electronics

Article

An Automatic Near-Duplicate Video Data Cleaning Method Based on a Consistent Feature Hash Ring

Yi Qin *, Ou Ye * and Yan Fu

College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an 710054, China; fuy@xust.edu.cn
* Correspondence: qiny@xust.edu.cn (Y.Q.); oye0928@xust.edu.cn (O.Y.)

Abstract: In recent decades, with the ever-growing scale of video data, near-duplicate videos continue to emerge. Data quality issues caused by near-duplicate videos are becoming more and more prominent, which has affected the application of normal videos. Although current studies on near-duplicate video detection can help uncover data quality issues for videos, they still lack a process of automatic merging for the video data represented by high-dimensional features, which makes it difficult to automatically clean the near-duplicate videos to improve data quality for video datasets. At present, there are few studies on near-duplicate video data cleaning. The existing studies have the sensitive problems of video data orderliness and initial clustering centers under a condition that prior distribution is unknown, which seriously affects the accuracy of near-duplicate video data cleaning. To address the above issues, an automatic near-duplicate video data cleaning method based on a consistent feature hash ring is proposed in this paper. First, a residual network with convolutional block attention modules, a long short-term memory deep network, and an attention model are integrated to construct an RCLA deep network with the multi-head attention mechanism to extract spatiotemporal features of video data. Then, a consistent feature hash ring is constructed, which can effectively alleviate the sensitivity of video data orderliness while providing a condition of near-duplicate video merging. To reduce the sensitivity of the initial cluster centers to the results of near-duplicate video cleansing, an optimized feature distance-means clustering algorithm is constructed by utilizing a mountain peak function on a consistent feature hash ring, which can implement automatic cleaning of near-duplicate video data. Finally, experiments are conducted based on a commonly used dataset named CC_WEB_VIDEO and a coal mining video dataset. Compared with some existing studies, simulation results demonstrate the performance of the proposed method.

Keywords: video cleaning; deep learning; consistent feature hash ring; feature distance means; mountain peak function; multi-head attention mechanism; near-duplicate videos

check for updates

Citation: Qin, Y.; Ye, O.; Fu, Y. An Automatic Near-Duplicate Video Data Cleaning Method Based on a Consistent Feature Hash Ring. *Electronics* **2024**, *13*, 1522. https://doi.org/10.3390/electronics13081522

Received: 26 March 2024
Revised: 12 April 2024
Accepted: 15 April 2024
Published: 17 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, with the innovative progress of video editing, 5G communication, and other related technologies, the popularity of video-related applications and services has led to the continuous expansion of the scale of video data, which shows a continuous exponential growth trend [1]. Take short videos as an example—the data from iiMedia Research show that the scale of short video users in China has an obvious growth momentum, which exceeded 700 million in 2020. These users deliver vivid and diverse information resources through video creation, video sharing, and video recommendations to enrich daily lives by using short video platforms.

In fact, as the scale of video data increases, many similar videos continue to emerge after video editing, with a new version of the modified original reissued and other operations for the videos, which are also referred to as near-duplicate videos (NDVs) [2]. In [3], near-duplicate videos are defined as identical or approximately identical videos that are close to each other and hard to distinguish but are different in some detail. In general, near-duplicate videos are derived from the original video, which not only illegally infringes upon the copyright of the video producer [4] but also affects the data quality of video datasets. For example, Liu et al. [5] use 24 keywords to search for videos on common websites of YouTube, Yahoo! Video, and Google Videos, and the results show that there are a lot of near-duplicate videos on the aforementioned websites. In individual cases, the redundancy can reach 93%. These near-duplicate videos will not only cause copyright issues and affect normal applications of video surveillance, video recommendation, etc. but also significantly reduce the data quality of video datasets, making the maintenance and management of video data more and more challenging. At present, some salient problems caused by near-duplicate videos have obtained increasing attention in academia and industry.

From the perspective of video data quality [6], a great deal of attention is paid to the overall quality of the video dataset, and stress on the degree of data consistency, data correctness, data completeness, and data minimization is satisfied in the information systems. The emergence of near-duplicate videos will reduce the degree of data consistency and minimization for video datasets. These near-duplicate videos can be taken to be a kind of dirty data, which have wide coverage and rich and diverse forms. Concretely, regardless of the stage of video collection, video integration, or video processing, it is possible to generate near-duplicate videos. For instance, in the video collection stage, they can be collected from different angles within the same scene; in the video integration stage, there may be near-duplicate videos with different formats and video lengths from different data sources; in the video processing stage, video copy, video editing, and other operations will produce mass near-duplicate videos. Studies on near-duplicate video detection can help us discover hidden near-duplicate videos in video datasets. Currently, various kinds of methodologies have been proposed in the literature, and the implementation process mainly includes feature extraction, feature signature, and signature index. In either of these methodologies, feature extraction can be regarded as a key component of near-duplicate video detection. From the perspective of video feature representation, near-duplicate video detection methodologies can be categorized into hand-crafted feature-based methodology and high-level feature-based methodology [7–9]. Nevertheless, near-duplicate video detection methodologies can only identify the near-duplicate videos in a video dataset [10,11] that lacks a process of feature sorting and automatic merging for the video data represented by high-dimensional features. Therefore, it is very challenging for them to automatically clean up redundant near-duplicate videos to reduce video copyright infringement and related issues caused by video copying, video editing, and other manual operations.

At present, data cleaning modeling techniques are important technical ways to effectively reduce near-duplicate data and improve data quality. By using this kind of modeling technique, near-duplicate data existing in the datasets can be automatically cleaned, so the datasets meet data consistency, completeness, correctness, and minimization, and achieve high data quality. At present, data cleaning modeling techniques have been studied more deeply in big data cleaning [12], stream data cleaning [13], contextual data cleaning [14], etc., which can effectively address the data quality issues at the instance layer and schema layer. However, there is still a significant gap in research on near-duplicate video cleaning, and there is a sensitive problem of video data orderliness and the initial clustering center sensitive problem under a condition that prior distribution is unknown in the existing studies, which seriously affects the accuracy of near-duplicate video cleaning.

In this paper, an automatic near-duplicate video cleaning method based on a consistent feature hash ring (denoted as RCLA-HAOPFDMC) is proposed to address the above-mentioned issues, which consists of three parts: high-dimensional feature extraction of video data, consistent feature hash ring construction, and cluster cleaning modeling based on a consistent feature hash ring. First, a residual network with convolutional block attention modules, a long short-term memory (LSTM) deep network model, and an attention model are integrated to extract temporal and spatial features from videos by constructing a multi-head attention mechanism. Then, a consistent feature hash ring is constructed, which can effectively alleviate the sensitivity of video data orderliness while providing a condition of near-duplicate video merging. Finally, to reduce the sensitivity of the initial cluster centers to the results of near-duplicate video cleansing, an optimized feature distance-means clustering algorithm is constructed by utilizing a mountain peak function on a consistent feature hash ring to implement automatic cleaning of near-duplicate videos. A commonly used dataset named CC_WEB_VIDEO [5] and a coal mining video dataset [15] are used to confirm the practical effect of our proposed method. The contributions are summarized as follows: (1) A novel consistent feature hash ring is constructed, which can alleviate the sensitivity issue of video data orderliness while providing a condition of near-duplicate video merging. (2) An optimized feature distance-means clustering algorithm is constructed by utilizing a mountain peak function on a consistent feature hash ring to merge and clean up near-duplicate videos. (3) The method presented in this paper is successful on the highly difficult CC_WEB_VIDEO and coal mining video dataset where the coal mining video dataset has complex context scenes.

The following is the organizational structure of the remaining parts of this article. In Section 2, a brief review of related works on near-duplicate video detection and data cleaning is presented. An automatic near-duplicate video cleaning method based on a consistent feature hash ring is proposed in Section 3. The experimental results that validate the performance of the method are presented in Section 4. Finally, the paper is summarized in Section 5.

2. Related Work

In this section, the previous near-duplicate video detection methodologies and image/video cleaning methodologies are briefly reviewed. First, some hand-crafted feature-based methodologies and high-dimensional feature-based methodologies for near-duplicate video detection are examined; then, some data cleaning methodologies for image cleaning and video cleaning are reviewed. Finally, the shortcomings of the above-mentioned methodologies are analyzed.

2.1. Near-Duplicate Video Detection Methodologies

In the past decade, hand-crafted features, such as SIFT, HOG, and MSER, have been widely used for near-duplicate video detection. For example, the study in [16] adopts the SIFT local feature to encode temporal information, generates temporal set SIFT features by tracking SIFT, and combines local sensitive hash algorithms to detect near-duplicate videos. Although SIFT features of a video frame can maintain invariance to rotation, scaling, and illumination changes, as well as maintain a certain degree of affine transformation and noise, some of the invariance of SIFT will be damaged to a certain extent during strong camcording. Henderson et al. [17] adopt key point features from a Harris corner, SURF, BRISK, FAST, and MSER descriptors to detect video frame duplication. This method incorporates different local features to represent video frames but ignores the global features and spatiotemporal features that video frames have. Zhang et al. [18] integrate Harris 3D spatiotemporal feature and HOG/HOF global feature descriptors to detect near-duplicate news web videos and apply the Jaccard coefficient to similarity metric; however, there exists the issue of inefficient detection in this method. In [19], a new near-repeat video detection system, Compound Eyes, is proposed, which combines seven hand-made features (such as color consistency, color distribution, edge direction, motion direction, etc.) to improve the efficiency of near-repeat video detection. However, this method is susceptible to feature changes. The work in [3] adopts an unsupervised cluster algorithm based on temporal and spatial key points to automatically identify and classify near-duplicate videos, but the results of near-duplicate video detection are sensitive to initializing the cluster center.

In general, the combination of spatial and temporal features can more comprehensively and accurately represent the spatiotemporal information contained in video data than the representation of a single low-level feature; hence, the methodologies based on spatiotemporal features can identify near-duplicate videos more accurately. However, the methodologies based on low-level features need to have prior knowledge, and the results of near-duplicate video detection are easily affected by disturbances from illuminations, occlusions, distortion, etc.

Recently, various deep network models have been utilized to detect near-duplicate videos, and these models have much better representational capacity than the methodologies based on hand-crafted features. For instance, the study in [7] presents a survey on the utilization of deep learning techniques and frameworks. Nie et al. [20] use a pre-trained convolutional neural network model to extract high-dimensional features of videos, and a simple but efficient multi-bit hash function is proposed to detect near-duplicate videos. This is a supervised joint view hashing method, which can improve the performance of accuracy and efficiency. However, the distribution of near-duplicate video data in the video dataset is usually unknown in practical applications, so the application of the supervised joint view hashing method is limited. In [21], the near-duplicate video is detected by combining the two-stream network model of RGB and optical flow, multi-head attention, and the Siamese network model. The limitation of this method is that it adopts a cosine distance function to measure the similarity between every two videos, which results in relatively low efficiency. Moreover, a neighborhood attention mechanism is integrated into an RNN-based reconstruction scheme to capture the spatial-temporal features of videos, which is used to detect near-duplicate videos [22]. The work in [23] uses a temporal segment network model to detect near-duplicate video data. These above-mentioned models based on temporal networks can detect near-duplicate videos by capturing the temporal features of videos. In [24,25], a Parallel 3D ConvNet model and a spatiotemporal relationship neural network model are adopted to extract spatiotemporal features to detect near-duplicate videos.

In summary, high-dimensional feature-based methodologies can achieve better performance than hand-crafted feature-based methodologies: they can reduce the impacts of disturbances from illuminations, occlusions, and distortion on the model results. Near-duplicate videos can be identified directly by either hash mapping or similarity metric, but the above-mentioned methodologies lack a process to automatically merge data with high-dimensional features, which makes it more difficult to clean up near-duplicate videos automatically.

2.2. Data Cleaning Methodologies

Data duplication may have the following reasons: data maintenance, manual input, device errors, and so on [26]. Data cleaning modeling techniques are effective ways to automatically clean and reduce near-duplicate data, which can effectively address the shortcomings of near-duplicate video detection methodologies. Recently, the amount of literature on the topic of data cleaning [27] has shown a rapid growth trend, and most of the existing works are on concentrated stream data cleaning and spatiotemporal data cleaning.

In the area of stream data cleaning, for instance, reference [28] proposes a stream data cleaning method named SCREEN, which can clean up the steam data by finding the repair sequence with the smallest difference from input, construct an online cleaning model, and calculate the local optimal of the data point. However, this method does not guarantee that near-duplicate data are exactly adjacent to each other in the same sliding window. The work in [29] proposes a streaming big data system, which is based on an efficient, compact, and distributed data structure to maintain the necessary state for repairing data. Additionally, it improves cleaning accuracy by supporting rule dynamics and utilizing sliding window operations. The limitation of this method is that the fixed size of the sliding window has a significant impact on the cleaning results. In [30], a sliding window and K-means cluster algorithm are adopted to clean stream data, but the result of this method is sensitive to the initialization of cluster centers.

To sum up, although methodologies of stream data cleaning can clean up the stream data effectively, there are limitations in that the results of models are sensitive to the fixed-size sliding window and the pre-defined initialized clustering center.

In time series data cleaning, for example, Ranjan et al. [31] unitize a k-nearest neighbor algorithm and a sliding window prediction approach to clean time series data on a set of nonvolatile and volatile time series. This method can optimize the width of the sliding window to enhance the performance, but to optimize parameters that affect performance, a general scheme needs to be developed. In [32], a top-N keyword query processing method is proposed, which is based on real-time entity parsing to clear datasets with duplicate tuples. The limitation of this method is that the selection of keywords has a salient impact on the results. The study in [33] proposes an approach of real-time data cleaning and standardization, which clarifies the workflows of data cleaning and data reliability, and it can be adapted to clean up near-duplicate time series data. However, this approach is not enough to describe the details of real-time data cleaning.

Through the studies of time series data cleaning, it is found that there is a prevalence of erroneous data in the industrial field; hence, the studies mainly focus on cleaning the erroneous data and less on cleaning near-duplicate data. In the existing studies, k-nearest neighbor and top-K algorithms are widely adopted to clean near-duplicate time series data since they do not rely on prior knowledge of the distribution of time series data. Nevertheless, the presetting of the k parameters has a significant impact on the cleaning results.

In spatiotemporal data cleaning, taking reference [34] as an example, a probabilistic system named Current Clean is presented, which uses a spatiotemporal probabilistic model and a set of inferences to identify and clean stale data in the database. This probabilistic system is applied to data cleaning with spatiotemporal features in relational databases, but it is difficult to apply to unstructured data cleaning with high-dimensional spatiotemporal features. To address this issue, the study in [15] proposes a method to clean up near-duplicate videos by using locality-sensitive hashing and a sorted neighborhood algorithm. However, in this method, it is challenging to use SIFT and SURF hand-crafted features accurately to portray video features, and the use of a sorted neighborhood algorithm causes a more prominent orderliness-sensitive problem of video data. The authors of [35] achieve an improvement in the quality of the image dataset by automatically clearing minority images using a convolutional neural network model. However, the completeness of image datasets may be destroyed when cleaning images of the minority classes. Fu et al. [36] propose a near-duplicate video cleaning method based on the VGG-16 deep network model and feature distance-means clustering fusion to improve the data quality of video datasets, which takes less account of the temporal feature representation of videos and suffers from the initial cluster center sensitive problem under a condition that prior distribution is unknown. Moreover, a novel content based on the video segmentation identification scheme is proposed to reduce the mass of near-duplicate video clips [37]. H. Chen et al. [38] utilize the similarity measurement to clean the duplicate annotations of video data in the MSR-VTT dataset.

In summary, there are few studies on data cleaning for unstructured data with spatiotemporal features, such as video and audio data, due to less consideration from the perspective of data quality. Recently, the studies on near-duplicate video cleaning mainly have the following issues: (1) It is more difficult to be able to arrange all near-duplicate videos near sorting algorithms, so the accuracy of cleaning is more sensitive to the data orderliness. (2) Utilizing the idea of clustering, video data with the most significant features can be retained in all near-duplicate data and the rest deleted, but the setting of initial clustering centers is more sensitive to cleaning results under a condition that prior distribution is unknown.

To address these two issues, we consider constructing a novel consistent feature hash ring based on optimizing video feature extraction to map video data to low-dimensional space, which is used to reduce data orderliness sensitivity issues caused by data sorting and provide a condition of near-duplicate video merging. On this basis, an optimized feature distance-means clustering algorithm is constructed, which merges a mountain peak function on a consistent feature hash ring to overcome the initial clustering center sensitive problem under the condition that the prior distribution is unknown.

3. The Proposed Method

This section describes how to utilize the proposed novel automatic near-duplicate video cleaning method based on a consistent feature hash ring, which can automatically clean up near-duplicate videos to improve the data quality of video datasets. It is important to note here that the concepts of the data quality of video datasets and video quality are different. Normally, video quality is concerned with the clarity of videos and involves performance metrics such as resolution and frames per second. The data quality of the video dataset is concerned with the degree to which the video dataset satisfies data consistency, data completeness, data correctness, and data minimization. The goal is to remove redundant near-duplicate videos from video datasets by the method proposed in this paper so that the video datasets consistently maintain high data quality.

To achieve this goal, three main stages need to be accomplished: feature representations of video data, identification of near-duplicate videos, and deletion of near-duplicate videos. It should be noted that if all identified near-duplicate videos are removed, the data completeness and data correctness of video datasets will be affected. If only some of the near-duplicate videos are removed, the data consistency and data minimization of video datasets will be affected. Therefore, it is a difficult challenge to retain video data with the most salient features to ensure the data completeness and data correctness of video datasets while removing the rest near-duplicate video data to ensure data consistency and data minimization of video datasets.

At present, considering the time cost of near-duplicate video cleaning, the key insight of existing studies is to overcome the above challenges by exploiting data ordering and clustering to retain video data with the most salient features and remove the rest of near-duplicate videos. However, the sensitivities of cleaning effect on data orderliness and the initial cluster center setting are major issues. Inspired by the distributed big data storage processing, a consistent feature hash ring is constructed in this paper. The advantage of utilizing a consistent feature hash ring is to reduce the impact of data sorting on the cleaning results by mapping video data with high-dimensional features to a feature hash ring while providing a condition for removing near-duplicate videos. Figure 1 outlines our approach, which consists of three parts: high-dimensional feature extraction of videos, construction of a consistent feature hash ring, and cleaning near-duplicate videos based on a consistent feature hash ring. Each of these sections is explained next.

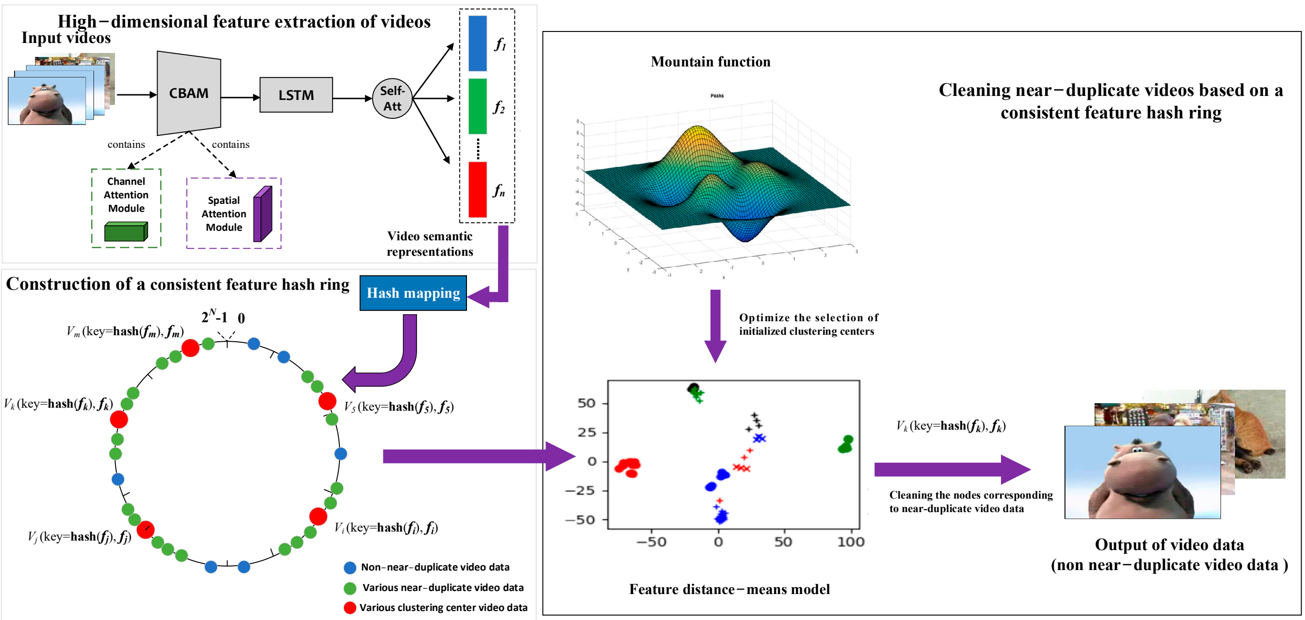


Figure 1. The overall framework of our proposed method.

In general, the proposed method is based on high-dimensional feature extraction from video data, drawing on the ideas of load balancing and high scalability in big data storage to construct a novel consistent feature hash ring. On this basis, optimizing the FD-Means clustering algorithm to automatic cleaning of near-duplicate video data is achieved.

3.1. High-Dimensional Feature Extraction of Videos

The feature representation of video data is an important stage in the data cleaning process of videos. Currently, several convolutional neural network models have been adopted in the study of image and video data cleaning for feature representation of image or video data, such as AlexNet [35], GoogleNet [35], VGG-16 [36], and ResNet50 [39]. Due to the spatiotemporal features of video data, it is not enough to rely on the above-mentioned convolutional neural network models for spatial feature representations of videos, and the extracted video features are less likely to highlight the spatiotemporal features of salient regions in video data, which will affect the accurate representation of video semantics. To overcome such a limitation, a residual network with convolutional block attention modules is adopted first to extract spatial features of video data. The channel attention and spatial attention modules in these convolutional block attention modules can effectively improve the representation capability of spatial features in the saliency region of video data. Then, the above network and a long short-time memory model are integrated to extract the spatiotemporal features of video data. Finally, to highlight the role of key information in video data on the video semantic representations, an attention model based on the above network models Is Introduced along three independent dimensions of channel, space, and time series to construct a video spatiotemporal feature extraction model with multi-head attention mechanism, which is named RCLA (Resnet-CBAM-LSTM-multi-head attention) model in this paper. The concrete architecture of the RCLA model is shown in Figure 2.

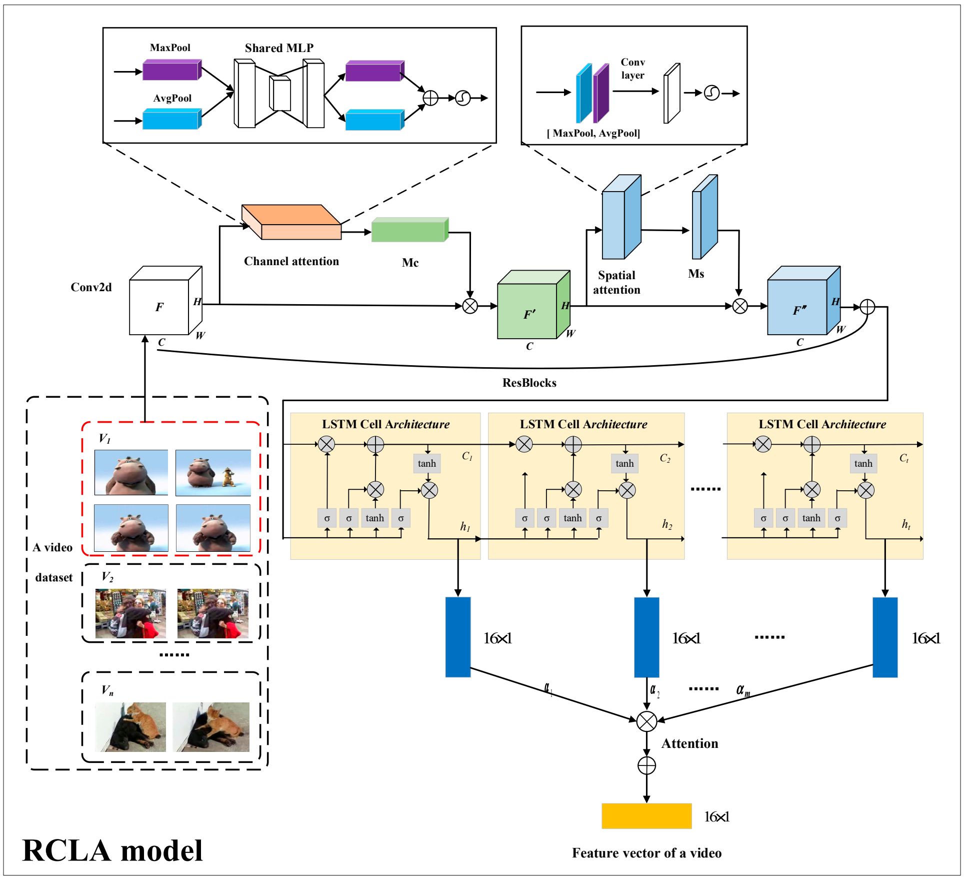


Figure 2. The architecture of the RCLA model.

Considering the scale of data used for training and testing in this paper, several video data training samples are first selected to input a residual network with convolutional block attention modules. Let the size of the above video data be $w \times h \times c \times l$ where $w \times h$ represents the size of a video frame, c represents the number of channels per frame, and l represents the number of frames of the video data [40]. Before training the 34-layer residual network, set the values of w and h to 224, and the value of c to 3. In addition, first, fix the 7×7 convolutional kernel with stride 2 in the convolution layer; then, fix the pooling window with stride 2 in the pooling layer 3×3 to implement the convolutional operation and max-pooling process. During the above process, use a BatchNorm2d method to normalize the input data and use a rectified linear units (ReLU) activation function to alleviate the problem of gradient dispersion. Thus, a feature map F of size $56 \times 56 \times 3$ for a video frame can be obtained. Then, input F into the middle part of the residual network with 34 layers. This part is composed of 4 blocks, which respectively include 3, 4, 6, and 3 residual blocks, and each residual block contains a convolutional block attention module (CBAM) [41]. In this module, the details of spatial features in a video frame can be profiled by constructing a channel attention map Mc and a spatial attention map Ms. Specifically, the spatial information of feature map F is aggregated by first performing maximum pooling and average pooling operations on F, respectively, and this spatial information is input into a multilayer perceptron (MLP) network model with 2 layers that share the wights $\mathbf{W}_0 \in \mathbb{R}^{c/r \times c}$ and $\mathbf{W}_1 \in \mathbb{R}^{c \times c/r}$ (r is the reduction ratio, r is set to 16 in this paper), respectively, so that the channel attention map Mc can be obtained by Equation (1) and an intermediate feature map $\mathbf{F}^{'}$ is computed by Equation (2):

$$\mathbf{M_C}(\mathbf{F}) = \sigma\Big(\mathbf{W}_1(\mathbf{W}_0(\mathbf{F^c_{avg}})) + \mathbf{W}_1(\mathbf{W}_0(\mathbf{F^c_{max}}))\Big) \tag{1}$$

$$\mathbf{F}^{'} = M_C(\mathbf{F}) \otimes \mathbf{F} \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function; the meanings of $\mathbf{F}^c_{avg}$ and $\mathbf{F}^c_{max}$ are average-pooled features and max-pooled features, respectively; and "$\otimes$" is used for element-by-element multiplication.

Then, the average pooling and maximum pooling operations are performed on $\mathbf{F}^{'}$ to generate features $\mathbf{F^s_{avg}} \in \mathbb{R}^{1 \times h \times w}$ and $\mathbf{F^s_{max}} \in \mathbb{R}^{1 \times h \times w}$, respectively. On this basis, the spatial attention map $\mathbf{M_s}$ is calculated by Equation (3), and the final refined feature map $\mathbf{F}^{''}$ can be obtained by Equation (4) as follows:

$$\mathbf{M_S}\left(\mathbf{F}^{'}\right) = \sigma\left(conv\left([\mathbf{F^s_{avg}} \otimes \mathbf{F^s_{max}}]\right)\right) \tag{3}$$

$$\mathbf{F}^{''} = \mathbf{M_S}\left(\mathbf{F}^{'}\right) \otimes \mathbf{F}^{'} \tag{4}$$

where $conv(\cdot)$ denotes a convolution operation.

Through the above-mentioned different residual blocks with convolutional block attention modules, the feature vectors $\boldsymbol{f_{rc}}$ of size [512, 1] can be obtained to represent the spatial features exhibited by video frames.

Then, the spatial features $\boldsymbol{f_{rc}}$ are input into the long short-term memory network (LSTM) to further extract the temporal features of video data. Considering the size of datasets used in this paper, a one-layer LSTM with N (we set N = 16 in this paper) hidden layer nodes is employed, which consists of an input gate, a forget gate, and and output gate. The hidden state $\mathbf{h^{ls}_t}$ at t moment is calculated as shown in Equation (5):

$$\mathbf{h^{ls}_t} = \mathrm{LSTM}\left(\mathbf{f}_{\mathbf{r}c},\ \mathrm{W_{ls}}, \mathbf{h^{ls}_{t-1}}; \theta_{ls}\right) \tag{5}$$

where LSTM$(\cdot)$ denotes the formalized function of an LSTM network model; $\mathrm{W_{ls}}$ denotes a parameter matrix learned during training of the LSTM network model; $\mathbf{h^{ls}_{t-1}}$ denotes the hidden state at $t-1$ moment; and θ_{ls} denotes the hyperparameters of an LSTM network model. Through the output layer of an LSTM network model, the spatiotemporal features f_{st} of size $5 \times 16 \times 1$ can be obtained to represent video data.

To focus on the visual features of different video frames to highlight the semantic contents of video data, an attention module based on the above models is introduced, as shown in Equation (6):

$$\mathbf{f} = \mathrm{Att}\left(\mathbf{h}_{\mathbf{t}}^{\mathbf{ls}}, \mathbf{f}_{\mathbf{st}} ; \mathbf{W}_{\mathbf{Att}} \right) \tag{6}$$

where $\mathrm{Att}(\cdot)$ denotes a standard additive attention function; $\mathbf{W}_{\mathbf{Att}}$ denotes the weight vectors in an attention module; $\mathbf{f}$ denotes the semantic features of video data obtained by using the multi-head attention mechanism, and the size is 16×1.

When training the RCLA model, the goal is to minimize the learning loss of each deep neural network model. Considering the cascade relationship between each of the above neural network models, the output of the above attention module as the input of a loss function is used to perform optimization of an RCLA model, and the above-mentioned loss function is constructed as shown in Equations (7) and (8):

$$\mathbf{y}_{\mathbf{s}} = \mathrm{argmin}\left(\left\|\mathbf{y}_{\mathbf{i}} - \mathbf{y}^{\mathbf{j}}_{\mathbf{video\text{-}seed}}\right\|_2^2\right) \; s.t.\; i \in [1, N_v]\,,\, j \in [1,\ N_{v-C}] \tag{7}$$

$$L(y'_i, \mathbf{y}_{\mathbf{i}}) = -\sum_{i=1}^{16} y'_i \times \log(sim(\mathbf{y}_{\mathbf{i}}, \mathbf{y}_{\mathbf{s}})) \tag{8}$$

where $\mathbf{y}_{\mathbf{i}}$ denotes a feature vector of the ith video data in a video dataset with N_v video data, $\mathbf{y}^{\mathbf{j}}_{\mathbf{video\text{-}seed}}$ denotes a feature vector of the jth seed video in the above video dataset, N_{v-C} denotes the total number of preset seed videos, y'_i denotes the label corresponding to $\mathbf{y}_{\mathbf{i}}$, and $sim(\cdot)$ denotes a function of the similarity measurement.

3.2. The Construction of a Consistent Feature Hash Ring

To efficiently identify near-duplicate video data with high-dimensional features and reduce the impact of high-dimensional feature sorting on near-duplicate video data cleaning, a consistent feature hash ring is constructed, which is inspired by the use of distributed hash tables to address the problems of load balancing and high scalability in big data storage, as shown in Figure 3.

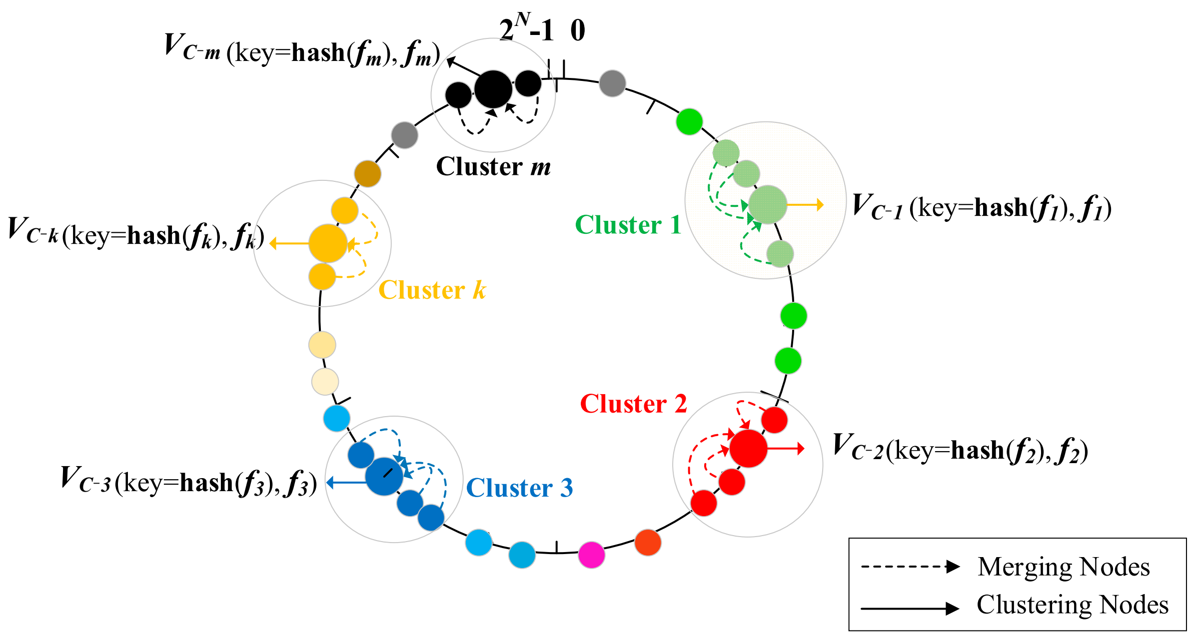


Figure 3. The structure of a consistent feature hash ring.

Assuming that a hash function can be used to map the high-dimensional features of all video data to a set of hash values, and the largest hash value is less than 2^N, then a consistent feature hash ring is a virtual ring constructed through this set of hash values, and the range of hash value space of this ring is [0, $2^N - 1$] (hash value is a 32-bit unsigned integer, and $N = 11$ is set in this paper). An entire spatial distribution of the consistent feature hash ring is organized in a clockwise direction. The hash value at the top of this ring as the starting point is 0, and the hash value at the first point on the left of 0 is $2^N - 1$, and the hash value at the first point on the right of 0 is 1. By analogy, all the hash values are distributed from small to large clockwise until they return to the starting point.

When constructing a consistent feature hash ring, a hash function is designed as shown in Equation (9), which can be used to calculate the hash values corresponding to each video and map all video data to the consistent feature hash ring.

$$hash(\mathbf{f_i}) = [binary(\mathbf{f}_i) + 1] \bmod (2^N - 1) \tag{9}$$

where $binary(\cdot)$ denotes a binary encoding function and $\mathbf{f_i}$ denotes a high-dimensional feature vector of the *i*-th video data.

By utilizing the binary function as shown in Equation (10), the high-dimensional features of all video data can be mapped to compact binary codes, which can not only decrease the dimensions of video features but also perform metric operations in the low-dimensional space.

$$binary(\mathbf{f_i}) = \mathrm{MD5}(\|\tanh(\mathbf{f_i})\|_1) \tag{10}$$

where $\mathrm{MD5}(\cdot)$ denotes an encryption function; and $\tanh(\cdot)$ is an activation function.

Specifically, a $\tanh(\cdot)$ activation function is utilized to perform a nonlinear variation of the video data with a high-dimensional feature. On this basis, to improve the sparsity of the hash distribution on the consistent feature hash ring, a $\updownarrow_1$ norm is used to calculate a value to represent video data. Subsequently, this value is encrypted with the MD5 encryption algorithm and converted into a fixed-length binary code as a hash value of the video data (the length of a binary code is set to 16 in this paper). Finally, a linear detection method is used when storing multiple hash values to avoid the same hash addresses being preempted by different hash values, i.e., an address is assigned by adding 1 to the back, and the modulus of the maximum value on a consistent feature hash ring is taken as the upper bound of the address range until there is a free address. Here, the modulo operation is to ensure that the location found is in the effective space of $2^N - 1$. Thus, the *i*-th video data can be mapped as video hash feature points xi on a consistent feature hash ring in the form of two-dimensional coordinates $(hash(\mathbf{f}_i), \mathbf{f}_i)$.

3.3. FD-Means Clustering Cleaning Optimization Algorithm with Fused Mountain Peak Function

The efficiency and accuracy of the partitioning clustering algorithm are closely related to the selection of the initial clustering center. The FD-Means clustering algorithm [36] randomly selects several initial cluster centers for multiple clustering and finally selects the optimal clustering centers as the initial clustering centers. However, it requires a large amount of calculation, and poor effect leads to the volatility of clustering results. To address this issue, a mountain peak function is fused with the FD-Means clustering algorithm to optimize the selection of the initial clustering centers to automatically clean the near duplicate video data accurately.

Specifically, assuming that a set of video features $\mathrm{S}_f = \left\{\|\mathbf{f_1}\|_2, \|\mathbf{f_2}\|_2, \cdots, \|\mathbf{f_i}\|_2, \cdots, \|\mathbf{f_{N_v}}\|_2\right\}$ where $\mathbf{f_i}$ denotes a vertical ordinate of x_i and Nv denotes the total number of video data. To select several initial clustering centers of video data, all the data samples on the consistent feature hash ring are first divided into a finite grid, and all the cross points of a $K \times K$ grid can be used as the candidate centroids of the clustering centers, as shown in Equations (11) and (12):

$$T_{interval} = \frac{\max(S_f) - \min(S_f)}{N_v} \tag{11}$$

$$P_C^{i_c} = \min(S_f) + i_c \times T_{interval} \tag{12}$$

where $T_{interval}$ denotes the partition interval of a grid; i_c denotes an index of the *i*-th cluster center $V_C^{i_c}$, and $i_c \in [1, K]$; $P_C^{i_c}$ denotes the value of the *i*-th cluster center $V_C^{i_c}$ in a grid. Moreover, K can be determined by inequality $(K-1)^2 < T_{interval} \leq K^2$.

Subsequently, the points with higher density in grid space are found by constructing a mountain peak function for each cross point to calculate its peak value H_{V_C}. For example, a peak value of the *i*-th cluster center $V_C^{i_c}$ is calculated as shown in Equation (13):

$$H_{V_C}^i = \sum_{i=1}^{N_v} \exp\left(-\frac{\left(P_C^{i_c} - \|\mathbf{f}_i\|_2\right)^2}{2\sigma^2}\right) \tag{13}$$

where σ is a constant value.

On this basis, the video hash feature points corresponding to the cross points of K maximum peak values are selected sequentially as the initialized clustering centers on a consistent feature hash ring, denoted as $V_C = \{V_{C-1}, V_{C-2}, \cdots, V_{C-K}\}$.

Since the FD-Means clustering algorithm seldom considers the curse of dimensionality when using Euclidean distance to measure the similarities between video features, a new similarity measurement function is constructed, as shown in Equation (14):

$$\mathrm{Dist}_{\mathrm{VFP}}(\mathbf{f_i}, \mathbf{f_j}) = \alpha \times |FD(\mathbf{f_i}, \mathbf{f_j})| + |hash(\mathbf{f_i}) - hash(\mathbf{f_j})| \tag{14}$$

where α denotes a weight factor; and $FD(\mathbf{f_i}, \mathbf{f_j})$ denotes the Euclidean distance between any *i*-th and *j*-th video hash feature points, as shown in Equation (15):

$$FD(\mathbf{f_i}, \mathbf{f_j}) = \begin{pmatrix} \|\mathbf{f}_1 - \mathbf{f}_1\|_2 \|\mathbf{f}_1 - \mathbf{f_2}\|_2 & \cdots & \|\mathbf{f}_1 - \mathbf{f}_m\|_2 \\ \|\mathbf{f}_2 - \mathbf{f}_1\|_2 \|\mathbf{f}_2 - \mathbf{f_2}\|_2 & \cdots & \|\mathbf{f}_2 - \mathbf{f}_m\|_2 \\ \vdots \quad \vdots & \ddots & \vdots \\ \|\mathbf{f}_q - \mathbf{f}_1\|_2 \|\mathbf{f}_q - \mathbf{f_2}\|_2 & \cdots & \|\mathbf{f}_q - \mathbf{f}_m\|_2 \end{pmatrix} \tag{15}$$

where q and m denote the dimensions of two feature vectors.

When updating an FD-Means cluster center, the sum of distances between a video hash feature point and other video hash feature points in a cluster is first calculated, and the video hash feature point with the smallest sum of distances is selected as a cluster center through iteration, as shown in Equation (16):

$$\begin{aligned} \mathrm{Dist}(x_i, d) &= \sum_{j \in d - x_i} \mathrm{Dist}_{\mathrm{VFP}}(\mathbf{f_i}, \mathbf{f_j}) \text{ , s.t. } x_i \in d \\ V_{C-i}^* &= \ \mathrm{argminDist}(x_i, d) \end{aligned} \tag{16}$$

where d denotes a cluster; and $V_C^{i_c{}^*}$ denotes a new cluster center after the initial cluster center of d is updated.

When updating a cluster, the distance between the video hash feature points of the non-cluster centers and all the cluster centers is first calculated. In the K-Means clustering algorithm, all points of non-cluster are divided into the nearest clusters according to the nearest neighbor principle. Unlike the K-Means clustering algorithm, the optimized FD-Means clustering algorithm compares the distances between the video hash feature points of non-cluster centers and all cluster centers. If the minimum distance is not less than the given threshold δ, the video hash feature points are used as the cluster centers of new clusters; otherwise, according to the nearest neighbor principle, they are grouped into the nearest clusters, and the automatic clustering of video hash feature points on a consistent hash ring is finally achieved, as shown in Equations (17) and (18):

$$\min \mathrm{Dist}_C(x_i, V_C;\ \delta) = \mathrm{Dist}(x_i, V_C) - \delta \tag{17}$$

$$d^* = \begin{cases} d \cup x_i, \text{ s.t. min Dist}_C(x_i, V_C; \delta) < 0 \\ x_i, \text{ otherwise} \end{cases} \tag{18}$$

where d^* denotes an updated cluster.

In the automatic cleaning of near-duplicate video data, all the near-duplicate videos obtained by clustering cannot be eliminated to ensure the data consistency, completeness, correctness, and minimization of a video set. Therefore, a representative video can be retained from the near-duplicate videos, and other near-duplicate video data can be automatically deleted to improve the video data quality.

At present, the traditional image or video data de-duplication methods tend to retain the first detected data and remove other near-duplicate data in an ordered sequence, but such methods are randomized at different settings of sequence length, which will lead to fluctuations in the results of data cleaning. Hence, a video in a cluster cannot be arbitrarily selected as a seed video or representative video to be reserved. In this paper, the video data corresponding to the cluster centers are the representative video data to be retained, and others in the clusters are deleted, as shown in Equation (19):

$$D* = F(V_C) \tag{19}$$

where $F(\cdot)$ denotes a mapping function between a set V_C of cluster centers and a set V of the corresponding video data and the mapping relationship is expressed as $F : V_C \rightarrow V$, $D*$ denotes that the original video dataset is updated after the mapping function $F(\cdot)$.

Through the above processes, the automatic cleaning of near-duplicate video data is finally achieved.

4. Experimental Evaluation

The extensive experiments on a commonly used dataset named CC_WEB_VIDEO and a coal mining video dataset are conducted to evaluate the performance of RCLA-HAOPFDMC (the proposed method in this paper) and compare them with other representative advanced methods, such as the CBAM-Resnet [42] and BS-VGG16 [36]. All experiments were conducted on the same machine, which had 8-Inter Xeon processors with 2.10 GHz and a graphics card NVIDIA Corporation GP102 with 16 G memory; the programs were implemented based on Python version 3.6.5 and PyTorch version 0.4.0. Next, we provide a detailed explanation of the experiment and results.

4.1. Dataset and Evaluation Criteria

In this paper, the CC_WEB_VIDEO and coal mining video datasets are used to carry out the comparative experiment of the proposed method. The CC_WEB_VIDEO dataset contains 24 scenes and a total of 13,129 video data. This paper randomly selects 63 videos from scenes ("The Lion Sleeps Tonight", "Evolution of Dance", "Folding Shirt", "Cat Massage", and "ok go-here it goes again" scenes) to verify the effectiveness of the proposed method. The coal mining video dataset includes 125 video data with 10 scenes, which are all used to test the performance of the method presented in this paper.

We use common metrics such as accuracy, precision, recall, and F1-score to evaluate video data cleaning in order to evaluate the performance of the method proposed in this paper. The expression is as follows:

$$precision = \frac{TP}{TP + FP} \tag{20}$$

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

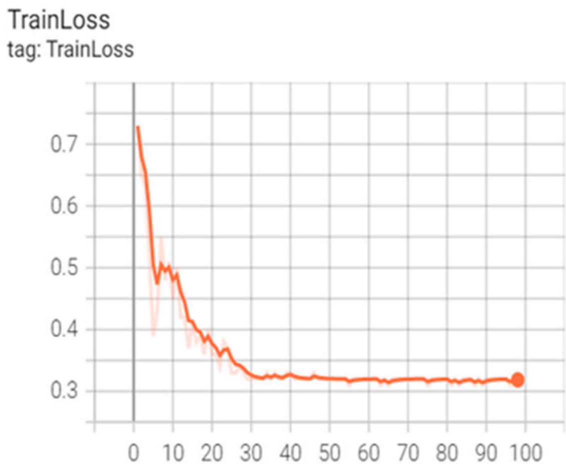
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{23}$$

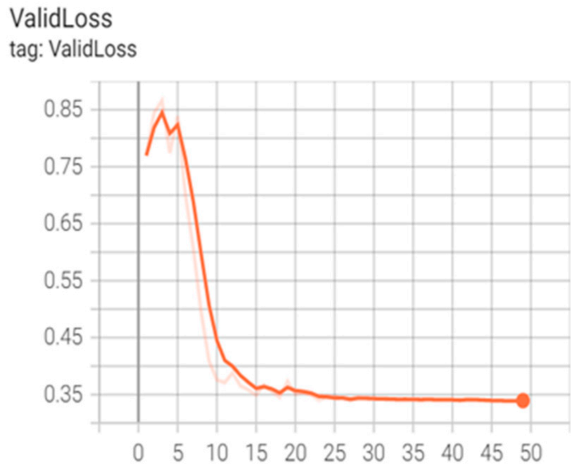
where *TP* is the number of true positive samples; *FN* is the number of false negative samples; *FP* is the number of false positive samples; and *TN* is the number of true negative samples.

4.2. Experimental Results and Analysis

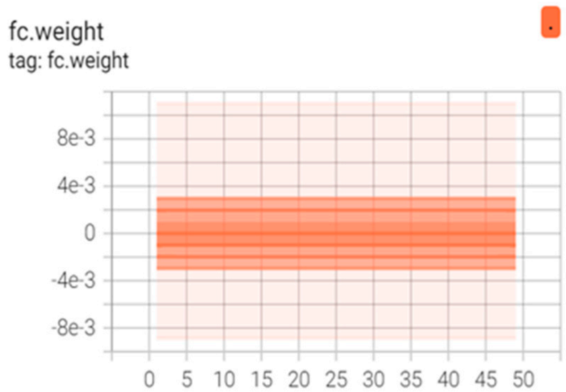
For the CC_WEB_VIDEO and coal mining video datasets, when training the RCLA model, the initialization of weights and bias variables is randomly generated. Moreover, the overfitting problem is solved by the dropout function and parameter-sharing methods, and the loss function is optimized by the Adam algorithm. Figures 4 and 5 show the weight variable changes and training losses in CC_WEB_VIDEO and coal mine video datasets. It can be seen from Figures 4 and 5 that the designed loss function in this paper is converged and that the different weights in the full connection layer of the RCLA model change in the range of −0.003 to 0.003. Moreover, Figure 4 shows that the range of different deviation values is −0.2 to 0.2 and Figure 5 shows the range of −0.24 to 0.2. The above experimental results show that there are no overfitting issues during the training and validation process of the experiment.



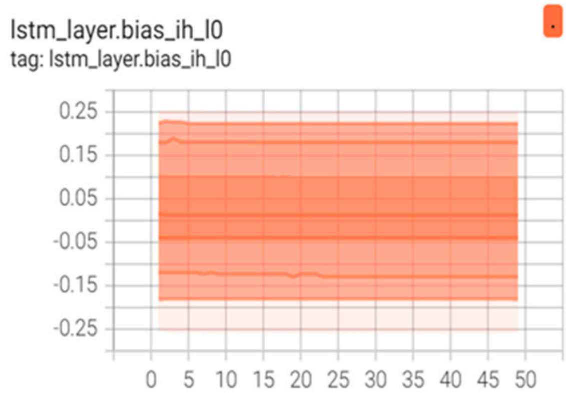
(**a**) Loss variation in training



(**b**) Loss variation in valid

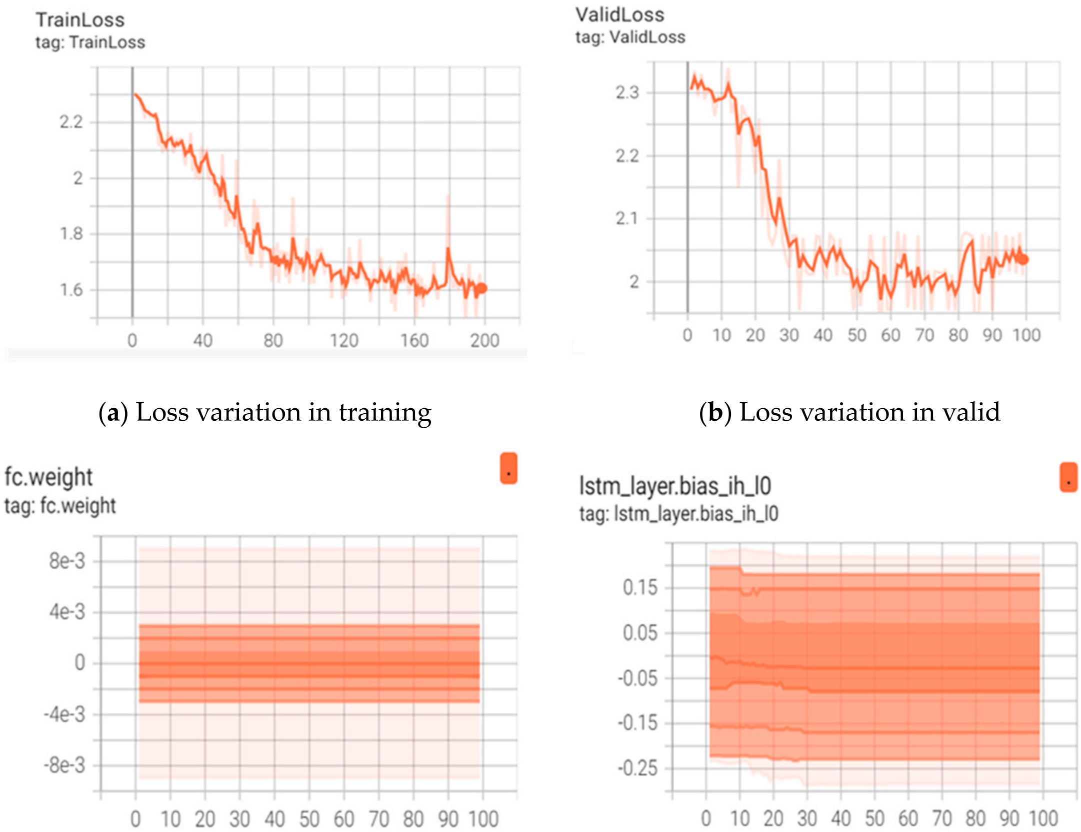


(**c**) weights vanati on in training



(**d**) biases variation in training

Figure 4. The visualization of loss and weight variation during the CC_WEB_VIDEO dataset.



(**a**) Loss variation in training (**b**) Loss variation in valid

(**c**) weights variation on in training (**d**) biases variation in training

Figure 5. The visualization of loss and weight variation during the coal mining video dataset.

Since the number of hidden layers in the LSTM network model and attention size in the attention module have a great impact on the performance of the proposed method, the CC_WEB_VIDEO public dataset is used to evaluate the performance indicators of different numbers of hidden layers and attention size. The experimental results are shown in Tables 1 and 2.

Table 1. The experimental results of different parameter settings for the number of hidden layers in the LSTM network model.

The Number of Hidden Layers	Precision	Recall	F1-Score	Accuracy
4	0.5577	0.611	0.583	0.7
8	0.9375	0.9375	0.9375	0.95
16	0.9375	0.944	0.941	0.975
32	0.8375	0.8	0.818	0.925

Table 2. The experimental results of different parameter settings for attention size.

Attention Size	Precision	Recall	F1-Score	Accuracy
4	0.7944	0.86	0.826	0.9
8	0.8375	0.9	0.868	0.925
16	0.8819	0.944	0.912	0.95
32	0.9375	0.978	0.941	0.975
64	0.85	0.9	0.874	0.925

It can be seen from Table 2 that it is more hidden layers do not result in higher performance indicators. In the above experiments, the performance indicators are the highest when the number of hidden layers is set to 16. It is considered that because of the limited scale of video data used in the experiments when the number of hidden layers is small, it is challenging to ensure the accuracy of video feature representation. Therefore, all indicators are significantly lower when the number of hidden layers is 4. If there are many hidden layers, the number of nodes in the RCLA model is large, which makes it easy to fall into the local optimization. For example, when the number of hidden layers is 32, all indicators are relatively low. Besides, when the parameter of attention size is set to a small value, the method proposed in this paper focuses on portraying the local features of a small region in a video keyframe. Since the near-duplicate video data have similar but different visual features in the same local area, amplifying the difference in feature representation will affect the performance of this proposed method. When the attention size is large, there is a confusion problem with near-duplicate video recognition causing incorrect video data cleaning. In addition, as shown in Figure 6, although the number of hidden layers and attention size are set to be different, the loss function converges during training, indicating no overfitting. Finally, according to the experimental results in Tables 1 and 2, the number of hidden layers and the attention size in LSTM are set to 16 and 32 in the proposed method.

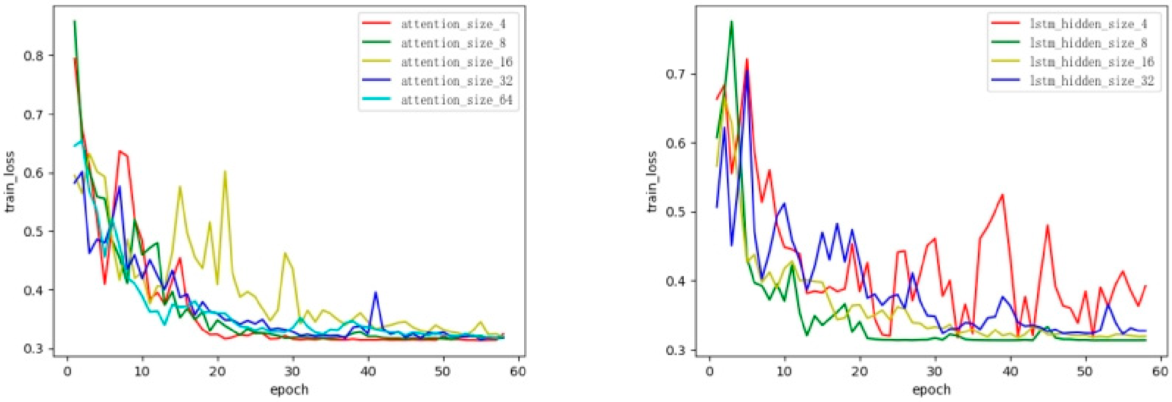


Figure 6. The visualization of loss variation under different parameter settings during training.

In addition, to evaluate the performance of the RCLA model for feature representation of video data, the softmax function is used to achieve the comparison of different feature representation models through the detection of near-duplicate video data. The experimental results are shown in Table 3.

Table 3. The experimental results of different parameter settings for attention size on CC_WEB_VIDEO dataset.

Models	CC_WEB_VIDEO Dataset			
	Precision	**Recall**	**F1-Score**	**Accuracy**
Spatiotemporal Keypoint [3]	0.61	0.96	0.75	0.64
BS-VGG16 [36]	0.79	0.92	0.85	0.85
LBoW [43]	0.63	0.85	0.72	0.66
MLE-MRD [39]	0.82	0.91	0.86	0.87
CBAM-Resnet [42]	0.77	0.92	0.84	0.88
3D-CNN [24]	0.88	0.76	0.84	0.93
RCLA	0.93	0.94	0.94	0.95

It can be seen from Tables 3 and 4 that the hand-crafted feature extraction models of spatiotemporal key points and LBoW have limited capabilities to represent the video features in the near-duplicate video detection task. Hence, all indicators are low. After introducing the CBAM module in each residual block, the ability of video spatial feature extraction is improved using the channel and spatial attention mechanisms. Therefore, all indicators are improved compared with the hand-crafted feature extraction models. Since the video data have the spatiotemporal feature, not only the LSTM deep neural network in the RCLA model is used to extract the temporal feature of the video data, but also the standard attention mechanism is used to enhance the feature representation of the local regions of the near-duplicate video data. Thus, the near-duplicate video can be accurately identified, and the indicators are generally high, but the recall indicator is lower than that of the spatiotemporal keypoint model. It is considered that the spatiotemporal keypoint model can extract the spatiotemporal feature of video data, and the number of detection results is enormous, where the number of correct near-duplicate video data is also massive. Hence, the recall indicator of the spatiotemporal keypoint model is high, but the precision and accuracy indicators are low. On this basis, taking the coal mine video dataset as an example, this paper visually shows the clustering results of 10 types of near-duplicate video data, as shown in Figure 7.

Table 4. The experimental results of different parameter settings for attention size on coal mine video dataset.

Models	The Coal Mine Video Dataset			
	Precision	**Recall**	**F1-Score**	**Accuracy**
Spatiotemporal Keypoint [3]	0.57	0.85	0.68	0.61
BS-VGG16 [36]	0.72	0.83	0.77	0.79
LBoW [43]	0.60	0.92	0.73	0.72
MLE-MRD [39]	0.85	0.84	0.84	0.87
CBAM-Resnet [42]	0.79	0.86	0.82	0.84
3D-CNN [24]	0.91	0.89	0.90	0.90
RCLA	0.93	0.94	0.93	0.92

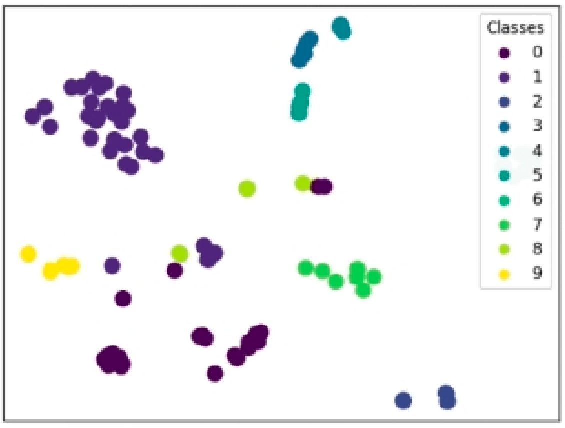


Figure 7. The clustering results of near-duplicate video data in the coal mine video dataset.

The performance of the proposed method (RCLA-HAOPFDMC) is verified in comparison with the existing studies on near-duplicate video data cleaning. The comparison methods are all reproduced through experiments, except that some key parameters are set to the values from those papers, and all other parameters are set to default values. The experimental results are shown in Table 5.

Table 5. The experimental comparison of the different methods.

Methods	Cluster Cleaning	CC_WEB_VIDEO Dataset			The Coal Mine Video Dataset		
		Acc	Rec	F1-Score	Acc	Rec	F1-Score
Spatiotemporal Keypoint [3]	K-Means	0.4527	0.451	0.451	0.466	0.5333	0.497
	FD-Means	0.4776	0.538	0.506	0.666	0.5333	0.592
	FD-Means fused with the peak function	0.522	0.835	0.612	0.857	0.6	0.706
LBoW [43]	K-Means	0.453	0.472	0.462	0.5	0.5333	0.516
	FD-Means	0.587	0.615	0.601	0.733	0.733	0.733
	FD-Means fused with the peak function	0.572	0.813	0.632	0.833	0.5	0.625
BS-VGG16 [36]	K-Means	0.275	0.615	0.436	0.465	0.362	0.382
	FD-Means	0.650	0.929	0.76	0.667	0.833	0.74
	FD-Means fused with the peak function	0.49	0.967	0.633	0.5	0.4	0.444
MLE-MRD [39]	K-Means	0.53	0.62	0.57	0.57	0.65	0.61
	FD-Means	0.72	0.79	0.75	0.69	0.76	0.72
	FD-Means fused with the peak function	0.76	0.82	0.79	0.75	0.86	0.80
CBAM-Resnet [42]	K-Means	0.423	0.56	0.481	0.5333	0.666	0.592
	FD-Means	0.587	0.615	0.601	0.733	0.733	0.733
	FD-Means fused with the peak function	0.825	0.681	0.779	0.777	0.7	0.736
3D-CNN [24]	K-Means	0.75	0.75	0.75	0.71	0.91	0.80
	FD-Means	0.80	0.69	0.74	0.87	0.70	0.77
	FD-Means fused with the peak function	0.875	0.7	0.778	0.936	0.723	0.816
RCLA-HAOPFDMC	K-Means	0.672	0.67	0.671	0.733	0.733	0.733
	FD-Means	0.901	0.802	0.848	0.864	0.9333	0.897
	FD-Means fused with the peak function	0.914	0.801	0.854	0.872	0.9333	0.902

This paper compares the proposed method with the existing studies and different clustering cleaning models, as shown in Table 5. First, it can be seen from Table 5 that the performance indicators of the near-duplicate video cleaning methods based on hand-crafted feature extraction are relatively low, such as spatiotemporal keypoint and LBoW models, which is due to the limited ability of hand-crafted features to represent video features. However, both aforementioned methods require less time, especially the spatiotemporal keypoint model, which solely extracts keypoint features from video frames, thus consuming less time than the LBoW model. Second, the BS-VGG16 model only extracts the spatial features of the video data, and the CBAM-Resnet model introduces the channel and spatial attention mechanisms in the spatial feature extraction. Since the depth of the BS-VGG16 model is only 16 layers, and the residual network depth in the CBAM Resnet model is 31 layers, the time consumed by the BS-VGG16 model is less than that of the MLE-MRD model, while the time consumed by the MLE-MRD model is less than that of the CBAM Resnet model. The 3D-CNN model, due to the lack of attention mechanism introduction, has the time consumed between BS-VGG16 and CBAM Resnet. On this basis, the ACNNBN-LSTM model can extract the spatiotemporal features of the video data, and the RCLA-HAOPFDMC method, based on the spatiotemporal feature extraction to introduce the standard attention mechanism, can more accurately depict the features of near-duplicate video data to help clean the near-duplicate video data accurately and automatically. In addition, by comparing the experimental results of the K-Means, FD-Means, and FD-Means fused with the peak function clustering algorithms, the performance indicators after near-duplicate video cleaning using the K-Means algorithm are low, which is caused by the randomness of the initial clustering center setting. When the FD-Means algorithm is used for near-duplicate video cleaning, the influence of the K value in the K-Means algorithm on the experimental results can be reduced. Thus, the performance indicators are relatively high. This paper constructs a consistent feature hash ring to decrease the impact of data ordering on near-duplicate data cleaning. On this basis, the fusion of the FD-Means algorithm and peak function can further reduce the influence of the random initial cluster center setting on the near-duplicate video cleaning. Therefore, the performance indicators of the proposed method (RCLA-HAOPFDMC) in this paper are higher than those of the existing methods. However, this method requires more computing resources and longer computation time. Finally, the results of near-duplicate video data cleaning are shown in Figure 8.

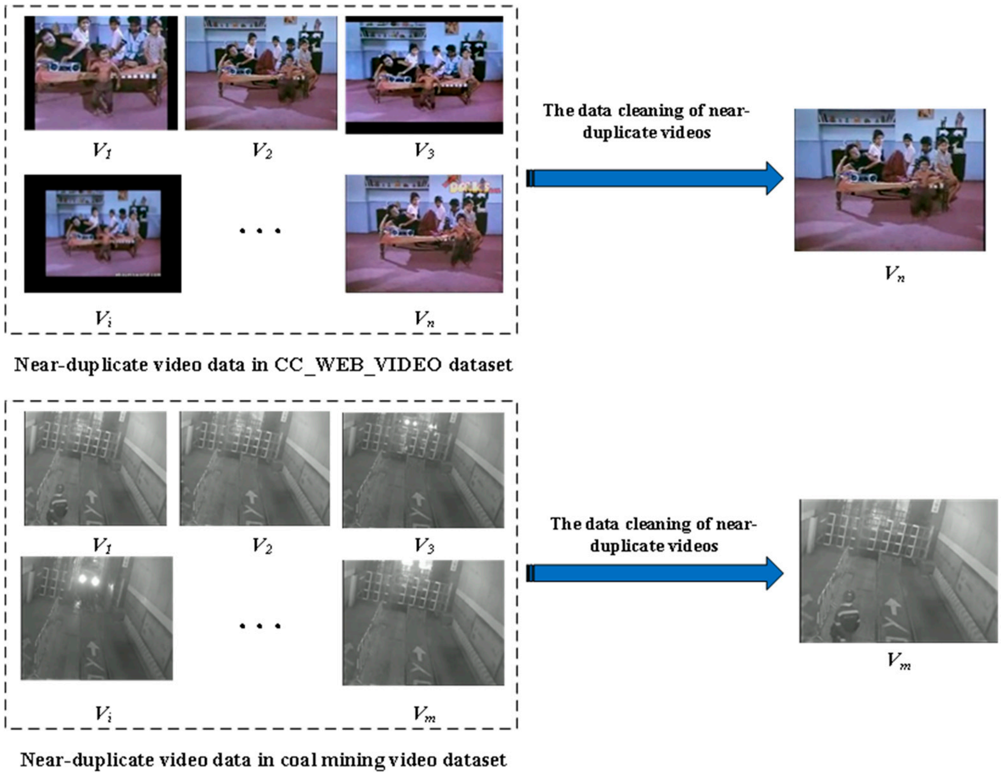


Figure 8. The results of near-duplicate video data cleaning on the CC_WEB_VIDEO and coal mining video datasets.

5. Conclusions

In this paper, an automatic near-duplicate video data cleaning method based on a consistent feature hash ring is proposed, which can be utilized to improve data quality for video datasets. In this method, a novel consistent feature hash ring is constructed to alleviate the sensitivity of video data orderliness. On this basis, an optimized feature distance-means clustering algorithm fusing the mountain peak function on a consistent feature hash ring is used to automatically clean the near-duplicate video data. The experiment results obtained for the CC_WEB_VIDEO and coal mining video datasets demonstrate the advantages of the proposed method, which can achieve automatic cleaning for near-duplicate video data. However, the method proposed in this paper is not an end-to-end deep neural network model that needs to be trained separately in the feature extraction and clustering stages. In addition, the computation of cleaning on the consistent feature hash ring is large. In the future, how to construct an end-to-end near-duplicate video data cleaning method will be explored. Moreover, it is of great interest to introduce the swarm intelligence optimization algorithms in the future to improve the accuracy of near-duplicate video data cleaning by optimizing the parameter selection.

Author Contributions: Software, O.Y.; Validation, Y.Q.; Writing—original draft, Y.Q.; Visualization, Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61873277, and in part by the Chinese Postdoctoral Science Foundation under Grant 2020M673446.

Data Availability Statement: Data are contained within the article.

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

References

1. 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, Beijing, China, 19–23 October 2009; pp. 145–154.
2. Basharat, A.; Zhai, Y.; Shah, M. Content based video matching using spatiotemporal volumes. *Comput. Vis. Image Underst.* **2008**, *110*, 360–377. [CrossRef]
3. Zhang, D.-K.; Sun, Z.-H.; Jia, K.-B. Near-duplicate video detection based on temporal and spatial key points. *Smart Innov. Syst. Technol.* **2020**, *180*, 129–137.
4. Nie, X.; Qiao, J.; Liu, J.; Sun, J.; Li, X.; Liu, W. LLE-based video hashing for video identification. In Proceedings of the IEEE 10th International Conference on Signal Processing, Beijing, China, 24–28 October 2010; pp. 1837–1840.
5. Liu, J.; Huang, Z.; Cai, H.; Shen, H.T.; Ngo, C.W.; Wang, W. Near-duplicate video retrieval: Current research and future trends. *ACM Comput. Surv.* **2013**, *45*, 1–23. [CrossRef]
6. Carvalho, C.; Moreira, R.S.; Jose, J.M. Data quality visual analysis (DQVA) A tool to process and pin spot raw data irregularities. In Proceedings of the 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 27–30 January 2021; pp. 1036–1045.
7. Phalke, D.-A.; Jahirabadkar, S. A survey on near-duplicate video retrieval using deep learning techniques and framework. In Proceedings of the 2020 IEEE Pune Section International Conference (PuneCon), Pune, India, 16–18 December 2020; pp. 124–128.
8. 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]
9. Abdu, S.A.; Yousef, A.H.; Salem, A. Multimodal Video Sentiment Analysis Using Deep Learning Approaches, a Survey. *Inf. Fusion* **2021**, *76*, 204–226. [CrossRef]
10. Mohiuddin, S.; Malakar, S.; Sarkar, R. Duplicate frame detection in forged videos using sequence matching. In Proceedings of the Communications Computational Information Science, Santiniketan, India, 7–8 January 2021; pp. 29–41.
11. Shen, L.; Hong, R.; Hao, Y. Advance on large scale near-duplicate video retrieval. *Front. Comput. Sci.* **2020**, *14*, 145702. [CrossRef]
12. Wang, X.-L.; Luo, J.-S.; Xu, T.; Zeng, G.-B. Research on electricity characteristic recognition method of clean heating based on the big data model. In Proceedings of the Lecture Notes Institutional Computational Science Society Informatics Telecommunication Engineering, Binzhou, China, 26–27 September 2020; pp. 25–35.
13. Zhou, H.-C.; Li, M.-H.; Gu, Z.-Q. Knowledge fusion and spatiotemporal data cleaning: A review. In Proceedings of the 7th IEEE International Conference on Data Science in Cyberspace, Hong Kong, China, 11–13 July 2020; pp. 295–301.
14. Zheng, Z. Contextual data cleaning with ontology FDs. In Proceedings of the ACM SIGMOD International Conference Management Data, Xi'an, China, 20–25 June 2021; pp. 2911–2913.
15. Ye, O.; Li, Z.-L.; Zhang, Y. Near-duplicate video cleansing method based on locality sensitive hashing and the sorted neighborhood method. In Proceedings of the EAI International Conference on Robotic Sensor Networks, Kitakyushu, Japan, 17 August 2019; pp. 129–139.
16. Zhu, Y.; Huang, X.; Huang, Q.; Tian, Q. Large-scale video copy retrieval with temporal-concentration SIFT. *Neurocomputing* **2016**, *187*, 83–91. [CrossRef]
17. Henderson, C.; Lzquierdo, E. Robust feature matching in long-running poor-quality videos. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1161–1174. [CrossRef]
18. Zhang, C.; Liu, D.; Wu, X.; Zhao, G.; Shyu, M.-L.; Peng, Q. Near-duplicate segments-based news web video event mining. *Signal Process.* **2016**, *120*, 26–35. [CrossRef]
19. Chen, Y.; Li, D.; Hua, Y.; He, W. Effective and efficient content redundancy detection of web videos. *IEEE Trans. Big Data* **2021**, *7*, 187–198. [CrossRef]
20. Nie, X.; Jing, W.; Cui, C.; Zhang, C.J.; Zhu, L.; Yin, Y. Joint multi-view hashing for large-scale near-duplicate video retrieval. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1951–1965. [CrossRef]
21. Wang, K.H.; Cheng, C.C.; Chen, Y.L.; Song, Y.; Lai, S.H. Attention-based deep metric learning for near-duplicate video retrieval. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 5360–5367.

22. Li, S.; Chen, Z.; Lu, J.; Li, X.; Zhou, J. Neighborhood preserving hashing for scalable video retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8211–8220.
23. Ai, X.; He, Y.; Hu, Y.; Tian, W. Inter-frame relationship graph based near-duplicate video clip detection method. In Proceedings of the 14th Conference on Image and Graphics Technologies and Applications, IGTA 2019, Beijing, China, 19–20 April 2019; pp. 70–79.
24. 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. [CrossRef] [PubMed]
25. Hu, Y.-Z.; Mu, Z.-K.; Ai, X.-B. STRNN: End-to-end deep learning framework for video partial copy detection. *J. Phys. Conf. Ser.* **2019**, *1237*, 022112. [CrossRef]
26. Mohamed, Z.H.; Vinila, J.S. A comparative study on data cleaning approaches in sentiment analysis. In Proceedings of the Lecture Notes Electrical Engineering, Thiruvananthapuram, India, 14–17 December 2019; pp. 421–431.
27. Ying, K.-H.; Hu, W.-Y.; Chen, J.-B.; Li, G.-N. Research on instance-level data cleaning technology. In Proceedings of the 2021 International Conference on Artificial Intelligence, Big Data and Algorithms, Xi'an, China, 28–30 May 2021; pp. 238–242.
28. Song, S.; Zhang, A.; Wang, J.; Yu, P.S. SCREEN: Stream data cleaning under speed constraints. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, VIC, Australia, 31 May–June 4 2015; pp. 827–841.
29. Tian, Y.C.; Michiardi, P.; Vukolic, M. Bleach: A distributed stream data cleaning system. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 113–120.
30. Qiaomin, L.; Anqi, F.; Min, P.; Qiang, X.; Kun, D.; Michael, S. Cleaning RFID data streams based on l-means clustering method. *J. China Univ. Post Telecom.* **2020**, *27*, 72–81.
31. Ranjan, K.G.; Prusty, B.R.; Jena, D. Comparison of two data cleaning methods as applied to volatile time-series. In Proceedings of the 2019 International Conference on Power Electronics Applications and Technology, Mangalore, India, 29–31 August 2019; pp. 1–6.
32. Zhu, L.; Du, X.; Ma, Q.; Meng, W.; Liu, H. Keyword search with real-time entity resolution in relational databases. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing, Macau, China, 26–28 February 2018; pp. 134–139.
33. Aissani, A.; Yi, E.T.J.; Thamilyanan, T. End to end real-time data cleansing, standardization and reliability towards a smooth digital oil field deployment. In Proceedings of the Offshore Technology Conference Asia, Kuala Lumpur, Malaysia, 2–6 November 2020; pp. 1–10.
34. Zheng, Z.; Milani, M.; Chiang, F. CurrentClean: Spatio-temporal cleaning of stale data. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macau, China, 8–11 April 2019; pp. 172–183.
35. Zhang, Y.; Jin, Z.; Liu, F.; Zhu, W.; Mu, W.; Wang, W. ImageDC: Image data cleaning framework based on deep learning. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), Dalian, China, 20–22 March 2020; pp. 748–752.
36. Fu, Y.; Han, Z.; Ye, O. FD-means clustering cleaning algorithm for near-duplicate videos. *Comput. Eng. Appl.* **2022**, *58*, 197–203.
37. Jia, W.; Li, L.; Li, Z.; Zhao, S.; Liu, S. Scalable hash from triplet loss feature aggregation for video de-duplication. *J. Vis. Commun. Image Represent.* **2020**, *72*, 102908. [CrossRef]
38. Chen, H.; Li, J.; Frintrop, S.; Hu, X. The MSR-video to text dataset with clean annotations. *Comput. Vis. Image Understand.* **2022**, *225*, 103581. [CrossRef]
39. Cheng, H.-Y.; Yu, C.-C. Automatic data cleaning system for large-scale location image database using a multilevel extractor and multiresolution dissimilarity calculation. *IEEE Intell. Syst.* **2021**, *36*, 49–56. [CrossRef]
40. Ye, O.; Deng, J.; Yu, Z.H.; Liu, T.; Dong, L.H. Abnormal event detection via feature expectation subgraph calibrating classification in video surveillance scenes. *IEEE Access* **2020**, *8*, 97564–97575. [CrossRef]
41. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolution block attention module. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
42. Wang, Z. Research on garbage image classification based on neural network. In Proceedings of the 2022 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 23–25 September 2022; pp. 214–217.
43. Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; Kompatsiaris, Y. Near-Duplicate Video Retrieval by Aggregating Intermediate CNN Layers. In Proceedings of the 23rd International Conference on MultiMedia Modeling, Reykjavik, Iceland, 4–6 January 2017; pp. 251–263.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.