become nowadays an important component in the applications of copy detection [1], video recommendations [2], video retrieval [3] and copyright protection [4]. There are two main components in a typical content-based video retrieval (CBVR) system, one is feature extraction and another is similarity comparison. Traditional video feature extraction methods mostly based on individual video frames and hand-crafted features, like local binary patterns (LBP) [5], color histogram [6], or key-point descriptors (like SIFT [7]). For the past few years, CNN has been widely used for the ability of learning rich image representation, and widely used in classification task [8], scene recognition [9,10], object detection [11,12], face recognition [13], and image retrieval [14–16], etc. Not surprisingly, CNN is also used to solve video retrieval problems. However, video retrieval methods using 2D CNN often ignore the spatial-temporal connection between video frames. In order to prove the spatial-temporal consistency, a temporal network [17] is used to embed temporal constraints into the network structure for video retrieval. A temporal Hough voting scheme [18] is introduced to rank the retrieved database videos and estimate the segments that match the query. A method named learning to align and match videos (LAMV) [19] is used for aligning the videos temporally. A video similarity learning network named ViSiL [20] is proposed by first computing frame-to-frame similarity and then video-to-video similarity which avoids feature aggregation before the similarity calculation between videos. A method combining CNN to extract frame features and a recurrent neural network (RNN) to retain the temporal information is also proposed by [21], but RNN is hard to train due to the excessive number of parameters needed.

Compared with other kinds of content-based information retrieval, the difficulty of video retrieval lies in the fact that video information has more features that need a large amount of storage space and computing consumption. Benefiting from the XOR operation in binary space, hashing methods have advantages in retrieval efficiency and memory cost, but but previous hashing methods [22–25] are mostly adopted for image retrieval. Several video hashing methods [1,26] focus on getting better video feature representation instead of learning hashing functions. Many of the latest video hashing methods [27,28] are based on CNN + RNN, and a common problem of those methods is that too many parameters make it hard to train. From this point of view, what we need to solve at present is the problem of video feature extraction and retrieval efficiency in video retrieval.

To handle these problems, we propose in this research a novel method called deep supervised video hashing (DSVH) for large-scale video retrieval. First, we apply a pre-trained 3D CNN model to extract the temporal and spatial features in videos. Then, by using supervised hashing, the hash functions are trained, and features extracted by 3D convolution are mapped to binary space. Finally, by calculating the similarity of query video and dataset video in low dimensional space, the retrieval efficiency and accuracy can be improved greatly. The contributions of the proposed method are listed below:

(1) We design an end-to-end framework for fast video retrieval using the idea of a Deep Supervised Video Hashing. By learning a set of hash functions to transfer video features extract by 3D CNN to a binary space.

(2) We choose a fixed number of frames for each video to represent the characteristics of the entire video, which can greatly reduce the computation.

(3) We apply the idea of transfer learning and use a 3D CNN model with residual links pre-trained on large-scale video dataset to obtain the spatial-temporal features in videos.

(4) We conduct a great quantity of experiments over three datasets to demonstrate that the proposed method outperforms many state-of-the-art methods.

The following sections of the paper are arranged as follows: We begin with introductions of some related video retrieval works in Section 2. We introduce our proposed method in details in Section 3. Subsequently, in Section 4 we describe comprehensive experimental results of three datasets to verify the superiority of our method. At last, in Section 5 we summarize the article and our conclusions are presented.

2. Related Works

2.1. D Convolutional Neural Network

Convolutional neural networks have been used in many fields of computer vision due to their powerful feature processing capabilities, including video retrieval. Siamese convolutional neural network (SCNN) [29] is proposed to process video information, which contains two standard CNN networks for extracting video frame features. Wang et al. [30] propose a method that uses CNN to extract video frame features, and then obtains the video features by sparse coding (SC). In addition, Kordopatis-Zilos et al. [31] use a pre-trained CNN model to get the features of video frames, and then apply metric learning to video retrieval. All of these methods mentioned above have a common drawback is that they ignore the temporal relationship between frames, which leads to insufficient video feature extraction.

2.2. D Convolutional Neural Network

In previous video retrieval methods using CNN for feature extraction, operations are usually carried out on single video frames. In this way, a significant drawback in processing video information is that the relationship between frames is ignored which certainly affects the precision of retrieval. Therefore, by performing 3D convolutions we expect to get spatial-temporal features of videos directly. 3D CNN [32] is first proposed for human action recognition, and then widely used for medical image processing [33] and video information processing [34] due to its spatial and temporal feature extraction ability. Some famous network models have been proposed, such as convolutional 3D [35], pseudo 3D [36], inflating 3D [37] and R (2 + 1) D [38]. As we can see from Figure 1, a 3D convolution operation is performed on the temporal and spatial dimensions of video frames arranged in chronological order and the feature map obtained is usually related to multiple video frames before and after, which can capture well the motion information and retain the features of the videos.

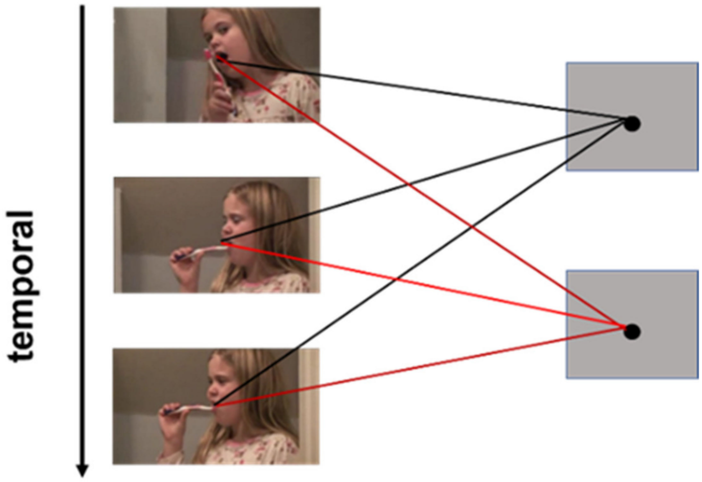

Figure 1. 3D convolution.

2.3. Hashing

Hashing is widely used for a variety of multimedia retrieval tasks with the purpose of transform the information from higher dimensional to lower dimensional, and it has attracted much interest in multimedia retrieval due to its search efficiency and memory saving features. According to whether label information is needed during the training step, we divide the existing hash methods into supervised and unsupervised hashing. Supervised hashing learns compact representations with the help of labels. Well known supervised hashing methods include: supervised hashing with kernels (KSH) [39] by minimizing the inner product of hash codes and minimizing loss hashing (MLH) [40] based on structural SVMs with latent variables and an effective online learning algorithm, etc.

On the contrary, in unsupervised hashing methods, the label information is not necessary for learning hash function. Locality sensitive hashing (LSH) [41] is a typical unsupervised hashing method, which can ensure that the closer to the object, the greater the probability of collision. In addition, other unsupervised methods like iterative quantization (ITQ) [42] are also widely used. Various unsupervised hashing methods [43,44] are used in the field of image retrieval. Some unsupervised hashing methods are also used in video retrieval. Self-supervised video hashing (SSVH) [45] proposed an unsupervised video hashing framework for temporal nature of videos and learning to hash. Unsupervised deep video hashing (UDVH) [46] utilizes feature clustering and a specific rotation to balance the variance of each dimension.

2.4. Video Retrieval

With the development of multimedia information technology, people often need to retrieve videos from the Internet, and video retrieval technology has become a hot topic. Common types of video retrieval include text-based queries [47], audio-based queries [48], and video-based queries [49]. In this paper we focus on retrieving videos through video queries, that is, for a given video, finding similar videos in a database.

The two most important parts of video retrieval are feature extraction and similarity comparison. In recent years, many deep learning-based approaches have been proposed. Kumar et al. [50] propose a video retrieval method with feature extraction combining CNN and RNN. In [49], a neighborhood preserving hashing approach is used for video retrieval with a neighborhood attention mechanism. Furthermore, a central similarity quantization method is employed to mine the central similarity of features [51]. A similarity-preserving deep temporal hashing method is proposed by Shen et al. [27] through CNN + RNN for feature extraction, and a deep metric learning objective named l_2 All_loss based on the improved triplet loss to preserve the similarity within the class and the difference between the classes. In [28], frame-level features are passed to a bidirectional LSTM for temporal information, then a supervised hashing method is employed to get the binary codes of videos. However, most of these methods have too many parameters to be difficult in model training.

3. Proposed Approach

Figure 2 illustrates the proposed framework. The method we proposed is composed of three main components. The first step is to choose representative frames from the short video. The second component is to use a pre-trained 3D CNN to extract video features. The third is to fine-tune the network with the hash layer to get the hash function and then retrieval videos similar to the query.

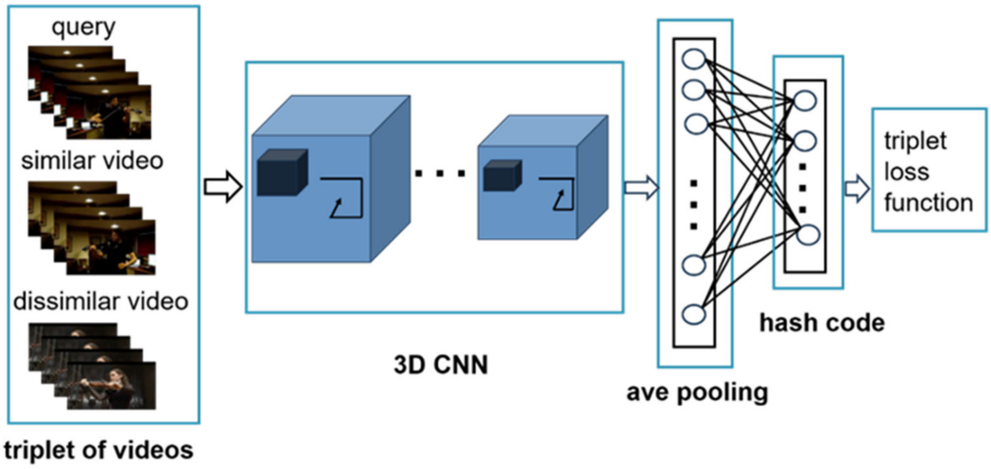

Figure 2. The proposed framework of video retrieval.

3.1. Feature Extraction

3.1.1. Frames Selection

The first step of processing video data is to extract video frames. A video is composed of several video scenes, a video scene is composed of various shots, and a shot is composed of many video frames. As we can see from Figure 3, video frame is the basic unit of a video, even a few seconds long shot video may contain a huge number of video frames. For example, the usual video frame rate is 30 frames per second (FPS), which means that even a one-second video clip contains 30 video frames, but with the development of camera and storage technology, there are more and more videos with high FPS. Figure 4 shows a sample of video frames in UCF-101 dataset. We can see that and there is often only a little difference between adjacent video frames.

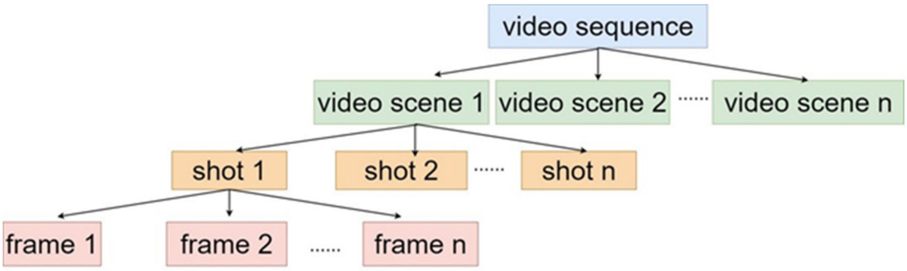

Figure 3. Typical structure of a video.

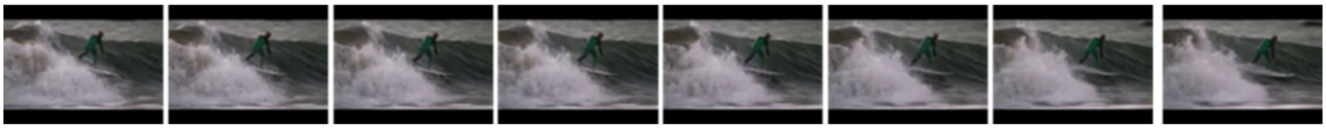

Figure 4. A continuous video frame sample in the UCF-101 dataset.

If we carry out feature extraction on all frames, the amount of computation is too large, and it is not necessary. Therefore, to reduce the computational amount and increase the retrieval efficiency, some representative video frames will be selected for feature extraction by certain strategies defined using Equation (1):

$$f(i) = i\frac{t * FPS}{n}(i = 1, 2, \ldots, n) \tag{1}$$

where t represents the video length in seconds and FPS the frame rate, n is the number of frames selected to represent the video. The schematic diagram of representative frame selection is shown in Figure 5.

3.1.2. Feature Extraction

The CNNs pre-trained on ImageNet are used in various computer vision tasks and achieved great success. The application of CNN models pre-trained on large datasets that uses the idea of transfer learning greatly saves workload, saves time, and reduces the problems caused by insufficient training data. Inspired by the huge success of the pre-trained model, we adopt a pretrained 3D CNN model [52] for feature extraction, which structure and parameters of the network are shown in Figure 6 and Table 1. By applying residual modules [53] to 3D CNNs, we except to improve the retrieval accuracy. Then we fine-tune the model by adding a hash layer with a fully connected structure on the target dataset.

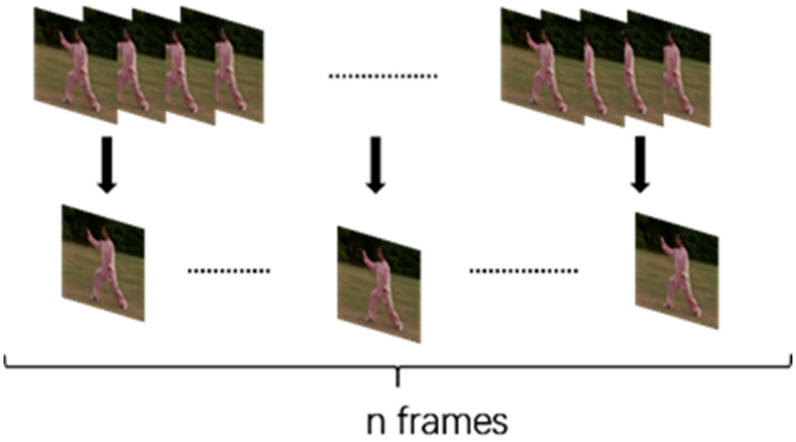

Figure 5. Selection of representative frames.

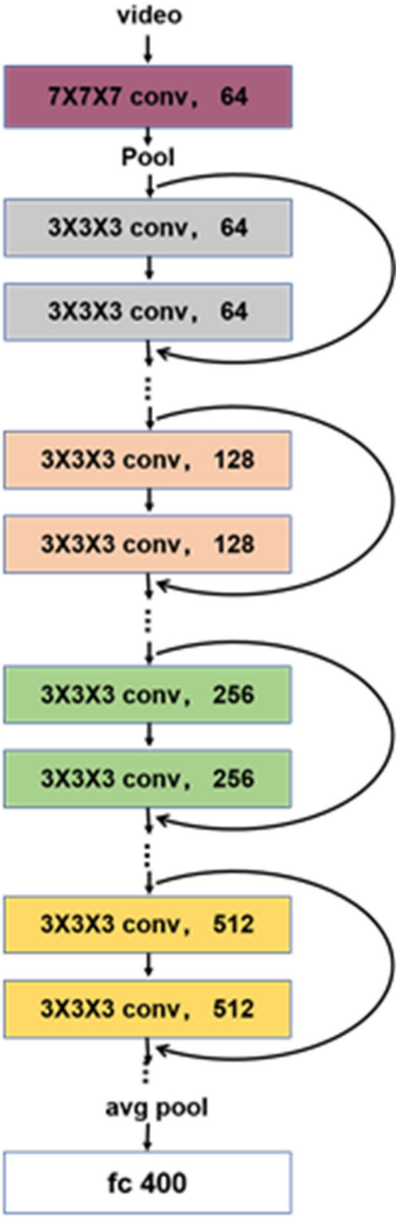

Figure 6. 3D ResNet structure.

Table 1. The details of the network architecture.

Layer Name	Architecture
Conv1	$7 \times 7 \times 7$, 64, stride 1(T), 2(XY)
	$3 \times 3 \times 3$ max pool, stride 2
Conv2_x	$\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times 3$
Conv3_x	$\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times 4$
Conv4_x	$\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times 6$
Conv5_x	$\begin{bmatrix} 3 \times 3 \times 3, & 64 \\ 3 \times 3 \times 3, & 64 \end{bmatrix} \times 3$
	Average pool, 400-d fc, softmax

3.2. Hash Layer

After obtaining video features from the network with stacked convolutional layers, we need to map features extracted by 3D CNN to Hamming space for quick retrieval. We build a hash layer with fully connected structure as show in Figure 7 to map the feature extracted by convolution into binary space and get the representation of video use a k-bit vector. In the training step, in order to limit the features value extracted by 3D CNN to [–1, 1], we use the tanh activation function in the hash layer.

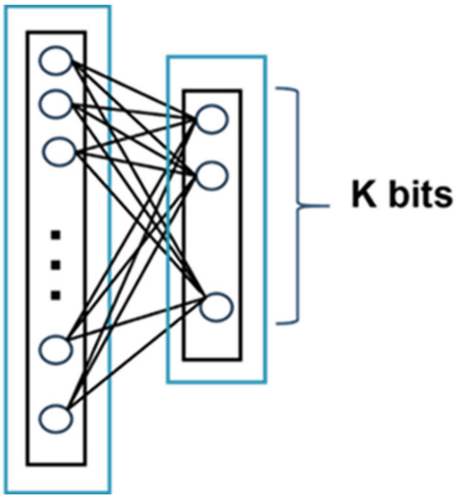

Figure 7. Hash layer structure.

In the test retrieval step, in order to use hamming distance to measure similarity between videos, we need binarize the hash code of videos, so we define Equation (2) to generate the hash code. If the output value of the *j*-th bit is greater than or equal to 0, its corresponding hash code is 1; otherwise, it is -1:

$$H^j = \begin{cases} 1, & out^j \geq 0 \\ -1, & otherwise \end{cases} \tag{2}$$

3.3. Loss Function

In our proposed algorithm, we use the triplet ranking loss in [26] during the training step. At present, most supervised hashing methods train with pairs of samples to prove the similarities/dissimilarities of video pairs. These methods design loss functions to preserve pairwise similarity of videos. But some recent studies have shown that triplet-based similarities can get better results than pairwise similarities. In the training step we use the strategy in Section 3.3.1 to form a triple $\left(\mathrm{I}, \mathrm{I}^{+}, \mathrm{I}^{-}\right)$, I is the query video, in the training set, I^{+} and I are from the same category of video, I^{-} and I are from different categories. By training we want to get a mapping F( ), After mapping features to hamming space, the distance between F(I) and $\mathrm{F}\left(\mathrm{I}^{+}\right)$ is less than F(I) and FI^{-}. Gradient descent is required during supervised learning and training, using distance metrics such as hamming distance may cause the loss function to be nondifferentiable. For ease of optimization, we use Euclidian distance as distance metric in training step. In the retrieval phase, the binary codes of the videos are obtained, Hamming distance is used to measure the similarity between the videos due to the fast bit XOR operation. Finally, we define the triplet loss by Equation (3):

$$L = \max\left\{0, \parallel F_i^0 - F_i^+ \parallel_2^2 - F_i^0 - F_i^- \parallel_2^2 + \alpha\right\} \tag{3}$$

where F_i^0, F_i^+ and F_i^- represent the feature vectors of the query video, similar video and dissimilar video, respectively. α is a margin parameter whose purpose is to ensure that there is enough divergence between the query-positive and query-negative. In our experiment, it is uniformly set to 1/4 of the hash code length. We use batch gradient decent to optimize the objective function in Equation (4):

$$\min_{\theta} \sum_{i=1}^{m} L_\theta + \lambda \parallel \theta \parallel_2^2 \tag{4}$$

where θ is the parameter to be solved, L_θ is the triplet loss, m is the quantity of samples in a mini batch, λ is a regularization parameter used to avoid overfitting.

3.3.1. Triplet Selection

For a query video Q, we first select the videos {P_1, P_2, . . . , Pn} that belongs to the same class and shears the same label as it in the dataset in a mini-batch. This creates a set of query-positive video pairs (Q, P_1), (Q, P_2), . . . , (Q, Pn). The rest of the videos {V_1, V_2, . . . , Vn} in the mini-batch are not similar to the query video. Finally, we can form a series of triples consisting of a query-positive video pair and a dissimilar video like (Q, P_1, V_1), (Q, P_1, V_2), . . . , (Q, Pn, Vn), etc.

4. Experiments

In this section, we conduct experiments on three public video datasets to prove the effectiveness of our method. We start with introducing the datasets and the pre-trained network then show our experimental results with comparison to some representative hashing methods on video retrieval task.

4.1. Datasets and Pre-Trained Model

UCF-101 Dataset [54]

This consists of 101 categories realistic videos collected from YouTube. UCF-101 gives the largest diversity of classes among video datasets, the videos are divided into 25 groups, each consisting of 4–7 videos. Groups of videos have some common features, such as similar backgrounds, similar perspectives, and so on. The clip duration for most videos in UCF101 is less than 10 s. The training set of the original dataset is used for supervised learning and test set is used for retrieve.

JHMDB Dataset [55]

This database contains a total of 928 videos grouped into 21 categories. Most videos involve single actions such as throwing, walking and kicking a ball, and each category has 36–55 samples containing 15–40 frames. We follow the setting in [27], 10 videos each category are selected for training, 10 videos each category are selected for query, 20 videos each category are selected as gallery sets.

HMDB-51 Dataset [56]

It consists of a total of 7000 videos in 51 categories collected from YouTube, and each category contains at least 100 samples. The video samples come from a wide range of sources, most of which are from movie clips. The training sets are used for supervised learning and test sets for retrieve.

Pretrained model

We use the pre-trained 3D ResNet model [49] on the Kinetics dataset, which contains 300,000 videos to extract features.

4.2. Experimental Settings and Evaluation Metrics

4.2.1. Experimental Settings

Three datasets are used to measure the performance of our method, split of the datasets is shown in Table 2. According to previous research experience [35,36,52], we use the method in Section 3.1.1 to select n non-overlapped video frames from each video. Then all the video frames are cropped to $h \times w$. Therefore, the final input dimensions are $c \times n \times h \times w$, where c is the channel number. In all of our experiment, we consistent set $n = 16$, $h = w = 112$. All experiments are performed on a workstation equipped with an Intel Xeon E5-2630 v3 CPU, 32 GB RAM and an NVIDIA Tesla K40c GPU. During the training phase, we used the SGD optimizer for gradient descent, the initial learning rate is 10^{-4}, and the learning rate was reduced to $1/10$ of the original rate every 40 epochs. The training epochs of UCF-101, JHMDB, HMDB are 150, 100, 150 the batch size of UCF-101, JHMDB, HMDB are set to 80, 60, 80.

Table 2. Split of datasets.

Datasets	Training Set	Test Set	Gallery Set
UCF-101	9537	3783	-
JHMDB	210	210	410
HMDB-51	3570	1530	-

4.2.2. Evaluation Metrics

Mean Average Precision (mAP)

We employ mean average precision (mAP) to evaluate the performance of the proposed video retrieval algorithm, where average precision (AP) is calculated by Equation (5):

$$AP = \frac{1}{R}\sum_{k=1}^{n}\frac{k}{R_k} \times rel_k \tag{5}$$

where n is the total number of the dataset, R represent the amount of videos that have relations to the retrieval videos in the dataset, and R_k represent the quantity of the similar videos in the top k returns. When the video at position k is similar to the query $rel_k = 1$; otherwise $rel_k = 0$. Finally, mAP is obtained by calculate the mean value of AP.

Precision@N

This represents the proportion of correct retrieval results in the top-N retrieval results. The definition of precision@N is shown in Equation (6):

$$Precision@N = \frac{\sum_{i=1}^{N} rel_k}{N} \tag{6}$$

where N is number of the top-N retrieved results, if the retrieved video similar to the query video $rel_k = 1$; otherwise $rel_k = 0$.

4.3. Experimental Results

To verify the performance of our algorithm, several state-of-the-art video retrieval methods are used to compare, including DVH [57], ITQ [42], DCNNH [3], SPDTH [27], BIDLSTM [28], DBH [22], DNNH [23].

4.3.1. Experimental Results on UCF-101

The experimental results and the comparison with other hashing-based methods on UCF-101 dataset are shown in Table 3a,b. We can see a huge improvement when compared with the traditional hashing method ITQ. DVH, DCNNH, SPDTH are all deep learning-based hashing methods, where DVH obtains frame-wise CNN feature representation by passing a set of video frames to the convolutional and pooling layers, and then temporal fusion is conducted in the fully connected layers. The selection of frames may seriously affect the result. Compared with DVH the mAP and precision of our method increase by 7.7–9.3% and 2.8–4.7%. DCNNH uses CNN to do feature extraction from video frames, and then fushes them into video features by average weighting, thus completely ignoring the temporal information. Compared with the referenced DCNNH use 2D CNN to extract features, the mAP increased by 2.8–4.7%.

Table 3. Experimental results on UCF-101 dataset.

(a) Comparison Results of mAP on UCF-101 Dataset

Methods	**UCF-101 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
ITQ	0.294	0.305	0.306
DVH	0.705	0.712	0.729
DCNNH	0.747	0.783	0.796
SPDTH	0.741	0.750	0.762
DSVH (ours)	**0.798**	**0.801**	**0.806**

(b) Comparison Results of Precision@N on UCF-101 Dataset

Methods	**Precision@30**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
ITQ	0.389	0.419	0.432
DVH	0.734	0.744	0.756
DCNNH	-	-	-
SPDTH	0.751	0.757	0.765
DSVH (ours)	**0.781**	**0.774**	**0.784**

(c) Comparison Results of mAP with Different Network Depths

Networks	**UCF-101 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
3D ResNet-18	0.735	0.756	0.757
3D ResNet-34	**0.798**	**0.801**	**0.806**

(d) Influence of Hash Layer on mAP

Networks	**UCF-101 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
With Hash Layer	**0.798**	**0.801**	**0.806**
Without Hash Layer	0.752	0.787	0.785

SPDTH uses CNN + RNN to extract features and generate hash codes through temporal-aware hashing, so that temporal information can be well preserved, but it is not easy to train due to too many parameters. Compared with SPDTH, our method still achieves an increase by 4.4–5.7% in mAP and 1.7–3.0% in precision with a simpler loss function.

In order to explore the relationship between the network depth and the retrieval accuracy, we also use 3D ResNet-18 under the UCF-101 dataset for the experiment, and the results are shown in Table 3c. It can be clearly seen from the table that a deeper network achieves better results. For each length of hash codes, the value of mAP increases by about 5%.

In order to demonstrate the effect of the hash layer, we remove the activation function in the hash layer, and directly use the full connection structure to map the video features into 64, 128 and 256 dimensions, and then compare the result with the previous one with a hash layer as shown in Table 3d. The results show that the value of mAP decreases obviously when the hash layer is removed, which proves the effectiveness of the hash layer.

4.3.2. Experimental Results on JHMDB

The experimental results and the comparison with other hashing-based methods on JHMDB dataset are shown in Table 4a,b. The results on JHMDB are similar to those on UCF101, and our method still has a significant advantage over ITQ and DVH, even compared with the latest method BIDLSTM we also have a slight advantage about 0.3–2% in mAP. BIDLSTM also use a CNN + RNN model to extract video features, then map the features into binary space to get compact binary codes. A CNN model with stack heterogeneous convolutional multi-kernel is used to do feature extraction of the frames, then bidirectional long short term memory (LSTM) network is applied for maintaining the temporal information. Compared with BIDLSTM, the method we proposed is simpler in structure but more efficient for retrieval.

We also use different pre-trained networks on the JHMDB dataset to conduct experiments, and the results are shown in Table 4c. It is obvious from the results that increasing the depth of the network is helpful for retrieval.

The influence of the hash layer is also listed in Table 4d. The results obtained are similar to those on the UCF101 dataset, and the value of mAP decreases obviously after removing the hash layer. The results on the two datasets prove the effectiveness of the hash layer.

Table 4. Experimental results on JHMDB dataset.

(a) Comparison Results of mAP on JHMDB Dataset			
Methods	**JHMDB dataset**		
Bits	**16 Bits**	**32 Bits**	**64 Bits**
DVH	0.332	0.368	0.369
ITQ	0.132	0.145	0.151
SPDTH	0.386	0.438	0.464
BIDLSTM	0.410	0.455	0.482
DSVH (ours)	**0.430**	**0.458**	**0.494**
(b) Comparison Results of Precision@N on JHMDB Dataset			
Methods	**Precision@20**		
Bits	**16 Bits**	**32 Bits**	**64 Bits**
DVH	0.342	0.368	0.382
ITQ	0.152	0.175	0.191
SPDTH	0.375	0.408	0.438
BIDLSTM	-	-	-
DSVH (ours)	**0.397**	**0.430**	**0.469**

Table 4. *Cont.*

(c) Comparison Results of mAP with Different Network Depths			
Networks	**JHMDB Dataset**		
Bits	**16 Bits**	**32 Bits**	**64 Bits**
3D ResNet-18	0.371	0.411	0.444
3D ResNet-34	**0.397**	**0.430**	**0.469**

(d) Influence of Hash Layer on mAP			
Networks	**JHMDB Dataset**		
Bits	**16 bits**	**32 bits**	**64 bits**
With Hash Layer	**0.397**	**0.430**	**0.469**
Without Hash Layer	0.320	0.384	0.416

4.3.3. Experimental Results on HMDB-51

The experimental results and the comparison with other hashing-based methods on HMDB-51 dataset are shown in Table 5a. Where, DBH and DNNH use the ResNet50 model for feature extraction of video frames, which is essentially a frame-based method. Compared with them, our method has obvious advantages due to the use of video-based features. The mAP of our method increases at least 6.2% compared with DNNH at 128 bits, and increases by 18.9% at most compared with DBH at 256 bits.

Table 5. Experimental results on HMDB-51 dataset.

(a) Comparison Results of mAP on HMDB-51 Dataset			
Methods	**HMDB-51 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
ITQ	0.408	0.416	0.436
DBH	0.389	0.391	0.386
DNNH	0.487	0.503	0.493
DCNNH	0.458	0.451	0.467
DSVH (ours)	**0.562**	**0.565**	**0.575**

(b) Comparison of Different N on Precision@ Based HMDB-51 Dataset			
Dateset	**HMDB-51 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
Precision@10	0.522	0.531	0.537
Precision@50	0.513	0.519	0.530
Precision@100	0.405	0.408	0.413

(c) Comparison mAP of Different Frames on HMDB-51 Dataset			
Dateset	**HMDB-51 Dataset**		
Bits	**64 Bits**	**128 Bits**	**256 Bits**
12 frames	0.490	0.516	0.539
16 frames	**0.562**	**0.565**	**0.575**
20 frames	0.527	0.550	0.556

In order to prove the robustness of our method, we test the precisions of top-10, top-50 and top-100 respectively, the results are shown in Table 5b. It can be seen from Table 5b that when N increases from 10 to 50, the precision basically does not decline, but when N increases to 100, the precision begins to decline. This indicates that our method has stable precision for the videos in the front of the retrieval results.

In order to verify the influence of the number of video frames n on the performance of our method, we test the mAP when n equal to 12,16,20 respectively, the results are shown in Table 5c. The results show that when we extract 20 frames of each video, the mAP does not increase, but decreases. Increasing the sampling frequency of video frames does not necessarily improve the retrieval accuracy.

All the results prove that DSVH outperform many advanced video retrieval methods not only in accuracy but also in stability By selecting video frames and using 3D CNN, we can directly obtain the spatial-temporal feature in videos, and then get compact binary codes of the whole video under the supervised training, without considering how to maintain the temporal information after the convolution operation like other frame-based video retrieval methods, the use of pre-trained model also saves a lot of work and time.

5. Conclusions

In this paper, we have proposed an efficient video retrieval method named DSVH, which combines 3D convolution and supervised hashing for efficient video retrieval. By using 3D convolution for feature extraction from a series of video frames arranged in chronological order, and then using the supervised hashing method to map video features into Hamming space for similarity comparison, the retrieval performance is greatly improved. Compared with traditional methods such as ITQ [42], the mAP of our method increases at least 13.9% on HMDB-51, and at most 50.4% on UCF-101. Moreover, compared with the representative deep learning-based method SPDTH [27], the mAP of our method increases by 4.4–5.7%; 0.3–2% on UCF and JHMDB, and precision@N increases by 1.7–3%; 2.2–3.1% on UCF and JHMDB, respectively. A series of experimental results prove that our algorithm has superiority over many state-of-the-art video retrieval methods. For long video retrieval, we consider using more frames to extract features for each video in the future to achieve satisfactory results.

Author Contributions: Methodology, H.C. and C.H.; Software, H.C. and C.H.; Data curation, C.L. and W.Y.; Validation, C.H. and L.C.; Writing—original draft preparation, H.C.; Writing—review and editing, F.L. and Q.C.; Supervision, F.L. and Q.C.; Funding acquisition, F.L. and Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Programme for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and also partially supported by JSPS KAKENHI Grant Number 15K00159.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Douze, M.; Jégou, H.; Schmid, C. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Trans. Multimed.* **2010**, *12*, 257–266. [CrossRef]
2. Deldjoo, Y.; Elahi, M.; Cremonesi, P.; Garzotto, F.; Piazzolla, P.; Quadrana, M. Content-based video recommendation system based on stylistic visual features. *J. Data Semant.* **2016**, *5*, 99–113. [CrossRef]
3. Dong, Y.; Li, J. Video Retrieval based on Deep Convolutional Neural Network. In Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing (ICMSSP), Shenzhen, China, 28 April 2018; pp. 12–16.
4. Muranoi, R.; Zhao, J.; Hayasaka, R.; Ito, M.; Matsushita, Y. Video retrieval method using shotID for copyright protection systems. In Proceedings of the Multimedia Storage and Archiving Systems III, Boston, MA, USA, 5 October 1998; pp. 245–252.
5. Shang, L.; Yang, L.; Wang, F.; Chan, K.P.; Hua, X.S. Real-Time Large Scale Near-Duplicate Web Video Retrieval. In Proceedings of the 18th ACM International Conference on Multimedia (ACM MM), Firenze, Italy, 25–29 October 2010; pp. 531–540.
6. Wu, X.; Hauptmann, A.G.; Ngo, C.W. Practical Elimination of Near-Duplicates from Web Video Search. In Proceedings of the 15th ACM International Conference on Multimedia (ACM MM), Augsburg, Bavaria, Germany, 24–29 September 2007; pp. 218–227.
7. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]

8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
9. Xie, L.; Lee, F.; Liu, L.; Yin, Z.; Chen, Q. Hierarchical coding of convolutional features for scene recognition. *IEEE Trans. Multimed.* **2020**, *22*, 1182–1192. [CrossRef]
10. Chen, L.; Bo, K.; Lee, F.; Chen, Q. Advanced feature fusion algorithm based on multiple convolutional neural network for scene recognition. *Comput. Model. Eng. Sci.* **2020**, *122*, 505–523. [CrossRef]
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 11–12 December 2015; pp. 91–99.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
13. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
14. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised Hashing for Image Retrieval via Image Representation Learning. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014.
15. Zhao, F.; Huang, Y.; Wang, L.; Tan, T. Deep Semantic Ranking based Hashing for Multi-Label Image Retrieval. In Proceedings of the 2015 IEEE Conference on Computer VISION and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1556–1564.
16. Lu, C.; Lee, F.; Chen, L.; Huang, S.; Chen, Q. IDSH: An improved deep supervised hashing method for image retrieval. *Comput. Model. Eng. Sci.* **2019**, *121*, 593–608. [CrossRef]
17. Tan, H.K.; Ngo, C.W.; Hong, R.; Chua, T.S. Scalable Detection of Partial Near-Duplicate Videos by Visual-Temporal Consistency. In Proceedings of the 17th ACM International Conference on Multimedia (ACM MM), Beijing, China, 19–24 October 2009; pp. 145–154.
18. Douze, M.; Jégou, H.; Schmid, C.; Pérez, P. Compact Video Description for Copy Detection with Precise Temporal Alignment. In Proceedings of the European Conference on Computer Vision (ECCV), Crete, Greece, 5–11 September 2010; pp. 522–535.
19. Baraldi, L.; Douze, M.; Cucchiara, R.; Jégou, H. LAMV: Learning to align and match videos with kernelized temporal layers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7804–7813.
20. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, I. Visil: Fine-Grained Spatio-Temporal Video Similarity Learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6351–6360.
21. Hu, Y.; Lu, X. Learning spatial-temporal features for video copy detection by the combination of CNN and RNN. *J. Vis. Commun. Image Represent.* **2018**, *55*, 21–29. [CrossRef]
22. Lin, K.; Yang, H.F.; Hsiao, J.H.; Chen, C.S. Deep Learning of Binary Hash Codes for Fast Image Retrieval. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 27–35.
23. Lai, H.; Pan, Y.; Liu, Y.; Yan, S. Simultaneous Feature Learning and Hash Coding with Deep Neural Networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3270–3278.
24. Wang, X.; Lee, F.; Chen, Q. Similarity-preserving hashing based on deep neural networks for large-scale image retrieval. *J. Vis. Commun. Image Represent.* **2019**, *61*, 260–271. [CrossRef]
25. Wang, L.; Qian, X.; Zhang, Y.; Shen, J.; Cao, X. Enhancing sketch-based image retrieval by cnn semantic re-ranking. *IEEE Trans. Cybern.* **2019**, *50*, 3330–3342. [CrossRef] [PubMed]
26. Song, J.; Yang, Y.; Huang, Z.; Shen, H.T.; Hong, R. Multiple Feature Hashing for Real-Time Large Scale Near-Duplicate Video Retrieval. In Proceedings of the 19th ACM International Conference on Multimedia (ACM MM), Scottsdale, AZ, USA, 28 November–1 December 2011; pp. 423–432.
27. Shen, L.; Hong, R.; Zhang, H.; Tian, X.; Wang, M. Video retrieval with similarity-preserving deep temporal hashing. *ACM Trans. Multimedia Comput. Commun. Appl.* **2019**, *15*, 1–16. [CrossRef]
28. Anuranji, R.; Srimathi, H. A supervised deep convolutional based bidirectional long short term memory video hashing for large scale video retrieval applications. *Digit. Signal Process.* **2020**, *102*, 102729. [CrossRef]
29. Jiang, Y.G.; Wang, J. Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE Trans. Big Data* **2016**, *2*, 32–42. [CrossRef]
30. Wang, L.; Bao, Y.; Li, H.; Fan, X.; Luo, Z. Compact CNN based Video Representation for Efficient Video Copy Detection. In Proceedings of the International Conference on Multimedia Modeling (MMM), Reykjavik, Iceland, 4–6 January 2017; pp. 576–587.
31. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, Y. Near-Duplicate Video Retrieval with Deep Metric Learning. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 347–356.
32. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [CrossRef]

33. Huang, X.; Shan, J.; Vaidya, V. Lung Nodule Detection in CT Using 3D Convolutional Neural Networks. In Proceedings of the 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI), Melbourne, VIC, Australia, 18–21 April 2017; pp. 379–383.
34. Molchanov, P.; Gupta, S.; Kim, K.; Kautz, J. Hand Gesture Recognition with 3D Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 1–7.
35. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning Spatiotemporal Features with 3d Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 4489–4497.
36. Qiu, Z.; Yao, T.; Mei, T. Learning Spatio-Temporal Representation with Pseudo-3d Residual Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5533–5541.
37. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6299–6308.
38. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.
39. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
40. Norouzi, M.; Fleet, D.J. Minimal Loss Hashing for Compact Binary Codes. In Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; pp. 353–360.
41. Gionis, A.; Indyk, P.; Motwani, R. Similarity Search in High Dimensions via Hashing. In Proceedings of the 25th International Conference on Very Large Data Bases (VLDB), Edinburgh, Scotland, UK, 7–10 September 1999; pp. 518–529.
42. Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2916–2929. [CrossRef]
43. Zhu, L.; Shen, J.; Xie, L.; Cheng, Z. Unsupervised topic hypergraph hashing for efficient mobile image retrieval. *IEEE Trans Cybern.* **2016**, *47*, 3941–3954. [CrossRef]
44. Zhu, L.; Shen, J.; Xie, L.; Cheng, Z. Unsupervised visual hashing with semantic assistant for content-based image retrieval. *IEEE Trans Knowl Data Eng.* **2016**, *29*, 472–486. [CrossRef]
45. Song, J.; Zhang, H.; Li, X.; Gao, L.; Wang, M.; Hong, R. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Trans. Image Process.* **2018**, *27*, 3210–3221. [CrossRef] [PubMed]
46. Wu, G.; Han, J.; Guo, Y.; Liu, L.; Ding, G.; Ni, Q.; Shao, L. Unsupervised deep video hashing via balanced code for large-scale video retrieval. *IEEE Trans. Image Process.* **2018**, *28*, 1993–2007. [CrossRef]
47. Galanopoulos, D.; Mezaris, V. Attention Mechanisms, Signal Encodings and Fusion Strategies for Improved Ad-Hoc Video Search with Dual Encoding Networks. In Proceedings of the 2020 International Conference on Multimedia Retrieval (ICMR), Dublin, Ireland, 8–11 June 2020; pp. 336–340.
48. Prathiba, T.; Kumari, R.S.S. Content based video retrieval system based on multimodal feature grouping by KFCM clustering algorithm to promote human–computer interaction. *J. Ambient Intell. Humaniz Comput.* **2020**, 1–15. [CrossRef]
49. Li, S.; Chen, Z.; Lu, J.; Li, X.; Zhou, J. Neighborhood Preserving Hashing for Scalable Video Retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 8212–8221.
50. Kumar, V.; Tripathi, V.; Pant, B. Learning compact spatio-temporal features for fast content based video retrieval. *Int. J. Innov. Tech. Explor. Eng.* **2019**, *9*, 2402–2409.
51. Yuan, L.; Wang, T.; Zhang, X.; Tay, F.E.; Jie, Z.; Liu, W.; Feng, J. Central Similarity Quantization for Efficient Image and Video Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3083–3092.
52. Hara, K.; Kataoka, H.; Satoh, Y. Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 3154–3160.
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
54. Soomro, K.; Zamir, A.R.; Shah, M. A dataset of 101 human action classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
55. Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; Black, M.J. Towards Understanding Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, NSW, Australia, 1–8 December 2013; pp. 3192–3199.
56. Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A Large Video Database for Human Motion Recognition. In Proceedings of the 2011 International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2556–2563.
57. Liong, V.E.; Lu, J.; Tan, Y.P.; Zhou, J. Deep video hashing. *IEEE Trans. Multimed.* **2017**, *19*, 1209–1219. [CrossRef]