applied sciences

MDPI

Article

Interested Keyframe Extraction of Commodity Video Based on Adaptive Clustering Annotation

Guangyi Man * and Xiaoyan Sun *

School of Information and Control Engineering, China University of Mining and Technology,
Xuzhou 221116, China
* Correspondence: mgy_paper@163.com (G.M.); xysun78@126.com (X.S.)

Abstract: Keyframe recognition in video is very important for extracting pivotal information from videos. Numerous studies have been successfully carried out on identifying frames with motion objectives as keyframes. The definition of "keyframe" can be quite different for different requirements. In the field of E-commerce, the keyframes of the products videos should be those interested by a customer and help the customer make correct and quick decisions, which is greatly different from the existing studies. Accordingly, here, we first define the key interested frame of commodity video from the viewpoint of user demand. As there are no annotations on the interested frames, we develop a fast and adaptive clustering strategy to cluster the preprocessed videos into several clusters according to the definition and make an annotation. These annotated samples are utilized to train a deep neural network to obtain the features of key interested frames and achieve the goal of recognition. The performance of the proposed algorithm in effectively recognizing the key interested frames is demonstrated by applying it to some commodity videos fetched from the E-commerce platform.

Keywords: key interested frame; commodity video; clustering; deep neural network

check for updates

Citation: Man, G.; Sun, X. Interested Keyframe Extraction of Commodity Video Based on Adaptive Clustering Annotation. *Appl. Sci.* **2022**, *12*, 1502. https://doi.org/10.3390/app12031502

Academic Editors: Stefanos Ougiaroglou and Dionisis Margaris

Received: 9 December 2021
Accepted: 27 January 2022
Published: 30 January 2022

1. Introduction

Videos have been successfully used in an increasing number of fields due to the development of video technology, including video retrieval [1], recommending interested videos to users in the personalized recommendation [2], recognizing and tracking moving targets based on surveillance videos as pattern recognition [3], and so on. The greatest advantage of applying videos in many scenarios is to continuously record what is happening and provide important information when required. However, it is also quite difficult to directly and efficiently find valuable frames due to the continuously recording. For example, when searching for a hit-and-run vehicle based on a traffic video, the police may spend several days watching every frame to find the target and may also suffer a great failure. If interesting keyframes can be recognized by removing redundant video frames, the workload of the users can be greatly reduced and the success rate of finding key information can be improved. Therefore, extracting the key and valuable frames from video has become one of the research hotspots in video processing [4,5].

In the existing relevant work on keyframe extraction, the keyframe of a video is generally defined as the frame containing the key action changing of the object [6]. The purpose of keyframe extraction is to find a set of images from the original video to represent the main action changes. Through the keyframe, users can understand the behavioral features of the main character or object in a relatively short time [7], and it can provide important information for users to make decisions. There are four kinds of keyframe extraction algorithms—that is, based on target, clustering algorithm, dictionary, and image features. Target-based keyframe extraction transforms the problem into the detection of important objects or people and extracts frames containing important elements from the video as the keyframe. Lee et al. [8] proposed a crucial person or target detection method based on the photographer's perspective and predicted the importance of new targets in video according to the significance region. This kind of algorithm is mainly used to detect important targets from the video. Another kind of keyframe extraction is based on clustering algorithm, which gathers each frame into different categories by designing an appropriate clustering method and selects one or more images in each category as the keyframe [9,10]. This kind of algorithm is simple, intuitive, and easy to implement but often needs to specify parameters such as clustering number or clustering radius, which limits its practical applicability. Dictionary-based keyframe extraction adopts a dictionary to reconstruct video, assuming that the video keyframe sequence is the best dictionary [11]. This kind of algorithm turns the keyframe selection into dictionary learning. Mademils et al. [12] proposed a keyframe extraction algorithm for human motion video based on significance dictionary and reconstructed the whole video using the benchmark of human activity; then, they extracted keyframes according to the dictionary. Furthermore, this sort of frame extraction algorithm pays more attention to the characteristics of the whole video but ignores the uniqueness of individual frames. Feature-based keyframe extraction generally uses the color, texture, or motion feature to realize the recognition of motion information. Zhang et al. [13] put forward a kind of keyframe extraction based on the image color histogram strategy. Yeung and Liu [14] come up with a fast keyframes recognition method based on the maximum distance of feature space. Meanwhile, Lai and Yi [15] extracted movement, color, and texture characteristics of frames and built dynamic and static significant mapping, improving the priority of movement information, which enhanced the extraction effect of motion information. Li et al. [11] mapped the video frame features into the abstract space, and then extracted keyframes in it. Existing keyframe extraction algorithms have achieved good results in motion target detection, key figure detection, creating video abstracts, and other fields but these algorithms either lack static feature extraction or pay attention to the overall features of the video while ignoring the uniqueness of a single frame or needing prior parameters that greatly limits the application value of keyframe extraction. In addition, these studies mainly focused on the movement information in the video and did not consider individual demands.

In fact, different users have different concerns when browsing a video, so extracting keyframes only by motion information is difficult to meet user's demands. For instance, on the e-commerce platform [16], the commodity video provides vital information for users searching for their needs, and the key is how the user can obtain the interesting information about the video in a very short time. Such information goes beyond movement changes, as representing global and local content about products is more crucial. Besides, the commodity video is commonly short video and movement of the object is not obvious, so traditional keyframe extraction algorithms based on motion changes no longer apply. At the same time, the definition of keyframe regarding the commodity video varies from person to person. Therefore, how to extract video keyframes according to the needs of different users has become a challenge to the traditional algorithm.

In recent years, many keyframe extraction algorithms have focused on adding an attention mechanism. Shih [17] designed an attention model based on semantic and visual information to mark frames, and selected frames with high scores as keyframes. Although the attention mechanism is integrated, there is no solution to deal with different users' interests.

Aiming at the above problems, we propose an algorithm of commodity video keyframe recognition based on adaptive clustering annotation to solve commodity video keyframe extraction. First of all, from the perspective of users' demands, the keyframe is defined as the frame containing global and local information that users are interested in, and these frames could not include noise and blur. Then, the frame-to-frame difference method is used to obtain the differential frame set by looking for the maximum difference between frames, where a differential frame is defined as the frame that has the greater difference and object movement compared to adjacent frames. For the set, an efficient and adaptive clustering strategy with few parameters based on frame difference degree and category scale is designed to realize the clustering of different frames and the differential frame sets are divided and annotated based on the keyframe definition defined by users' requirements. Furthermore, a small number of marked commodity keyframe samples are utilized to train the deep neural network by means of transfer learning [18] to realize accurate keyframe recognition and extraction.

Contributions of this paper mainly include the following four parts: (1) From the perspective of users' attention and interest in commodities, we propose the definition of commodity video keyframe. (2) An adaptive image clustering strategy based on the frame-to-frame difference and keyframe labeling method are proposed. (3) A deep neural network model is presented fusing the frame-to-frame difference with the transfer mechanism to realize the extraction of commodity video keyframes. (4) The proposed algorithm is applied to the self-built video library of clothing commodities, and the results show its effectiveness in meeting the personalized needs of users. Figure 1 shows the algorithm framework of our work.

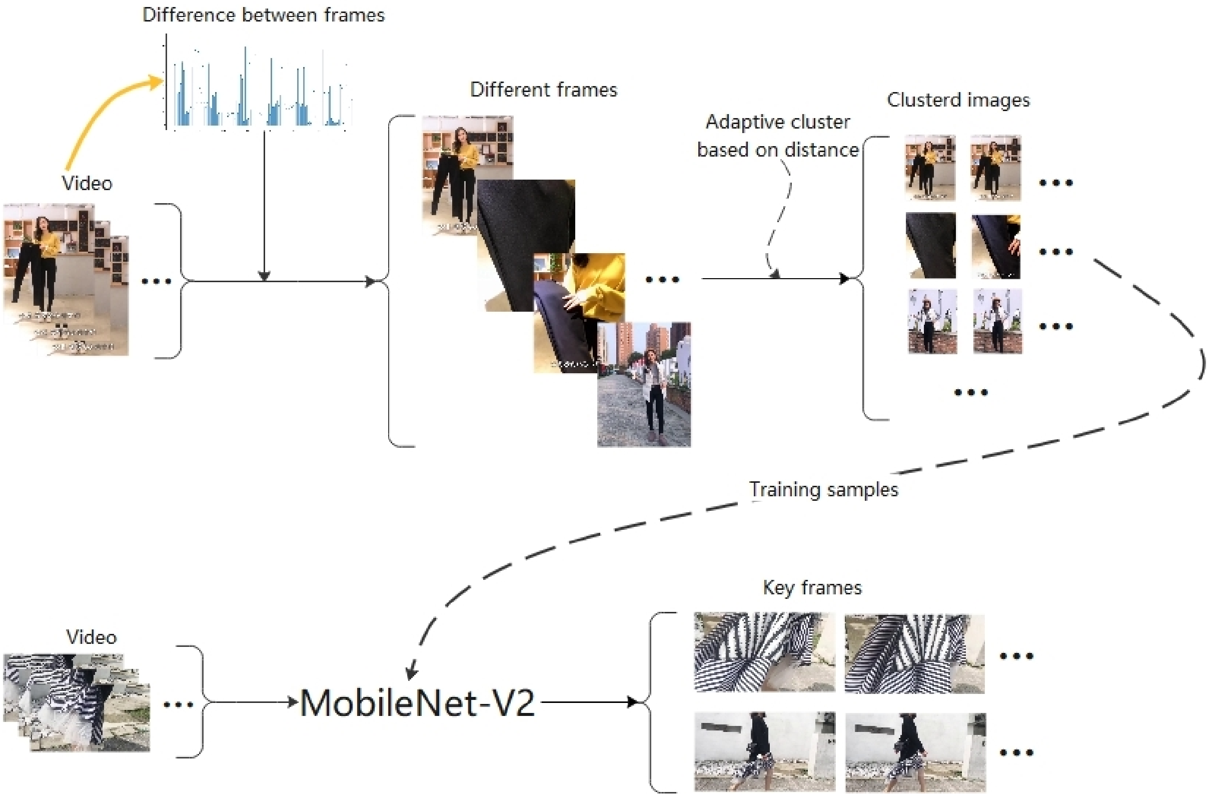


Figure 1. Proposed algorithm framework.

The paper is organized as follows: In Section 2, we provide a review of related work about keyframe extraction, image clustering, and deep neural network algorithm. In Section 3, we introduce algorithms proposed in this paper in detail, including the adaptive clustering strategy (Adaptive Cluster Based on Distance, ACBD), commodity video keyframe labeling algorithm, and keyframe extraction strategy. In Section 4, we evaluate the algorithm on the commodity dataset and analyze the results. In the Discussion section, we discuss the advantages and significance of our algorithm. In the Conclusion section, we conclude the paper.

2. Related Work

Keyframe extraction has become a hotspot in video processing technology nowadays and mainly uses clustering strategies, such as K-means clustering [19], mean-Shift clustering, density clustering [20], fuzzy C-mean clustering [21], etc. to generate video summaries and retrieve information. Generally, in the field of video retrieval, the video is usually transformed into keyframes so as to improve the efficiency. Sze et al. [22] proposed a keyframe extraction algorithm based on pixel features, which improved the retrieval performance compared with the traditional algorithm based on histogram feature. Pan et al. [21] proposed a keyframe extraction algorithm based on improved fuzzy C-means clustering, extracting frame images from each class as keyframes according to the maximum entropy. Xiao et al. [23] dynamically divided frames into different classes in accordance with the captured content and selected the frame closest to the class center as the keyframe. Wang and Zhu [24] extracted a moving target from the original video as keyframes through lens boundary detection. Chen et al. [25] used posture information to identify and select keyframes with abrupt posture changes on the basis of human targets, in order to make the network have adaptive ability in attitude changes and, thus, extract keyframes having motion information.

Normally, in the field of generating video summary, the extracted keyframe is used as the video summary. Ren et al. [26] divided the video into different segments according to lenses; then, they clustered different video segments, and finally, selected several frames from every category as keyframes of the video. In general, image features are usually fused into the keyframe extraction algorithm based on clustering. For instance, Gharbi et al. [10] extracted SURF features from video frames and then clustered features to extract keyframes. Likewise, Mahmoud et al. [27] extracted and clustered color features, and sequentially selected the center of each category as keyframes of the video. Liu et al. [28] designed feature description windows to extract the objects and reduced the number of windows through prior knowledge to reduce the possibility of overfitting. Gygli et al. [29] considered multiple target detection; they extracted the features of different targets through supervised learning and fused multiple features to extract the keyframe. Li et al. [11] extracted a frame per second to greatly shorten the length of the video, thus improving the efficiency of the algorithm.

With the rapid development of deep learning in the field of video processing, some scholars applied it to extract and recognize keyframe feature. Zhao et al. [30] used Recurrent Neural Networks (RNN) to extract video keyframes, which make the keyframe sequence represent the semantic information about the original video better. Agyeman et al. [31] extracted video features by 3D-CNN and ResNet and then recognized keyframes using a Long Short-Term Memory (LSTM) network trained by these features. Universally, RNN and LSTM are used to process time series data. Although video is a kind of time series data, for commodity video keyframe extraction, users do not focus on the temporal characteristics between two frames and pay more attention to the contents of each image. Therefore, we should consider other neural network models to extract image features that users are interested in to improve the accuracy of keyframe extraction.

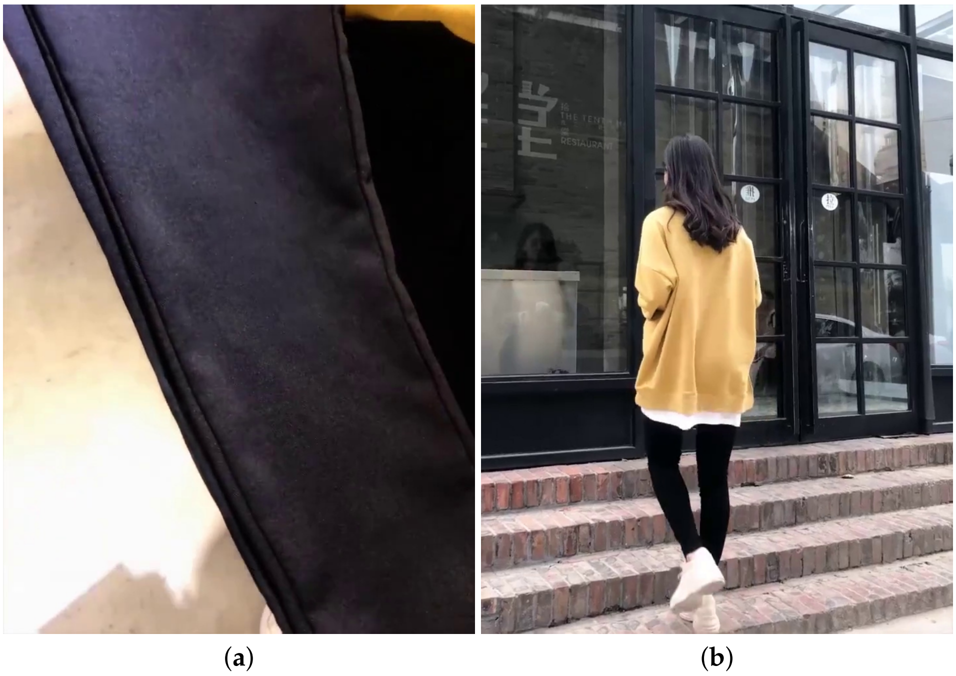
At present, the convolutional neural network is mainly used to extract image features. Since AlexNet [32] made a breakthrough, the architecture of convolutional neural network (CNN) has been getting deeper and deeper. For example, VGG [33] network and GoogleNet [34] have reached 19 and 22 layers, respectively. However, with the increase in network depth, the problem of ineffective learning caused by gradient vanishing will lead to the saturation or even degradation of the performance of the deep convolutional neural network. Hence, He K. et al. proposed a deep residual network (ResNet) [35], adding the identity mapping and relying on the residual module to overcome the gradient disappearance and enhance the convergence of the algorithm. There are five models of ResNet network, including 18 layers, 34 layers, 50 layers, 101 layers, and 152 layers. With the network layers being deepened, the fitting ability of models is gradually improving but the training complexity is increasing sharply. From the perspective of reducing the complexity of deep convolutional neural network, Howard et al. proposed the structure of MobileNet [36], which greatly reduced the number of training parameters through the use of deep separable convolution. Further, on the basis of this network, Sandler et al. integrated the residual module and proposed MobileNet-V2 [37] with higher accuracy. On this basis, Gavai et al. [38] used MobileNet to classify flower images, and Yuan et al. [39] used Mobilenet-V2 to detect surface defects of galvanized sheet. Fu et al. [40] extracted text information from natural scenes by combining Mobilenet-V2 and U-NET. On account of the high efficiency and accuracy of Mobilenet-V2 in image feature extraction, we apply MobileNet-V2 to extract image features of commodity video keyframes in order to obtain frame features reflecting users' interest.

3. Keyframe Extraction by Adaptive Clustering Annotation Facing User's Interestingness

3.1. Keyframe Definition for User's Interest

In commodity videos, users will concentrate on images that reflect the global and local information of commodities. In addition, the clarity and quality of images will also greatly affect user experience and decision-making. Therefore, we first provide quantitative descriptions of image quality and the information mentioned previously and then define the keyframe based on the descriptions.

Figure 2 illustrates the global and local information of two frames. Visibly, the image containing global commodity information has many contour features and, oppositely, the other image lacks these features. Thus, we adopt the Laplace operator, which can embody contour features of images to define the global and local features of images that express user's interest.



(**a**) (**b**)

Figure 2. (**a**) Local feature, (**b**) Global feature. Local features reflect commodity details including texture, material, workmanship, etc. and global features reflect the overall effect of the commodity on the model.

The pixel point matrix of the commodity video frame is denoted as $f(u,v)$, where u and v represent the row and column of the image, and its Laplace transform is shown in Equation (1) [41].

$$\begin{aligned}&\nabla^2 f(u,v) = [f(u+1,v)+f(u-1,v)+f(u,v+1)+f(u,v-1)]-4f(u,v),\\&u\in\{1,2,\ldots,h\}, v\in\{1,2,\ldots,w\}\end{aligned} \quad (1)$$

In the Equation (1), h and w represent the sum of row and column, respectively. Further, we binarize the image after Laplace transform and then count the number of white points in the binary image to calculate the ratio denoted as $\eta = \frac{N_w}{N_t}$, which reflects the richness of contour information, where N_w and N_t represent the number of white points and total pixels, respectively. When η is large, it indicates that the frame of commodity video contains more contour features and likely reflects the global feature of the commodity. On the contrary, the frame may only contain the local features of goods. For example, for two frames in Figure 2, the η is 0.0029 and 0.028, respectively, showing large differences.

Mean value, standard deviation, and mean gradient are commonly used to measure image quality and describe the brightness, the color saturation, and the clarity of image, respectively. In commercial video frames, poor image quality is caused by blur and lack of the main object due to lens conversion, so the mean gradient can be used. Meanwhile, lack of the main object results in a large area of blank and the average gradient of blank part is 0, resulting in a small average mean of the whole image. For this reason, we adopt mean gradient to evaluate the quality of commodity video frames, as in Equation (2). In the equation, respectively, w and h represent the width and height of the picture; $\frac{\partial f(u,v)}{\partial x}$ and $\frac{\partial f(u,v)}{\partial y}$ represent horizontal gradient and vertical gradient. When the mean gradient of a frame is greater than the mean value of all frames' gradients, the image quality is higher.

$$G = \frac{1}{w \times h} \sum_{u=1}^{h} \sum_{v=1}^{w} \sqrt{\frac{(\frac{\partial f(u,v)}{\partial x})^2 + (\frac{\partial f(u,v)}{\partial y})^2}{2}} \tag{2}$$

Nonetheless, we encounter two problems using the above quantitative description in practical application. That is, (1) when a commodity has many designs, its image has abundant local and global contour features. At this condition, the η of every frame are similar to each other, so the global and local features cannot be completely divided by η. (2) When mean gradient is used to measure the image quality, the mean gradient of frames without complicated designs is similar to the value of frames that lack the main object, and these frames may be misclassified into low-quality. Therefore, we introduce user's interest to make up for the deficiency of the quantitative descriptions—that is, on the basis of these descriptions, every user participates in the judgment of the global information, local information, and image quality. In general, the commodity video keyframe is determined by η, G and the correction operations of the user.

In order to accurately identify keyframes in commodity video, our algorithm is divided into the following steps: (1) We extract a part of frames according to frame-to-frame difference and design an efficient automatic clustering strategy to cluster these frames. (2) We calculate η and G based on categories to make the clustering result more accurate by means of introducing local features, global features, and quality evaluation and then submit them to users for correction. Finally, we obtain reliable keyframe labels. (3) Aiming to extract keyframe features, we utilize a small number of keyframes containing labels to train Mobilenet-V2 by transfer learning and, finally, we can obtain keyframes using the network trained before.

3.2. Personalized Keyframe Adaptive Clustering Based on Frame-to-Frame Difference

Before extracting keyframes reflecting personalized demands, frames should be annotated. Obviously, it is time-consuming and difficult for users to annotate each frame. Thus, we expect to reduce this burden with the method of frame-to-frame difference, which can be used to identify the frames of shot transition and model's movements as the important information of the video. Therefore, we use it to measure the difference between two adjacent frames, and then design the Adaptive Cluster Based on Distance (ACBD) to shorten the labeling process.

3.2.1. Differential Frames Extraction

We define the set of all video frames as $\Psi = \{X_1, X_2, \ldots, X_n\}$, and after grayscaling the Ψ, it is expressed as $\Psi' = \{X'_1, X'_2, \ldots, X'_n\}$. Then, the frame-to-frame difference between frame i and frame $i - 1$, denoted as $D(i)$, is shown in Equation (3).

$$D(i) = \frac{\sum_{m=1}^{w}\sum_{j=1}^{h} \|X_i'(j,m) - X_{i-1}'(j,m)\|}{w \times h}, i \in \{2,3,\ldots,n\} \tag{3}$$

Frame-to-frame difference is often used to detect object movement. The frame having larger difference value than the threshold is selected as a keyframe. In our work, we adopt a differential frame selection strategy based on maximum value. For example, $D(i) \geq \frac{D(i-1)+D(i+1)}{2}$ reveals that the frame X_i has great difference from other frames in $\{X_{i-2}, X_{i-1}, X_i, X_{i+1}\}$, so we consider that X_i is a differential frame. The set of differential frames is denoted as $\Psi_d = \{X_1, X_2, \ldots, X_L\}$. In order to filter redundant information ulteriorly, we set the threshold α, and when $L \leq \alpha$, the algorithm outputs the result; otherwise, the algorithm repeats the above process of differential frame extraction on Ψ_d.

3.2.2. Adaptive Clustering of Differential Frames

In order to improve annotation efficiency, we propose an Adaptive Cluster Based on Distance (ACBD) to cluster the set of differential frames and select the class center for user annotation. The ACBD algorithm uses Euclidean distance to measure the similarity between images and divides the categories through dynamic threshold to achieve accurate clustering of images. Assume that the set to be clustered is a differential frame set, as $\Psi_d = \{X_1, X_2, \ldots, X_L\}$. The ACBD algorithm is completed in four steps: (1) Obtain the initial classes—that is, calculate the similarity of the images and group L images into $L-1$ classes. (2) Remove noise. (3) Combine related categories and delete abnormal images by intraclass difference. (4) According to the number of samples of every class, dynamically adjust the cluster threshold to optimize the clustering results. The details are shown as follows.

(1) Obtain the initial classes. We transform the image X_i to row vector, as $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,(w\times h)}]$, and the Euclidean distance between any two frames is shown in Formula (4).

$$dis(X_m, X_l) = \sqrt{\sum_{i=1}^{w\times h} (x_{m,i} - X_{l,i})^2}, \quad m \in \{1,2,\ldots,L\}, l \in \{1,2,\ldots,L\} \tag{4}$$

Thus, we can obtain the image distance matrix of Ψ_d represented by $Dis(\Psi_d)$.

$$Dis(\Psi_d) = \begin{bmatrix} 0 & dis(X_1,X_2) & \cdots & dis(X_1,X_L) \\ dis(X_2,X_1) & 0 & \cdots & dis(X_2,X_L) \\ \vdots & \vdots & \ddots & \vdots \\ dis(X_L,X_1) & dis(X_L,X_2) & \cdots & 0 \end{bmatrix}$$

Afterwards, we calculate the mean of $Dis(\Psi_d)$ by Formula (5).

$$E = \frac{\sum_{j=i+1}^{L}\sum_{i=1}^{L} dis(X_i, X_j)}{C_L^2} \tag{5}$$

$$dis(X_p, X_q) < K \tag{6}$$

For each image, we gather its similar images into a group. According to Formula (6) in which K equals E for the first time, we group Ψ_d into $L-1$ classes denoted as $A = \{A_1, A_2, \ldots, A_{L-1}\}$. For example, the clustering result of image i is represented as $A_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,p}\}, i = 1,2,\ldots,L-1$. $A_{i,p}$ represents that the image p is similar to the image i.

(2) Remove noise. Firstly, we calculate the number of elements in A_i denoted as a_i and set the threshold β to divide noise. If $a_i < \beta$, it means that images in A_i are quite different from most other images and the number of elements in it is not enough to be considered a separate cluster, so we determine A_i as a noise and delete it. Besides, for $A_{i,q}, q = 1, 2, \ldots, p$ we seek similar elements in A_i and count them denoted as $Count_q$ to delete $A_{i,q}$ if $Count_q$ is less than β.

(3) Merge categories based on associated images: We assume that sets with the same elements describe the same class, and therefore merge the intersecting sets. For example, a sample belongs to A_i and A_j, so we merge them to a category. And the result is denoted as $A' = \{A'_1, A'_2, A'_3, \ldots, A'_m\}$.

(4) Optimize cluster based on dynamical cluster threshold. In Formula (6), K is usually unable to accurately cluster different clusters, so we have to update K to improve accuracy. For this, we count the number of elements in $A'_t, t = 1, 2, \ldots, m$ denoted as Num_t and we set a threshold μ to change Formula (6) into Formula (7) so as to recalculate A and redo step (2) and step (3) if $\frac{Num_t}{L} > \theta, t = 1, 2, \ldots, m$. θ is a decimal representing the percentage of each category in the video.

$$dis(A_p, A_q) < \mu K \tag{7}$$

Finally, we can obtain the result until K stops changing or the number of iteration ε is reached. After clustering, we calculate η and G to separate images into containing local features, containing global features, and containing distorted information. Then, these images are sent to users to correct labels, so this process integrates user's preferences to achieve personalized keyframe extraction. The ACBD algorithm framework is shown in Algorithm 1.

Algorithm 1 The process of ACBD.

```
Input: The image set $\Psi_d = \{X_1, X_2, \ldots, X_L\}$
Output: Clustered images
 Step 1: Transform the image $X_i$ to row vector $x_i$ and get $Dis(\Psi_d)$ and $E$. Obtain the initial classes $A = \{A_1, A_2, \ldots, A_{L-1}\}$ according to Formula (6)
 Step 2:
 for $i$ from 1 to $L-1$ do
     Count$a_i$ of $A_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,p}\}$
     if $a_i < \beta$ then
         Delete $A_i$
     for $q$ from 1 ti $p$ do
         Count $Count_q$
         if $Count_q < \beta$ then
             Delete $A_{i,q}$
 Step 3: Merge categories that have same elements and get the result $A' = \{A'_1, A'_2, A'_3, \ldots, A'_m\}$
 Step 4:
 for $t$ from 1 to $m$ do
     Count $Num_t$
     if $\frac{Num_t}{L} > \theta$ then
         According to Formula (7), change $K$ to $\mu K$ to recalculate $A$ and redo step (2) and step (3)
     else
         return $A' = \{A'_1, A'_2, A'_3, \ldots, A'_m\}$
```

3.3. The Extraction of Keyframes of Interest Combined with Frame-to-Frame Difference and Deep Learning

After obtaining labeled images, it is time to train the network used for keyframe extraction. Considering the practicability of the network on mobile, the MobileNet-V2 network model is used in this paper, because under the condition of high accuracy, compared with Resnet-50, MobileNet-V2 runs faster and has fewer parameters.

However, the network may not be adequately trained only using a self-built dataset. Therefore, inspired by transfer learning [42], we use the pretrained model of ImageNet to retrain and fine-tune the network parameters on our dataset. After differential frame extraction and neural network classification, the keyframe containing user interest information is finally obtained.

4. Experiment

In this part, we verify the effectiveness and rationality of the algorithm proposed in this paper through experiments. Firstly, we design experiments to verify the effectiveness of the differential frame extraction algorithm and the adaptive clustering algorithm, and compare the difference between MobileNet-V2 and RESNET-50 in commodity video keyframe recognition. Finally, we give the overall output of the proposed algorithm. The experiment will be run and tested on the Clothes Video Dataset constructed in this paper.

4.1. The Experiment Background

In recent years, the volume of commodity video data has grown rapidly, but there is no publicly available commodity video dataset. Therefore, we downloaded 30 videos of each jacket, pants, shoes, and hat from Taobao (www.taobao.com) to construct a new Clothes Video Dataset, which contains 120 videos, and the length of videos ranging from ten seconds to one minute. The basic parameters of the dataset are shown in Table 1.

Table 1. Basic parameters of the Clothes Video Dataset.

	Number	Length
jacket	30	1 m 12 s
pants	30	49 s
hat	30	53 s
shoes	30	47 s

We consider that for the product, the details are the most important part, and the overall effect is second. Therefore, commodity video frames are divided into four categories: the first category shows commodity details reflecting texture, material, workmanship, etc.; the second category shows the overall effect of the commodity on the model; the third category contains a lot of information unrelated to commodities, such as scenes and models' faces; the fourth category is the distorted image. Thus, the first and second categories are defined as the keyframe of the video. We randomly selected 3 videos from each category in the dataset, a total of 12 videos, and then obtained the initial clustering by ACBD algorithm. Next, we corrected and divided the clustering results into the four categories mentioned before to obtain 790 images of the first category, 247 images of the second category, 58 images of the third category, and 45 images of the fourth category.

We used an Intel Core i7-8700K CPU with 64 GB RAM and an NVIDIA GeForce GTX 1080Ti graphics card. For the software environment, we used Python version 3.7. The parameters α, β, μ, θ, ε used in the experiment are 200, 3, 0.95, 1/3, and 5, respectively, through many experiments.

4.2. Differential Frame Extraction

We compared the algorithm proposed by Li et al. [11] with our algorithm for differential frame extraction to analyze the advantages and disadvantages. The algorithm proposed by Li et al. extracts one frame of image per second. The essence of the algorithm is to extract video frames at the same time interval. In this experiment, we set the extraction interval as 4 frames; the duration of the experiment video is 60 s and contains 1501 frames.

As can be seen in Figure 3, the video of 1501 frames is reduced to 176 frames, which greatly reduces the amount of data for subsequent processing and retains the main information of the video. By comparing Figures 4 and 5, a large number of frames remain after equal interval extraction, so it can be seen that the interval size of 4 frames is not reasonable, and the extraction interval should be expanded to improve the result. In practical application, the video length is often unknown, so it is impossible to determine the size of the extracted interval. Therefore, the differential frame extraction algorithm is more applicable.

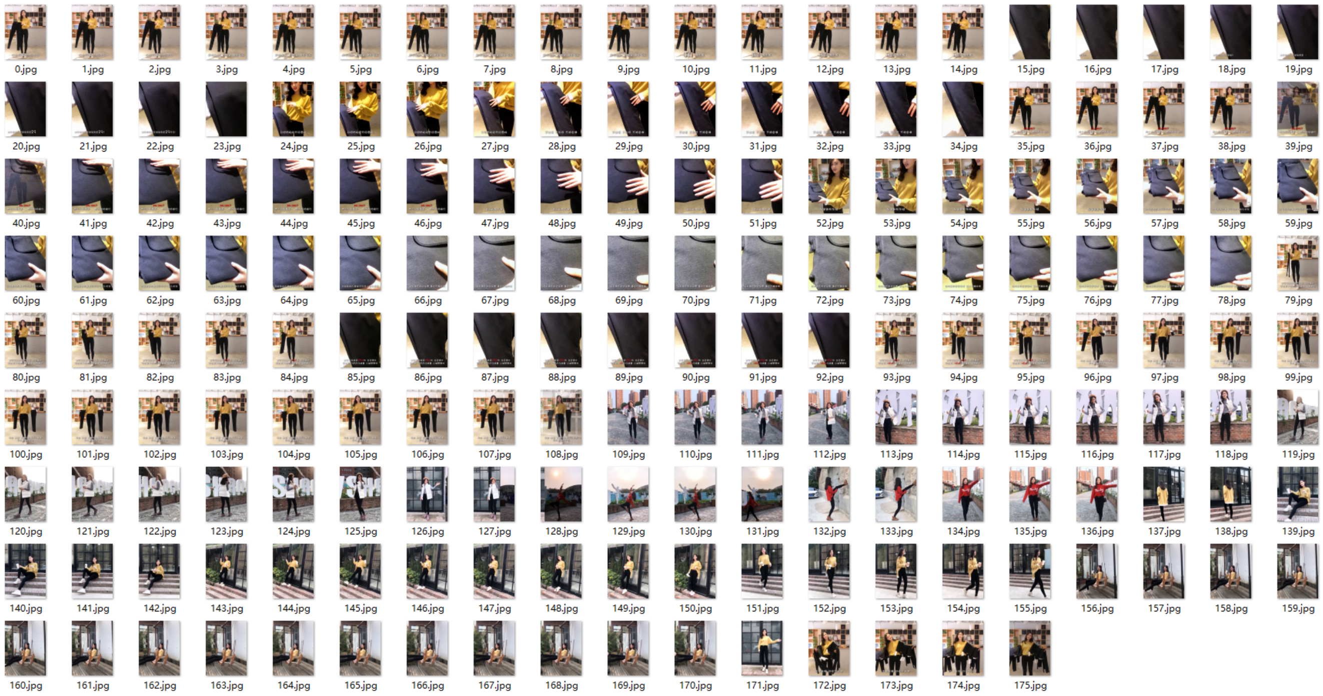


Figure 3. Extraction results of differential frames.

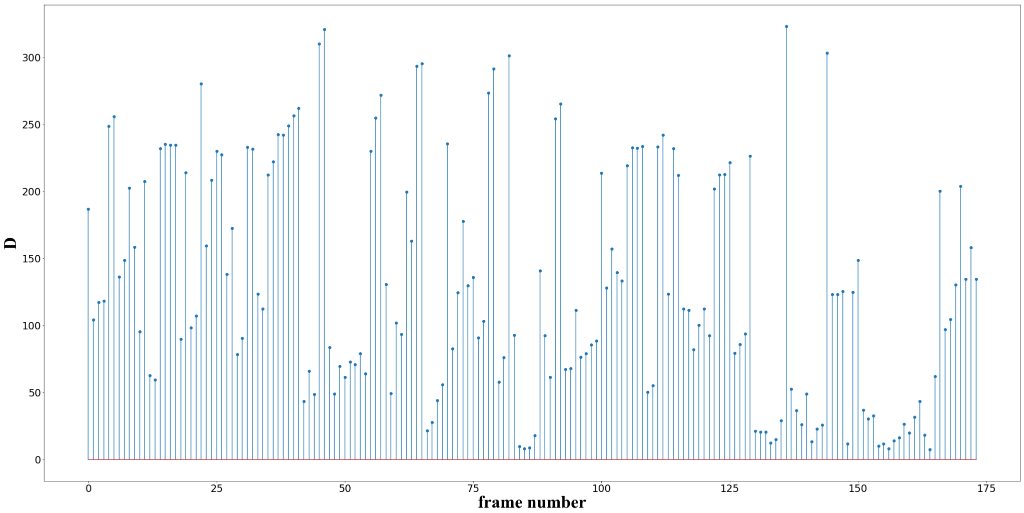


Figure 4. Interframe difference after differential frame extraction.

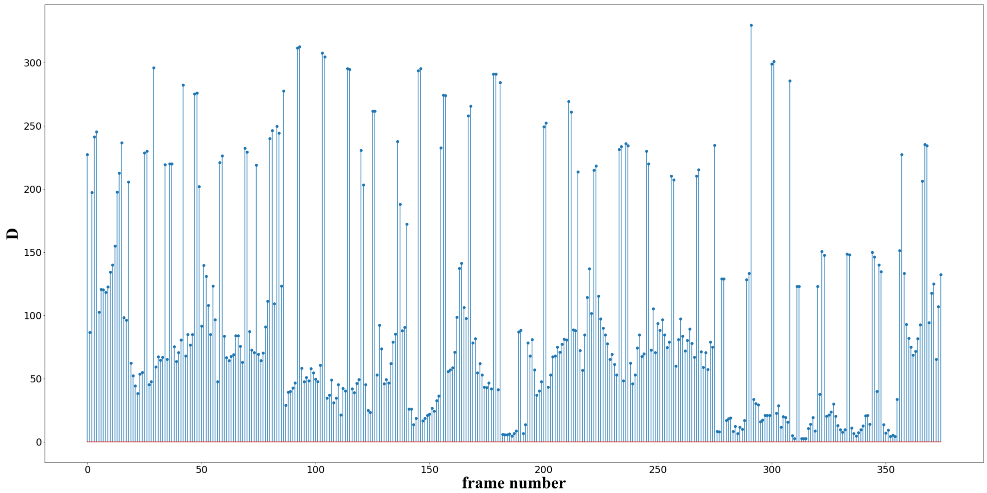


Figure 5. Interframe difference after equal interval extraction.

4.3. Acbd Algorithm

We designed three experiments to verify the effectiveness of our proposed ACBD algorithm. First, we conducted experiments about the effect of ACBD, then compared the effect of the ACBD algorithm with the DBSCAN algorithm on the commodity dataset constructed in this paper and tested its applicability with DBSCAN and K-Means on the UCI dataset.

4.3.1. Effect of ACBD

We used the above video for experiment and used the ACBD algorithm to cluster the obtained differential frames.

From the distribution of Figure 6b, it can be clearly seen that compared with Figure 6a, the ACBD algorithm deletes some differential frames and leaves 106 frames. The deleted frames are mainly the shot transition frames shown in Figure 7, and the transition frames contain a lot of useless and even misleading information, which cannot be used for subsequent algorithm processing. The removal of transition frames can improve the accuracy and efficiency; thus, the ACBD algorithm has the function of removing noise. Comparing the clustering results shown in Figure 8 with the difference frames in Figure 3, the ACBD algorithm removes the 128th to 136th pictures in Figure 3. According to statistics, the proportion of this scene in the video is 5.1%. For commodity videos, it can be considered that the scene with a low proportion is less descriptive for the commodity, so it can be deleted. From the clustering results, the ACBD algorithm can distinguish each cluster and achieve accurate clustering of differential frames.

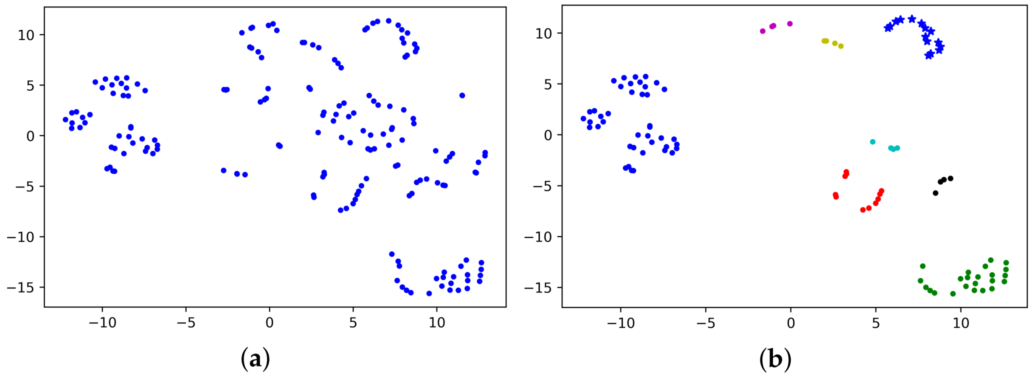


Figure 6. (**a**) Differential frame distribution. (**b**) Distribution after clustering.

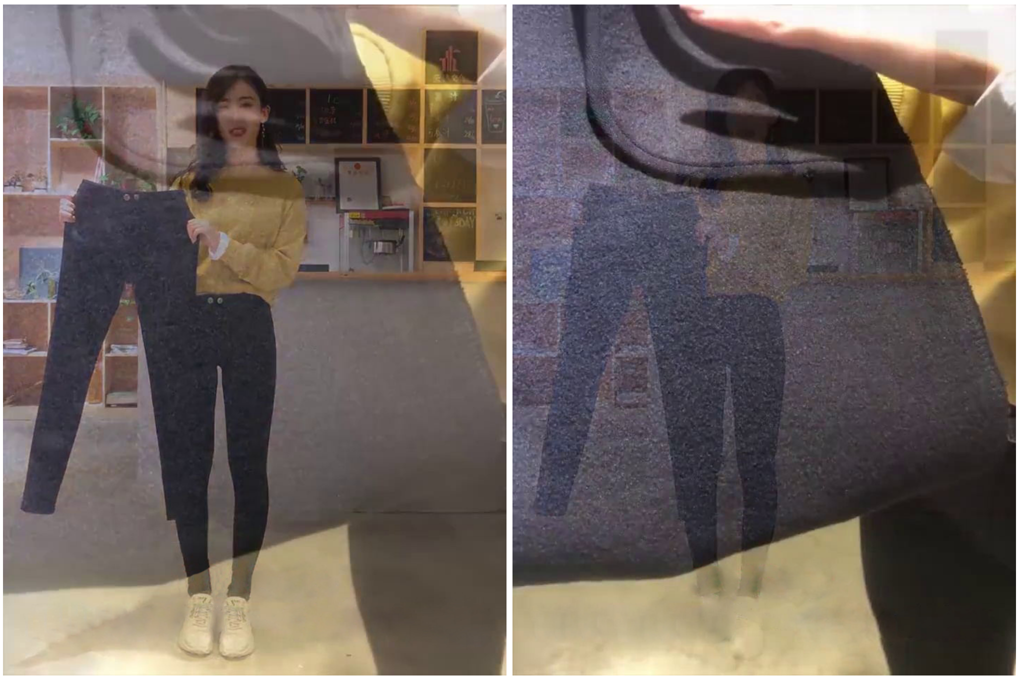


Figure 7. Transition frames.

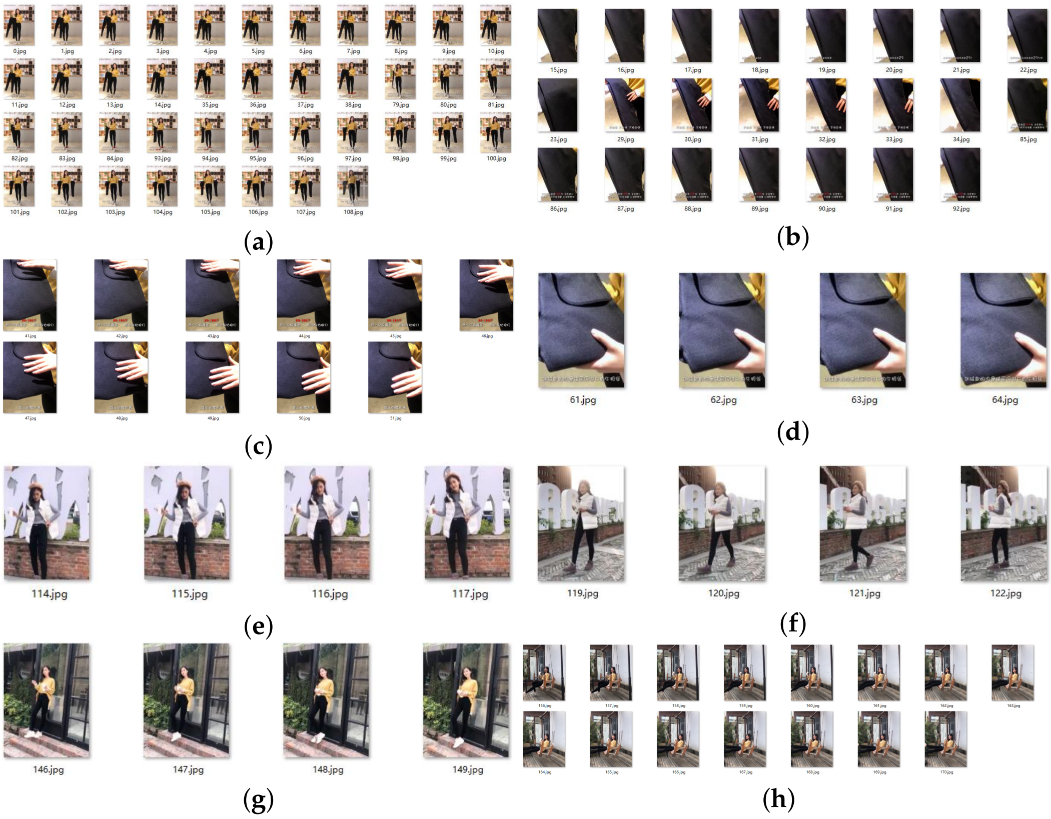
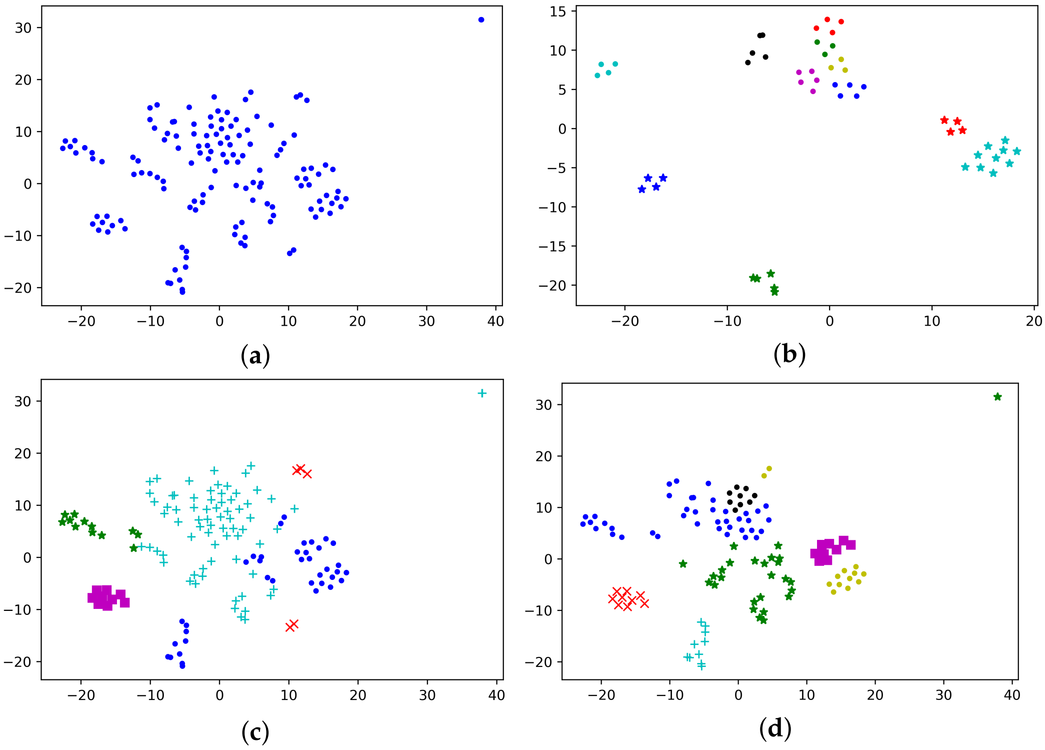


Figure 8. ACBD clusters the differential frames into 8 clusters (**a**–**h**).

4.3.2. Compared with DBSCAN and K-Means

In the commodity dataset, we compared ACBD with the common clustering algorithms K-means and DBSCAN to judge whether the ACBD algorithm is superior. We randomly selected a video from the dataset for experiment, and the clustering radius of DBSCAN was determined according to the mean distance of all images. The radius in this experiment was 20,000, and the minimum number of sample points was set to 3 after repeated debugging. The number of K-means clustering was set as 4.

From Figure 9a,b, we can see that, compared with the original video, DBSCAN algorithm abandoned many differential frames resulting in the loss of some important information. In addition, the DBSCAN algorithm needs to adjust the clustering radius and density artificially, and readjust the parameters for different videos, which greatly increases the workload. By comparing (c) and (d), the number of categories represented by "+" in the K-means clustering results is far greater than other categories, so the results are not accurate enough. However, the categories represented by "·" and "×" may be misclassified due to the long distance within classes. We also cannot determine the number of clusters for different videos in advance. It can be seen from the result presented in (d) that ACBD can solve the problems existing in K-means well. For the outliers in the upper right corner of Figure 9, the image has a small number of transition regions due to the scene transition and becomes an outlier after dimensionality reduction. However, this image should be classified into the green "·" category in Figure (d) after comparison one by one.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 9. Comparison of clustering effects between the ACBD algorithm and DBSCAN algorithm. (**a**) Differential frame distribution. (**b**) DBSCAN algorithm clustering results. (**c**) K-means algorithm clustering results. (**d**) ACBD algorithm clustering results.

4.3.3. Effects on the UCI Dataset

In order to test the applicability of the ACBD algorithm, we verified the effect on IRIS, WINE, and HAPT datasets and compared it with K-means algorithm and DBSCAN algorithm. We used Adjusted Rand index (ARI) [43], Fowlkes–Mallows index (FMI) [44], and Adjusted Mutual Information (AMI) [45] to evaluate the clustering results, and these evaluation criteria are defined as follows. For a given set S of n instances, assume that $U = \{u_1, u_2, \ldots, u_R\}$ represents the ground-truth classes of S and $V = \{v_1, v_2, \ldots, v_C\}$ represents the result of the clustering algorithm. n_{ij} represents the number of instances in u_i and v_j. $n_{i.}$ and $n_{j.}$ represent the number of instances in u_i and v_j, respectively. Rand index (RI) is calculated by $\sum_{i,j}\binom{n_{ij}}{2}$, and thus, ARI Can be expressed as below.

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)} \tag{8}$$

In Formula (8), $E(RI) = \frac{[\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}]}{\binom{n}{2}}$, $max(RI) = \frac{1}{2}[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}]$. ARI is used to measure the degree of consistency between the two data distributions, and its range is [−1,1]. In order to measure the effect of the clustering algorithm more comprehensively, FMI and AMI are introduced for evaluation through different methods. Their value range is [0,1] and, the larger the value is, the more similar are the clustering results to the ground-truth. FMI and AMI can be expressed by Formulas (9) and (10), respectively.

$$FMI = \frac{\sum_{i,j} \binom{n_{ij}}{2}}{\sqrt{\sum_i \binom{n_{i.}}{2} \sum_j \binom{n_{.j}}{2}}} \tag{9}$$

$$AMI = \frac{I(U,V) - E\{I(U,V)\}}{max\{H(U), H(V)\} - E\{I(U,V)\}} \tag{10}$$

In Formula (10), $I(U,V) = \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{i,j}}{n} log \frac{\frac{n_{ij}}{n}}{\frac{n_{i.}n_{.j}}{n^2}}$, $H(U) = -\frac{i=1}{R} \frac{n_{i.}}{n} log \frac{n_{i.}}{n}$. The results are shown in the following table.

It can be seen from Tables 2–4 that the K-means algorithm performs the best on the Iris, Wine, and HAPT datasets, and ACBD can compete with K-means on the HAPT dataset. The DBSCAN algorithm did not obtain available clustering results after adjusting parameters many times on the WINE and HAPT datasets, so its clustering evaluation index was 0 on these two datasets. From the experimental results, the performance of the proposed algorithm (ACBD) on the large datasets, such as HAPT, is far better than that on the small datasets, such as IRIS and WINE.

Table 2. Comparison of IRIS experimental results.

	ARI	FMI	AMI
DBSCAN	0.4175	0.6723	0.5476
K-means	**0.7302**	**0.8208**	**0.7483**
ACBD	0.5681	0.7715	0.5768

Table 3. Comparison of Wine experimental results.

	ARI	FMI	AMI
DBSCAN	0	0	0
K-means	**0.3711**	**0.5835**	**0.4226**
ACBD	0.2451	0.4576	0.2708

Table 4. Comparison of HAPT experimental results.

	ARI	FMI	AMI
DBSCAN	0	0	0
K-means	**0.3878**	0.4713	**0.5524**
ACBD	0.3259	**0.5182**	0.3908

4.4. Neural Network Comparison and Overall Algorithm Effects

After ACBD clustering, user auxiliary annotation was carried out for each category to divide images into the above four categories, which greatly reduced the workload of manual annotation. The four labeled images were divided into training set and testing set according to 7:3, so as to train the deep neural network. The hyperparameters are set as follows: batch size, 32; epoch, 400; learning rate, 0.0001. Finally, the trained network is used to classify the video frames. The first category showing commodity details and the second category showing commodity overall effect are the video keyframes. Considering the deep network layer of ResNet, the accuracy is higher, but if the network layer is too deep, the machine is difficult to load. So, we verified whether MobileNet-v2 meets practical requirements while ensuring accuracy compared with ResNet-50. In the experiment, the batch size was set to 64.

As can be seen in Figures 10 and 11, the accuracy rates of ResNet-50 and MobileNet-V2 were similar, both at around 90%. Therefore, in consideration of applicability, we compared the training time and storage space of the two models.

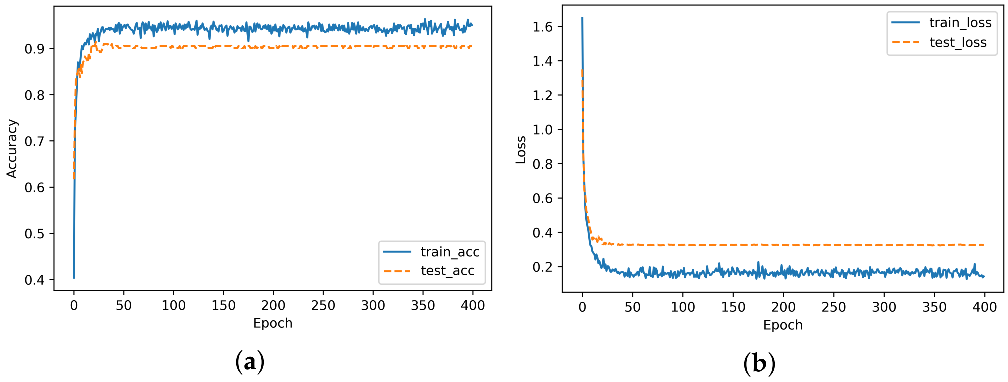


Figure 10. ResNet-50 training accuracy and loss (**a**,**b**).

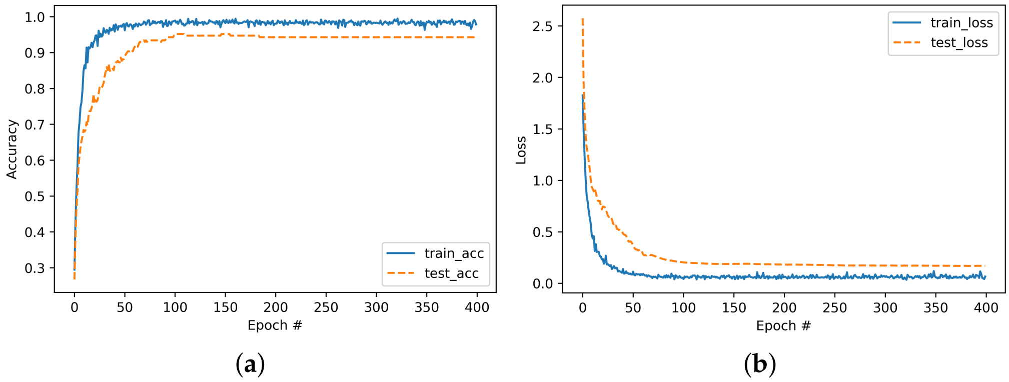


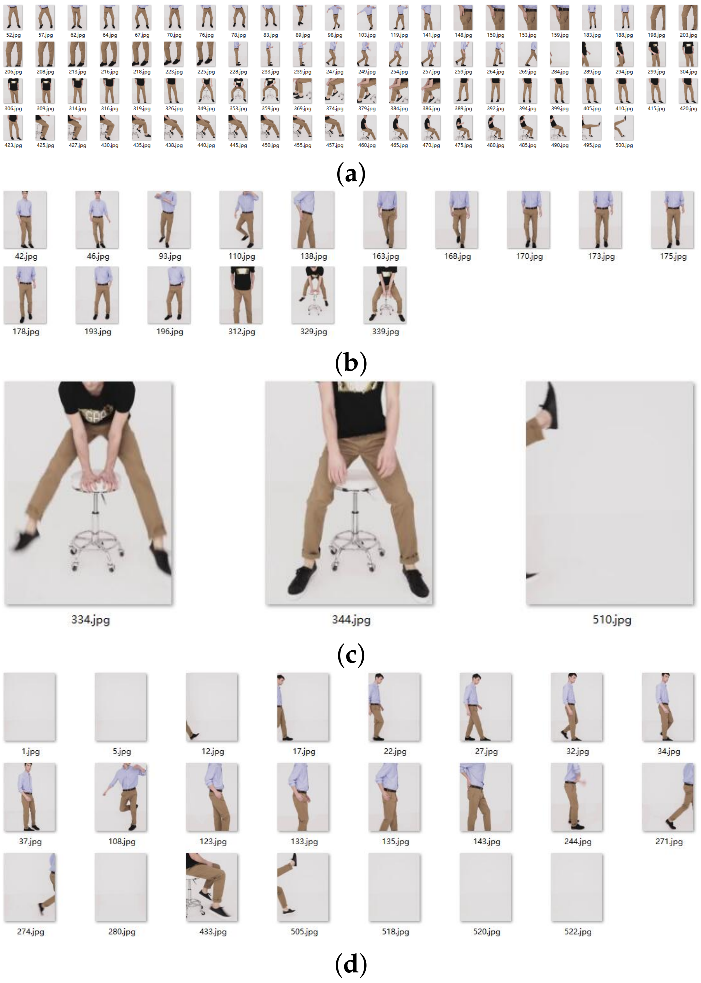
Figure 11. MobileNet-V2 training accuracy and loss (**a**,**b**).

Compared with ResNet-50, the training speed of MobileNet-v2 increased by 14.3% and the number of parameters decreased by 66.9%, shown in Table 5. Therefore, MobileNet-v2 has a wider available range from the applicability of mobile terminal, so we adopted the MobileNet-v2 network.

Table 5. Comparison of ResNet-50 and MobileNet-V2.

	ResNet-50	MobileNet-V2
Time spent per epoch (s)	7 ± 0.5	6 ± 0.05
Number of weights	42,472,325	14,064,709
Storage (MB)	324	107

Finally, we randomly selected one video from the remaining 108 videos in the database and inputted it into the MobileNet-V2 after training. Figure 12 shows the final classification results; (a), (b), (c), and (d) correspond to the first, second, third, and fourth categories, respectively, demonstrated in “Supplementary Materials”. The first and second categories are the keyframes defined in this paper. From the experimental results, it can be concluded that the algorithm proposed in this paper can achieve the purpose of extracting keyframes, user preferences are reflected in the auxiliary annotation stage, and the ACBD clustering algorithm can greatly improve the efficiency of annotation. Therefore, the overall framework of the algorithm in this paper is reasonable and has good results.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 12. Final result of algorithm. (**a**–**d**) correspond to the first, second, third, and fourth categories, respectively.

5. Discussion

Since different users have different preferences, the existing keyframe extraction algorithms do not provide solutions for different interests. For this reason, we propose an algorithm of commodity video keyframe extraction based on adaptive clustering annotation. Compared with the existing keyframe extraction algorithms, our algorithm reflects the preferences of different users through the process of user annotation, and achieves accurate and personalized keyframe extraction. It provides a feasible method for users to find products faster and more accurately.

6. Conclusions

In order to solve the problem that different users have different definitions of keyframes in commodity video, this paper proposes a keyframe recognition algorithm of commodity video that integrates transfer learning and interest information, and extracts the keyframes of different users’ preferences. First, the differential frames were extracted from commercial videos by frame-to-frame difference; then, the ACBD algorithm was used to cluster these frames to simplify the user annotation process. The data annotated by the user were used to train the Mobilenet-V2 network, and finally, the keyframes for individuals were extracted. The experimental results on the commodity video dataset constructed in this paper show that the proposed algorithm is effective and extracts the keyframes accurately. However, user preferences tend to change over time, which is not considered in this paper, so how to cater to the changing interests of users is the focus of our next research.

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/app12031502/s1, Video S1: Demonstration.

Author Contributions: Conceptualization, G.M. and X.S.; methodology, G.M.; software, G.M.; validation, G.M. and X.S.; writing—original draft preparation, G.M.; funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under grants No. 61876184.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Abbreviations

The following abbreviation is used in this manuscript:

ACBD	Adaptive Cluster Based on Distance

References

1. Araujo, A.; Girod, B. Large-Scale Video Retrieval Using Image Queries. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *28*, 1406–1420. [CrossRef]
2. Wang, S.; Cao, L.; Wang, Y. A Survey on Session-based Recommender Systems. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
3. Bi, F.; Lei, M.; Wang, Y.; Huang, D. Remote Sensing Target Tracking in UAV Aerial Video Based on Saliency Enhanced MDnet. *IEEE Access* **2019**, *7*, 76731–76740. [CrossRef]
4. Liu, X.; Song, M.; Zhang, L.; Wang, S.; Bu, J.; Chen, C.; Tao, D. Joint shot boundary detection and key frame extraction. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 2565–2568.
5. Liu, H.; Pan, L.; Meng, W. Key frame extraction from online video based on improved frame difference optimization. In Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 19–21 October 2012; pp. 940–944.
6. Gygli, M.; Grabner, H.; Riemenschneider, H.; Van Gool, L. Creating summaries from user videos. In *European Conference on Computer Vision*; Springer: Cham, Switzerland , 2014; pp. 505–520.
7. Santini, S. Who needs video summarization anyway? In Proceedings of the International Conference on Semantic Computing (ICSC 2007), Laguna Hills, CA, USA, 27–29 January 2007; pp. 177–184.
8. Lee, Y.J.; Ghosh, J.; Grauman, K. Discovering important people and objects for egocentric video summarization. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1346–1353.
9. Wu, J.X.; Zhong, S.H.; Jiang, J.M.; Yang, Y.Y. A novel clustering method for static video summarization. *Multimed. Tools Appl.* **2017**, *76*, 9625–9641. [CrossRef]
10. Gharbi, H.; Bahroun, S.; Massaoudi, M.; Zagrouba, E. Key frames extraction using graph modularity clustering for efficient video summarization. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 1502–1506.
11. Li, X.; Zhao, B.; Lu, X. Key Frame Extraction in the Summary Space. *IEEE Trans. Cybern.* **2017**, *48*, 1923–1934. [CrossRef]
12. Mademlis, I.; Tefas, A.; Pitas, I. Summarization of human activity videos using a salient dictionary. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 625–629.
13. Zhang, H.J.; Wu, J.H.; Zhong, D.; Smoliar, S.W. An integrated system for content-based video retrieval and browsing. *Pattern Recognit.* **1997**, *30*, 643–658. [CrossRef]
14. Yeung, M.M.; Liu, B. Efficient matching and clustering of video shots. In Proceedings of the IEEE International Conference on Image Processing, Washington, DC, USA, 23–26 October 1995; pp. 338–341.
15. Lai, J.L.; Yi, Y. Key frame extraction based on visual attention model. *J. Vis. Commun. Image Represent.* **2012**, *23*, 114–125. [CrossRef]

16. Cong, Y.; Yuan, J.; Luo, J. Towards Scalable Summarization of Consumer Videos Via Sparse Dictionary Selection. *IEEE Trans. Multimed.* **2012**, *14*, 66–75. [CrossRef]
17. Shih, H.C. A Novel Attention-Based Key-Frame Determination Method. *IEEE Trans. Broadcast.* **2013**, *59*, 556–562. [CrossRef]
18. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Int. Conf. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
19. Tomasz, H. Key Frames Detection in Motion Capture Recordings Using Machine Learning Approaches. *Int. Conf. Image Process. Commun.* **2016**, *525*, 79–86.
20. Tang, H.; Liu, H.; Xiao, W.; Sebe, N. Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion. *Neurocomputing* **2019**, *331*, 424–433. [CrossRef]
21. Pan, R.; Tian, Y.; Wang, Z. Key-frame Extraction Based on Clustering. In Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, Shanghai, China, 10–12 December 2010; Volume 2, pp. 867–871.
22. Sze, K.W.; Lam, K.M.; Qiu, G. A New Key Frame Representation for Video Segment Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2005**, *15*, 1148–1155.
23. Xiao, Y.; Xia, L. Key Frame Extraction Based on Connectivity Clustering. In Proceedings of the 2010 Second International Workshop on Education Technology and Computer Science, Wuhan, China, 6–7 March 2010.
24. Wang, Z.; Zhu, Y. Video Key Frame Monitoring Algorithm and Virtual Reality Display Based on Motion Vector. *IEEE Access* **2020**, *8*, 159027–159038. [CrossRef]
25. Chen, Y.; Huang, T.; Niu, Y.; Ke, X.; Lin, Y. Pose-Guided Spatial Alignment and Key Frame Selection for One-Shot Video-Based Person Re-Identification. *IEEE Access* **2019**, *7*, 78991–79004. [CrossRef]
26. Ren, J.; Jiang, J.; Feng, Y. Activity-driven content adaptation for effective video summarization. *J. Vis. Commun. Image Represent.* **2010**, *21*, 930–938. [CrossRef]
27. Mahmoud, K.M.; Ghanem, N.M.; Ismail, M.A. Unsupervised Video Summarization via Dynamic Modeling-based Hierarchical Clustering. In Proceedings of the 2013 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 4–7 December 2013; pp. 303–308.
28. Liu, D.; Hua, G.; Chen, T. A Hierarchical Visual Model for Video Object Summarization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2178–2190. [CrossRef]
29. Gygli, M.; Grabner, H.; Gool, L.V. Video summarization by learning submodular mixtures of objectives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3090–3098.
30. Zhao, M.; Guo, X.; Zhang, X. Key Frame Extraction of Assembly Process Based on Deep Learning. In Proceedings of the 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Tianjin, China, 19–23 July 2018; pp. 611–616.
31. Agyeman, R.; Muhammad, R.; Choi, G.S. Soccer Video Summarization using Deep Learning. In Proceedings of the 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 28–30 March 2019; pp. 270–273.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
34. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Rabinovich, A. Going deeper with convolutions. *IEEE Conf. Comput. Vis. Pattern Recognit.* **2015**, *1*, 1–9.
35. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
37. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
38. Gavai, N.R.; Jakhade, Y.A.; Tribhuvan, S.A.; Bhattad, R. MobileNets for Flower Classification using TensorFlow. In Proceedings of the 2017 International Conference on Big Data, IOT and Data Science (BID), Pune, India, 20–22 December 2017; pp. 154–158.
39. Shen, Y.; Sun, H.; Xu, X.; Zhou, J. Detection and Positioning of Surface Defects on Galvanized Sheet Based on Improved MobileNet v2. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8450–8454.
40. Fu, K.; Sun, L.; Kang, X.; Ren, F. Text Detection for Natural Scene based on MobileNet V2 and U-Net. In Proceedings of the 2019 IEEE international conference on mechatronics and automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 1560–1564.
41. Tian, Q.; Xie, G.; Wang, Y.; Zhang, Y. Pedestrian Detection Based on Laplace Operator Image Enhancement Algorithm and Faster R-CNN. In Proceedings of the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 13–15 October 2018; pp. 1–5.
42. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
43. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [CrossRef]
44. Ramirez, E.H.; Brena, R.; Magatti, D.; Stella, F. Topic model validation. *Neurocomputing* **2012**, *76*, 125–133. [CrossRef]
45. Vinh, N.X.; Epps, J.; Bailey, J. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* **2010**, *11*, 2837–2854.