Personalized Mobile Video Recommendation Based on User Preference Modeling by Deep Features and Social Tags

Jiafeng Li 1, Chenhao Li 1, Jihong Liu 1, Jing Zhang 1,*, Li Zhuo 1 and Meng Wang 2

1 Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China
2 School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
* Correspondence: zhj@bjut.edu.cn

Received: 5 August 2019; Accepted: 6 September 2019; Published: 13 September 2019

Abstract: With the explosive growth of mobile videos, helping users quickly and effectively find mobile videos of interest and further provide personalized recommendation services are the developing trends of mobile video applications. Mobile videos are characterized by their wide variety, single content, and short duration, and thus traditional personalized video recommendation methods cannot produce effective recommendation performance. Therefore, a personalized mobile video recommendation method is proposed based on user preference modeling by deep features and social tags. The main contribution of our work is three-fold: (1) deep features of mobile videos are extracted by an improved exponential linear units-3D convolutional neural network (ELU-3DCNN) for representing video content; (2) user preference is modeled by combining user preference for deep features with user preference for social tags that are respectively modeled by maximum likelihood estimation and exponential moving average method; (3) a personalized mobile video recommendation system based on user preference modeling is built after detecting key frames with a differential evolution optimization algorithm. Experiments on YouTube-8M dataset have shown that our method outperforms state-of-the-art methods in terms of both precision and recall of personalized mobile video recommendation.

Keywords: personalized mobile video recommendation; user preference modeling; deep features; social tags

1. Introduction

With the rapid development of the mobile Internet and multimedia technology, more and more users browse and watch videos through mobile terminals such as mobile phones and tablets. These videos in mobile terminals are called mobile videos, which are shared, transmitted, and accessed by the mobile network [1]. Due to the limitation of network traffic, mobile videos are characterized by their wide variety, single content, and short duration [2]. Until now, many applications (apps), such as Tik Tok, Kuaishou, Watermelon, etc., have become popular mobile video playback platforms and the main entertainment channels that consume people’s leisure time in life. For large-scale mobile video resources, how to help users quickly and effectively find mobile videos of interest to further provide personalized mobile video recommendation services has become the development trend of mobile video applications.

In real life, people tend to have different understandings and preferences for the same mobile video based on their cultural backgrounds, aesthetic standards, and environments. Therefore, a key issue
for personalized video recommendation is to obtain users’ interests [3]. The research of personalized recommendation has widely recognized the importance of user preference modeling [4], which aims to storage and manage user preferences for predicting a personalized interest by recording and learning user historical behaviors [5].

Mobile video are ultimately viewed and understood by people. Therefore, how to effectively represent mobile video content by extracting powerful descriptive features is the basic premise of user preference modeling for personalized recommendation. Fortunately, many researchers have made many outstanding works in the field of feature representation of mobile videos [6]. In addition, the mobile video platform generally allows users to add tags to the uploaded mobile videos. Effective tags can provide semantic clues of the video content, as well as reflect the user preferences [7]. As we know, tags and visual features can express the content of mobile videos from two different levels, that is to say, the information of them is often complementary. Therefore, it is a better way for user preference modeling by combining visual features with social tags. In our previous works, we proposed a personalized recommendation of social image methods, and the experimental results showed the effectiveness of constructing a user interest model with deep features and social tag trees [8]. Another key issue in personalized mobile video recommendation is to design an effective recommendation mechanism to help users preview the summary of video content and further make a decision. A general and widely validated strategy is to represent the summary description of a mobile video with key frames [9]. This is because mobile video key frames can be utilized to illustrate the main content of a mobile video by a single or few frames [10].

In recent years, various deep learning networks have been proposed, such as VGG (Visual Geometry Group) Net [11], GoogLeNet [12], ResNet [13], etc., which have been widely applied in the fields of image classification, image recognition, target detection, and semantic segmentation [14] and have achieved superior performance over traditional methods. This is because deep learning allows computational models that are composed of multiple processing layers to learn data representation with multiple levels of abstraction. Furthermore, deep learning discovers intricate structures in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer [15]. Considering the advantages of deep learning, a personalized mobile video recommendation method is proposed based on user preference modeling by deep features and social tags in this paper. Firstly, deep features of mobile videos are extracted by an improved exponential linear units-3D convolutional neural network (ELU-3DCNN) for representing the video content. Then, user preference is modeled by combining user preference for deep features with user preference for social tags that are respectively modeled by the maximum likelihood estimation (MLE) and exponential moving average (EMA) methods. Finally, a personalized mobile video recommendation system based on user preference modeling is built after detecting key frames with a differential evolution optimization algorithm.

The remainder of this paper is organized as follows. Section 2 reviews the related work on personalized mobile video recommendation. The overview of our personalized mobile video recommendation is provided in Section 3. Section 4 illustrates the feature representation of mobile videos based on ELU-3DCNN. Section 5 introduces the user preference modeling. Section 6 applies the obtained user preference to build a personalized mobile video recommendation system. Experimental results and analysis are given in Section 7. Conclusions are drawn in Section 8.

2. Related Work

Recent years have witnessed a considerable interest in personalized mobile video recommendation systems. Increasingly more advanced technologies can contribute to modeling user preference. We will introduce a brief literature review of personalized mobile video recommendation from different perspectives as mentioned above.
2.1. Feature Extraction

Feature representation is a compact way to represent the content of mobile videos. Personalized recommendation performance depends on the representational ability of a feature. The computer vision community has studied feature representation for decades, especially for videos. Some supervised learning schemes for extracting deep features have been developed. For example, Karpathy et al. [16] and Ji et al. [17] extended convolutional neural networks (CNNs) by performing convolutions in both time and space to make video content analysis. Simonyan et al. [18] proposed a two-stream ConvNet architecture that incorporates spatial and temporal networks trained on a multi-frame dense optical flow. However, because these kinds of methods do not consider the time information of mobile videos, deep features above cannot always achieve satisfactory results from mobile video classification. Unsupervised learning schemes for extracting deep feature have also been developed. For example, Srivastava et al. [19] and Donahue et al. [20] applied multilayer long short-term memory (LSTM) networks to extract features from mobile videos with an unsupervised way, in which LSTM as an encoder is used to map frame sequences into a fixed representation. Although these deep features perform well in video classification, their performance can be further improved by designing a dedicated neural network. Considering the structure of mobile videos, our previous work proposed a simple, yet effective approach for spatial-temporal feature learning using a 3D convolutional neural network called ELU-3DCNN, which takes a step forward by substituting exponential linear units (ELU) [21] for rectified linear units (ReLUs) to produce better results of feature representation.

2.2. User Preference Modeling

User preference modeling is the core module of personalized recommendation. Many researchers have proposed and explored user preference modeling methods based on some video elements, such as social tags, audio features, and visual features. For example, Zhu et al. [22] aimed to capture user preference by using a topic model to represent the video, and then generated recommendations by looking for those videos that fit to the topic video distribution of the user preference the most. Deldjoo et al. [23] proposed a new content-based recommender system, in which a set of stylistic features, such as lighting, color, and motion are extracted to represent video content. For an experienced domain expert, it is still hard to design a suitable feature extractor to transform the raw data into a feature vector, which can be used to represent the content of the mobile video accurately. Yoshida et al. [24] utilized the tags related to a video with model user preference, in which the semantic contents of videos can be well captured by analyzing tags associated with video content. Many researchers have proven that social tags related with video content can more efficiently describe video content and show user preference [25]. For example, we proposed a user interest tree constructed by deep features and tag trees for recommending social images in previous work [8]. Although it has been proven that combing visual features with social tags can improve recommendation accuracy, this method needs to be redesigned if applying to a recommended video because the video has an extra temporal stream compared with the static image. Moreover, social tag-based user modeling is difficult to represent the latent user preferences. It means that videos close to the current temporal period are usually more important than videos temporally far from the current period when user preference gradually changes over time [26]. However, none of these methods can deal with the relationship between user preference and time change very well. Considering the complementary of social tags and video features, user preference modeling is done by combining deep features with social tags in this paper.

2.3. Personalized Recommendation

An effective personalized recommendation mechanism can help users preview the summary of video content and further make a decision regarding what to watch. More recently, personalized recommendation methods have made great progress, including semantic-based recommendation [27–30], emotions-based recommendation [31], and multi-site recommendation [32].
A general and widely validated strategy is to represent the summary description of a mobile video with key frames. This is because mobile video key frames can be used to illustrate the main content of a mobile video by a single frame or a few frames. More recently, inspired by the deep learning breakthroughs in many fields, some researchers have tried to apply deep learning technology to the recommender system. Filho et al. [33] proposed a novel deep learning approach that represents items through the features extracted from key frames of movie trailers and then leveraged these features in a content-based recommender system. Wang et al. [34] generalized recent advances in deep learning and proposed a hierarchical Bayesian model called collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and collaborative filtering for the ratings matrix. Wang et al. [35] proposed a probabilistic formulation for a stacked denoising auto encoder (SDAE) and then extended it to a relational SDAE (RSDAE) model. A RSDAE synchronously performs deep representation learning and relational learning in a principled way under a probabilistic framework. Cheng et al. [36] proposed a wide and deep learning model to jointly train wide linear models and deep neural networks in order to combine the advantages of memorization and generalization for recommender systems. However, these kinds of methods do not realize the importance of recommendation mechanisms and cannot extract the key frame that represents the entire video. Therefore, the key frame will be used to make a summary of video content in this paper. Meanwhile, in view of the great advantages of deep learning, GoogLeNet is utilized to detect the key frame of a mobile video.

3. Overview of Our Method

The overview of our method is shown in Figure 1, which includes feature representation, user preference modeling, and personalized video recommendation. (1) Feature representation: a mobile video is firstly segmented into many video clips with several-frame overlap between two consecutive clips. These clips are passed to the ELU-3DCNN to extract the deep features. Then, these clip activations are averaged by an average pooling operation to form the deep features of a mobile video. (2) User preference modeling: firstly, deep feature representation and social tag representation of a mobile video is generated from a mobile video dataset and social tag dataset. Then, user preference modeling for deep features is created by MLE, and user preference modeling for social tags is constructed by EMA. Finally, user preference is modeled by combining user preference for deep features and user preference for social tags. (3) Personalized recommendation: the deep feature similarity and the social tag similarity between user preference and video in the dataset are computed in the procedure of personalized video recommending. Then, the union similarity based on the deep feature similarity and social tag similarity is calculated by the weight voting method, and the pre-recommended video is generated by union similarity. Next, the pre-recommended video is passed into GoogLeNet for the key frame level-deep feature. After that, a differential evolution algorithm is adopted to detect the key frame. Finally, these videos and their key frames are recommended to users who are interested in it.
The essence of mobile video features is a data representation of video content in feature dimension. A mobile video has many characteristics, such as wide variety, single content, and short duration, which makes it difficult to form an effective video representation if the visual features such as color, texture, and shape are used directly. Although there is many researches on video feature extraction, the traditional video features for video recommendation are always neither flexible nor satisfactory enough. We proposed a deep learning architecture called ELU-3DCNN to extract deep features for mobile videos in our previous work [21], in which ELU-3DCNN indeed processed a superior learning ability compared with ReLU-3DNN. In addition, the ELU-3DCNN not only outperforms ReLU-3DCNN on training accuracy and training loss, but also reaches a higher training efficiency. Meanwhile, the validation accuracy of ELU-3DCNN is also better than ReLU-3DCNN. Moreover, the final test accuracy of ELU-3DCNN is 84.5%, which outperforms the test accuracy of ReLU-3DCNN by 2.2%.

4. Feature Representation of Mobile Video

Currently, the most popular activation function for neural networks is the ReLU. The main advantage of ReLUs is that they can alleviate the vanishing gradient problem. However, ReLUs have a mean activation greater than zero due to the characteristic of being non-negative. Units with a non-zero mean activation act as bias for the next layer. From our previous work [21], a greater bias shift will bring the standard gradient far more to the natural gradient and further reduce the training speed. Similar to ReLUs, ELUs can also alleviate the vanishing gradient problem via identity and positive values. Moreover, because ELUs have negative values, the mean activation for every unit can be pushed to zero. Owing to the characteristic of being a no bias shift, the neural networks based on ELUs are more efficient and effective. An ELU is defined as

\[ f_{ELU} = \begin{cases} 
  x & x > 0 \\
  \alpha(\exp(x) - 1) & x \leq 0 
\end{cases} \]  

where \( \alpha \) is the ELU hyper parameter, which controls the convergence value when \( x \) tends to be negative infinite. In view of the effectiveness of ELUs, an ELU is adopted as the activation function in our architecture.
4.2. The Architecture of ELU-3DCNN

The proposed ELU-3DCNN structure is shown in Figure 2, which includes eight convolution layers, five max-pooling layers, six ELU activation functions, and two fully connected layers followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. The number of filters are denoted in each box. The 3D pooling layers are denoted from Pool 1 to Pool 5. All pooling kernels are $2 \times 2 \times 2$, except for Pool 1 ($1 \times 2 \times 2$). Each fully connected layer has 4096 output units. We used the existing YouTube-1M pre-trained ReLU-3DCNN model as our ELU-3DCNN initializations, then fine-tuned the ELU-3DCNN on an UCF-101 (University of Central Florida) dataset with 20 epochs. The initial learning rate is set to 10^{-3}, the batch size is 10, the momentum is 0.9, and the weight decay is 0.0005.

![Figure 2. The flowchart of deep feature extraction with an exponential linear units-3D convolutional neural network (ELU-3DCNN).](image)

4.3. Deep Feature Extraction of a Mobile Video with ELU-3DCNN

The detailed deep feature extraction process is illustrated in Figure 2. Firstly, a video is split into 16-frame long clips with 8-frame overlap between two consecutive clips. Then these clips are passed to the ELU-3DCNN to extract fc7 activations. Fc7 is the last fully connected layer. After that, these clip activations are averaged to form a 4096-dim deep mobile video feature. Finally, L2-normalization is used to normalize the deep feature. L2-normalization is defined as

$$y = x / \sqrt{\max \left( \sum_{i}^{N} x_i, \varepsilon \right)}$$

(2)

where $x$ and $y$ are the input and output of L2-normalization, respectively, $N$ is the dimension of input, and $\varepsilon$ is a very small number for avoiding the zero denominator.

5. User Preference Modeling

The traditional user preference modeling only relies on a single tag or visual feature of a mobile video to obtain user preference, which makes it difficult to effectively describe user behavior preferences. Therefore, we proposed a more feasible user preference modeling by visual features and social tags. In this section, user preference will be modeled by user preference for deep features and by user preference for social tags. Firstly, the user preference for deep features is modeled using the MLE method. Then, the user preference for social tags is modeled by EMA after the tag is sorted by tag popularity and frequency. Finally, the user preference is modeled based on the weighted voting method.
5.1. User Preference Modeling for Deep Features

User preference modeling for deep features is mainly based on normal assumptions and maximum likelihood estimation. As shown in Figure 3, the representation of each mobile video in a feature space can be regarded as sampling under the normal assumption of user preference modeling for deep features, which can be obtained through MLE.

![Diagram](image)

**Figure 3.** The flowchart of user preference modeling for deep features.

5.1.1. Normal Assumption

User preference means that the user likes to see the kind of mobile video, in which the user’s viewing log is a concrete embodiment of the user preference. If the user preference modeling is regarded as a probabilistic model, then the user may prefer mobile videos that have a higher probability to be sampled from this probabilistic model.

If \( H = (v_1, v_2, \ldots, v_M) \) denotes the video set containing \( M \) videos watched by a user, the representation of each mobile video in a feature space can be regarded as sampling in the user preference modeling for deep features. For the \( i \)-th user \( u^{(i)} \), assume that each dimension in a feature space obeys a normal distribution and the deep features are linearly independent in each dimension

\[
u_j^{(i)} \sim N(a_j, \sigma_j^2)
\]

where \( a_j \) and \( \sigma_j^2 \) denote the mean and variance of the \( j \)-th dimension of the user preference modeling for deep features, respectively, and \( u_j^{(i)} \) denotes the values of the user preference modeling of the \( j \)-th dimension of the \( i \)-th user.

5.1.2. Maximum Likelihood Estimation

For a model that obeys a normal distribution, the model parameters can be obtained by maximum likelihood estimation. Therefore, the parameters of user preference modeling for deep features obtain the mean value and standard deviation by maximum likelihood estimation. The parameters \( a_j \) and \( \sigma_j^2 \) of user preference modeling for deep features are \( u_j^{(i)} \). The \( P(H|a_j, \sigma_j^2) \) function represents the maximum likelihood calculation

\[
P(H|a_j, \sigma_j^2) = \prod_{v \in H} P(v|a_j, \sigma_j^2)
\]

Intuitively, MLE is an attempt to find a value that maximizes the likelihood of data appearance in all possible values. The above equation is further optimized as

\[
LL(a_j, \sigma_j^2) = \sum_{v \in H} P(v|a_j, \sigma_j^2)
\]

Finally, the maximum likelihood estimates for parameters \( a_j \) and \( \sigma_j^2 \) are

\[
\hat{a}_j = \frac{1}{M} \sum_{i=1}^{M} P(v_j^{(i)})
\]

\[
\hat{\sigma}_j^2 = \frac{1}{M} \sum_{i=1}^{M} \left( v_j^{(i)} - \hat{a}_j \right)^2
\]
After completing the maximum likelihood estimates $\hat{a}_j$ and $\hat{\sigma}_j^2$, the user preference modeling for deep features can be represented as

$$U_d = P(u_d | \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_n; \hat{\sigma}_1^2, \hat{\sigma}_2^2, \ldots, \hat{\sigma}_n^2)$$

(8)

where $U_d$ represents the user preference for deep features and $u_d$ represents the independent variable of user preference modeling. The $P$ function is used to calculate the probability density and $n$ represents the dimension of the parameter.

5.2. User Preference Modeling for Social Tags

In applications of mobile videos, tagging activity has been recently identified as a potential source of knowledge about personal interests, preferences, goals, and other attributes known from user models. Thus, social tags can provide semantic clues of video content to reflect the user’s preferences. Here, user preference modeling for social tags will be constructed in this section.

5.2.1. Social Tag Representation of Video

It is a fact that users are reluctant to label a video in mobile video recommendations because labeling a mobile video is time-consuming work, especially for those users who only want to watch the mobile videos for a few minutes. Meanwhile, tags labeled by the user always are invalid even when not related with the mobile video. We tried to validate our statements by collecting 100 mobile videos from the Watermelon App to analyze the relationship between video duration and number of comments. The results are demonstrated in Figure 4.

![Figure 4. The relationship between time and number of comments.](image)

From Figure 4, we can see that the number of comments reduces with the drop of the duration of videos. Therefore, it is impossible to expect users to label a tag for mobile videos. An available way of removing invalid tags is to recommend mobile videos based on the tags that have been processed and standardized. Therefore, the first thing that should be done for a video recommender system is to decide which words can be used as tags and then make a word dictionary. Here, we choose to represent the video labeled by experts with a vector.

Firstly, the popularity of a social tag is calculated, which is an estimate of how common a social tag is in general usage. For instance, the social tag “animal” in mobile videos is very common, and social tag frequency tends to incorrectly emphasize videos with the more frequent social tag “animal” instead of more meaningful tags such as “dog” or “cow”. The social tag “animal” is not a good keyword for distinguishing relevant and non-relevant videos and terms. Obviously, the less-common words “dog” and “cow” are suitable selections. As a result, the social tag popularity factor is incorporated to diminish the weight of social tags that occur very frequently in the video set and increase the weight of social tags that occur rarely. The social tag popularity $p$ can be acquired by

$$p = \log\left(\frac{N}{1 + ||v \in V : t \in v||}\right)$$

(9)
where $N$ is the total number of videos in the database $V$; and $|\{v \in V : t \in v\}|$ is the number of videos where the social tag $t$ appears. Finally, a video $v$ with social tag popularity can be defined as:

$$v = (w_1, w_2, \ldots, w_i, \ldots, w_{\|D\|})$$

(10)

$$w_i = f(t_i, v) \times p_i$$

(11)

where $\|D\|$ is the size of the social tag dictionary, $f(.)$ is equal to 1 when video $v$ has the social tag $t_i$, and the social tag weight in video $v$ is the product of function $f(.)$ and popularity $p$.

5.2.2. User Preference Modeling for Social Tags with EMA

User preference modeling for social tags is built based on two intuitions: (1) The frequency of a social tag represents the strength of user preference on this category defined by this social tag. A social tag with a higher frequency implies that the user might prefer those videos labeled with this social tag. In addition, a social tag with a lower frequency represents that the user might not be interested in this social tag. (2) Social tag data close to the current temporal period are usually more important than data temporally far from the current period. For example, a certain user was interested in a personal digital assistant (PDA) six months ago. He is currently interested in the iPad, and the social tag of an iPad is used frequently. It is more appropriate to recommend an iPad over a PDA to this user.

If only the social tag frequency is considered, the influences of time will be ignored in user preference modeling for social tags. Therefore, EMA is utilized to calculate the social tag frequency, which is a weighted method that can make full use of time information. Here, the weight of the social tag that is closer to the current time will be higher than the social tag that is farther from the present moment [37]. Suppose the social tag representation of a video is $t_v$, the user preference model for the social tag is $U_0$ before it is updated, and $U_t$ is the user preference model for the social tag after it is updated. Their relationship is illustrated as

$$U_t = a t_v + (1 - a) U_0$$

(12)

where $a$ is an updating parameter of a user who has watched $m$ videos and $n$ is the number of videos with the social tag. The above equation can be rewritten as

$$U_t = \frac{t_{v_1} + (1 - a)t_{v_2} + (1 - a)^2 t_{v_3} + \ldots + (1 - a)^n t_{v_n}}{1 + (1 - a) + (1 - a)^2 + \ldots + (1 - a)^m}$$

(13)

With the increase of viewed videos, the denominator of Equation (12) is equal to $1/a$. Therefore, for a user who has watched $m$ videos, his or her user preference for a social tag can be modeled by

$$U_t = a \left( t_{v_1}^1 + (1 - a)t_{v_2}^2 + (1 - a)^2 t_{v_3}^3 + \ldots + (1 - a)^n t_{v_n}^n \right)$$

(14)

5.3. User Preference Modeling by Deep Features and Social Tags

In order to combine the advantages of deep features with social tags, user preference is jointly modeled with deep features and social tags. Set $U_d$ represents the user preference model for deep features and $U_t$ represents the user preference model for social tags. A user preference model can be formalized as

$$U = a U_t + (1 - a) U_d$$

(15)

where $a$ is the parameter to balance the weight of user preference modeling for social tags and user preference modeling for deep features. Since it is hard to have an optimal $a$ mathematically, an experiment is implemented to decide the value of $a$. The detail of this experiment will be shown in Section 7.
6. Personalized Recommendation System

After modeling user preference, a personalized mobile video recommendation system is built based on user preference modeling. Firstly, user preference is modeled with user preference for deep features and user preference for social tags. Then, the top $N$ videos closer to the user preference are scheduled to detect the key frame. Finally, these videos and their key frames are recommended to the user who may be interested in them.

6.1. Personalized Mobile Video Recommendation Based on User Preference Modeling

Based on the user preference with deep features and social tags, mobile videos that users may be interested in will be recommended to users. The deep feature similarity and social tag similarity are calculated based on user preference. Both deep feature similarity and social tag similarity is combined to construct joint similarity. This joint similarity has a better effect on the similarity measurement of mobile videos. Finally, the $N$ mobile videos are recommended with higher similarity to users.

Similarity matching aims to determine whether the mobile video meets the user preference. When the content of some mobile videos is similar to the user preference, the user may be more interested in these mobile videos. Similarity matching criterion of two preference models is shown in Equation (16) and Equation (17)

\[
S_d = \frac{1}{M} \sqrt{\langle U_d \rangle^T \tilde{v}_d}
\]

\[
S_t = \frac{1}{N} \sqrt{\langle U_t \rangle^T \tilde{v}_t}
\]

where $S_d$ and $S_t$ are the similarities between video $v$ and the user preference model $U_d$ for deep features, and the user preference model $U_t$ for social tag, respectively. $\tilde{v}_d$ and $\tilde{v}_t$ respectively represent the mobile video under the deep feature representation and social tag representation. $N$ and $M$ represent the number of dimensions.

After matching the similarity between the mobile video and user preference for deep features and the similarity between mobile video and the user preference for social tags, the joint similarity $S$ is calculated as

\[
S = \lambda S_t + (1 - \lambda) S_d
\]

where $\lambda$ is the weight of the user preference model for social tags, which has the same value as $\alpha$ in Equation (15).

6.2. Key Frame Detection with Differential Evolution

One significant issue in mobile video recommendation is how to present these videos in sight of users more efficiently. A widely investigated and efficient scheme treats a video as a composition of key frames. Therefore, key frames can be utilized to represent video content and displayed to users in the process of recommendation. Unfortunately, due to cubical computational complexity, most methods such as histogram-based methods, entropy analysis, and correlation of images are time-consuming. In this paper, GoogLeNet is firstly used to extract deep features for each frame offline, as discussed in Section 6.2.1. Then, the differential evolution optimization algorithm is used to detect key frames. The flowchart of key frame detection is illustrated in Figure 5.
6.2.1. Problem Definition of Key Frame

There are always two steps for a general key frame detection process (i.e., the shot boundary detection of the video and key frames extraction for each shot). However, because the vast majority of mobile videos are short and have only one scene, the shot boundary detection is redundant for mobile videos. Therefore, we will directly detect the key frames for each video in the database. Suppose a mobile video contains \( m \) frames. Let \( f_i \) represent the \( i \)-th frame in the videos. Intuitively, when all the frames of a video are divided into \( k \) clusters to maximize intra-cluster similarity and minimize inter-class similarity, the frame closer to the center of the cluster is the key frame. Therefore, we let \( u_j \) represent the center of the \( j \)-th cluster. Formally, the key frame is determined by minimizing the loss function as

\[
E = \sum_{j=1}^{k} \sum_{x \in c_j} ||x - u||^2
\]  

(19)

In general dimension for \( k \geq 2 \), minimizing Equation (19) is often referred to as a NP(Non-deterministic Polynomial)-hard problem in [38]. Therefore, we used a heuristic algorithm differential evolution algorithm to iteratively approximate the result which is discussed in Section 6.2.3.

6.2.2. Extracting Deep Features of Key Frame with GoogLeNet

According to the analysis above, deep learning based key frame detection method is superior to other methods such as histogram-based methods, entropy analysis, and correlation of images. In addition, it can be accelerated by GPU and achieves the purpose of dimensionality reduction. Therefore, GoogLeNet is used to extract deep features of key frame to reduce the algorithm complexity.

GoogLeNet is built by stacking convolutional building blocks (shown in Figure 6) on top of each other [11]. The main idea of the inception architecture is to find out how an optimal, locally sparse structure in the convolutional visual network can be approximated and covered by dense components that are readily available. GoogLeNet stacked the inception models to make a deeper and wider neuron network than AlexNet. In our work, GoogLeNet is only used as a feature extractor. We analyzed all the frames on disk and calculated the bottleneck values for each of them. ‘Bottleneck’ is an informal term, which is often used for the layer just before final output layer that actually does the classification. This penultimate layer has been trained to output a set of values that is good enough for the classifier to distinguish all the classes that needed to be recognized. That means it has to be a meaningful and compact summary of the frames and can be used as a representation of frames to extract key frames, since it has to contain enough information to represent the content of the frames in a very small set of values.
In the past decade, evolutionary algorithms have been extensively used in various problem domains and have successfully and effectively found a solution that is nearly optimal [39]. In this paper, differential evolution algorithm [40] is utilized to detect the key frame. As shown in Figure 7, it performs iterative execution of mutation-crossover-selection process cycle over the set of candidates in the present population.

```
1 Population Initialization: 
2 IP = {C1, C2, C3...CPS} 
3 Compute the Fitness of the Candidates: 
4 Σi(Fitness(Ci)) 
5 Generate child for each parent in IP: 
6 For (Generation (G) = 1 to Maximum_Generation(MaxGen)) do 
7 For (i = 1 To PS) for every Ci do 
8 Mutation process 
9 Select Parents for Mutation 
10 Find the Difference Vector 
11 Find the Mutant Vector (Mi) 
12 Crossover Process 
13 Perform Crossover between Ci and Mi to
14 Generate Trial Vector (Ti) 
15 Selection Process 
16 Ci for G + 1 = Fittest_Among (Ci, Ti) 
17 End For 
18 End For
```

Figure 7. Algorithmic structure of classical differential evolution algorithm.

The differential evolution algorithm starts its search by a population initialization that randomly initializes a set of problems dependent on candidate solutions. Then, the fitness of the candidates is computed by the problem-specific objective function. After that, the child of each parent in the initial population is generated. The generation process included three steps—mutation process, crossover process, and selection process—will be executed iteratively until the user-defined stop criteria is met. Thus, at the end of each generation, a new group of survivors will be generated. The framework is designed in such a way as to generate 10 key frames for each video. Parameters used in differential evolution algorithm along with their values are described below.

In differential evolution algorithm, there are many hyper parameters to be determined before the execution. The size of each vector D that decides the number of key frames required for the video processing application is set to 10 constantly in our experiment. The population size NP decides the number of candidates in the population. In our experiment, NP is set to 10. The scaling factor F is the step size used for differential mutation. In our experiment, F is set to 0.9. The crossover weight Cr...
is the crossover probability used in the crossover operation to determine whether the components of the trial vectors are from the parent vector or mutant vector. In our experiment, \( Cr \) is set to 0.6. The maximum number of generations \( \text{MaxGen} \) is set to 40, for each run. The population is a set of \( NP \) candidate vectors with dimension of \( D \). The initial population is initialized by randomly selecting some frames in the video as the key frames. The average Euclidean distance is used as the fitness function that evaluates the quality of each candidate solution in the population.

Figure 8 shows part of the detection results of several mobile videos, in which five frames are detected at equal intervals from all video frames and it displays them in the first column of the image. The second row is the detected key frame; it can be seen that the key frame can fully summarize the video content so as to help the user previewing the video content in general.

![Figure 8. The detection results of the key frame of several mobile videos. (a) The detection result of video 1; (b) the detection result of video 2; (c) the detection result of video 3; (d) the detection result of video 4.](image)

6.3. The GUI (Graphical User Interface) Design of Personalized Mobile Video Recommendation System

The GUI for the proposed personalized mobile video recommendation system is shown in Figure 9. The GUI is mainly based on a Windows image interface, and the interaction mode is mainly a mouse and keyboard. After completing the registration, the user can enter his or her username and password through the keyboard, and then click the mouse to select the mobile video to enter the playback interface to complete the video playback. The design of the whole interactive interface is based on JavaFX (Sun, Santa Clara, California, USA) open source framework, and video playback is mainly based on VLC (Video LAN Client) core. The left image is the “Log in” and “Sign up” screens. The middle one is the screen of recommendation list. The right is the playing interface.

![Figure 9. The GUI for personalized mobile video recommendation system. (Left) The “Log in” and “Sign up” screens; (middle) screen of the recommendation list; (right) playing interface.](image)

7. Experiments Results and Analysis

In this section, a quantitative study and comparative analysis are conducted to demonstrate the effectiveness of our recommendation system in this paper. In Experiment I, the results between the
original key frame extraction algorithm and the deep features-based key frame extraction algorithm are compared. Then, an experiment is implemented to decide the value lambda in Experiment II. In Experiment III, the correlation coefficient results between the predictive user preference and the real user preference are illustrated. The precision and recall of the tag ranking method, video topic method, and our method are compared for personalized mobile video recommendation in Experiment IV.

7.1. Experimental Dataset and Setting

In order to evaluate the performance of our personalized recommendation, experiments were conducted on a dataset sampled from YouTube-8M dataset. YouTube-8M is a large-scale labeled video dataset that consists of millions of YouTube video IDs (Identity Documents) and associated social tags from a diverse vocabulary of 4700+ visual entities. Since each video in our dataset is between 120 and 500 seconds long, this dataset is available for our mobile video recommendation experiments.

The deep feature extraction and key frame detection are implemented under the Ubuntu system using TensorFlow platform (Google, Mountain view, California, USA) and Python programming language. The remaining experiments are completed in a PC with 3.60 GHz CPU, 4.00 G memory and windows 7 operating system, IntelliJ IDEA programming, and SQL Lite database. It is worth mentioning that IntelliJ IDEA is an integrated environment for Java programming language development. We used Java as the main technical framework in Windows.

To verify the superiority of the proposed method, we compared with other user preference modeling methods.

- Video topic method [22]: this method represents user preference modeling by a topic model that has a closer distribution with the user’s history.
- Tag ranking method [24]: this method builds user preference modeling by ranking all the social tags associated with all videos that the user has watched.

7.2. Experiment I: Comparison of Operation Efficiency of Key Frame Detection

In order to speed up computation and achieve real-time recommendations in a mobile environment, the differential evolution optimization algorithm was used with deep frame feature to detect key frames. Therefore, in this section, an experiment was implemented to compare the original key frame detection and deep features-based key frame detection. The relationship between iterations and distance is illustrated in Figure 10. The running time for extracting results per video is shown in Table 1. It can be seen from the experiment that the distance of both methods will increase with iterations because the differential evolution optimization algorithm tries to find a solution satisfying Equation (19) in the whole solution space. Meanwhile, the running time per video proves that deep learning-based method can speed up the process of key frame detection. Therefore, it is applicable that differential evolution algorithm is used to detect the key frame.

![Figure 10. The relationship between iterations and distance. (a) The original method; (b) our method.](image-url)
7.3. Experiment II: Parameter Analysis of User Preference Modeling

As mentioned above, we jointly modeled user social tag preference and user deep feature preference into one model to combine the advantages of social tags with deep features. However, it is difficult for the weight lambda of user preference modeling for social tags to have an optimal value mathematically, therefore an experiment is implemented to decide the value of lambda. Different lambda such as 0.05, 0.1, 0.2, 0.4, 0.6, and 0.8 are used to recommend 10 videos to each of the 10 users with different backgrounds. Then, their interested videos are recorded to analyze which lambda can get the best recommended performance. The final results are shown in Figure 11. From the experimental results we can see that the interested videos of the recommendation results are optimal when the lambda is equal to 0.35 approximately. Therefore, lambda is set to 0.35 in our method.

![Figure 11. The relationship between the lambda and the interested videos.](image)

7.4. Experiment III: Correlation Coefficient Comparison Results between the Predictive User Preference and the Real User Preference

In the field of user preference modeling, the Pearson correlation coefficient between the predictive user preference and real user preference is often used to verify the superiority of the user preference modeling method. The Pearson correlation coefficient \( r \) is defined as

\[
r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\sum X^2 - \frac{(\sum X^2)^2}{N}} \sqrt{\sum Y^2 - \frac{(\sum Y^2)^2}{N}}}
\]  

(20)

where \( N \) is the number of mobile video categories, \( X \) is the user real preferences for each category, and \( Y \) is the predictive user preferences for each category.

Meanwhile, in order to ensure the fairness and feasibility of the experiment, 10 users with different backgrounds were tracked in the experiment for one month. Their preferences were modeled based on their viewing behavior. Then the total number of videos viewed by each user was counted. The users’ Pearson correlation coefficients for this category are shown in Table 2 and Figure 12. It can be seen that the correlation coefficient of our method between the predictive user preference and the real user preference is the highest. The average Pearson correlation coefficient of 10 users is 0.836 in our method, which is much higher than the 0.621 of the video topic methods and is lightly higher than 0.803 of the tag ranking method. At the same time, the superiority of our method can be seen more intuitively from Figure 12. This is because our method fully realizes the complementarity of users’ different levels of information and can effectively describe user preference.

| Methods                  | Differential Evolution Algorithm | Our Method |
|-------------------------|---------------------------------|------------|
| Time                    | 10 s                            | 5 s        |

Table 1. The running time results of two methods.
Table 2. The comparison of correlation coefficient of three methods.

| User  | Video Topic [22] | Tag Ranking [24] | Our Method |
|-------|------------------|------------------|------------|
| User 1| 0.64             | 0.82             | 0.82       |
| User 2| 0.55             | 0.79             | 0.86       |
| User 3| 0.60             | 0.88             | 0.91       |
| User 4| 0.45             | 0.69             | 0.66       |
| User 5| 0.67             | 0.69             | 0.76       |
| User 6| 0.56             | 0.93             | 0.98       |
| User 7| 0.74             | 0.86             | 0.88       |
| User 8| 0.78             | 0.79             | 0.88       |
| User 9| 0.58             | 0.85             | 0.87       |
| User 10| 0.64            | 0.73             | 0.74       |
| Average| 0.621           | 0.803            | 0.836      |

7.5. Experiment IV: Comparison Results on Personalized Video Recommendation

In order to analyze these algorithms in fairness, 10 different background users’ video browsing history was tracked for a month to model their user preferences. Then, tag ranking, topic video, and our method were used to rank 50 videos, respectively. Finally, in order to calculate precision and recall of each method, the information about the user preference for the 50 videos was collected.

The results of the final recall and precision are shown in Table 3. Taking k = 20 as an example, the precision of the video topic method [22], the tag ranking method [24], and our method are 0.51, 0.59, and 0.65, respectively. Meanwhile, the recall of the three methods are 0.55, 0.64, and 0.70, respectively. Therefore, it can be seen that the recall and precision of our method are better than the other two methods.

The data in Table 3 is plotted as a graph shown in Figure 13. The green dashed line is the recall and precision curve of our method, the blue is the video topic method, and the yellow line is the tag ranking method. From the experimental results, our method has the highest precision and recall compared to the others. The reason is that our method models high-level semantic information and visual information simultaneously. The user preference modeling for social tags can capture the semantic information of user preference, while the user preference modeling for deep features can capture the latent representation of user preference.
Table 3. The precision and recall of three methods.

| Number of Video | Video Topic [22] | Tag Ranking [24] | Our Method |
|-----------------|------------------|------------------|------------|
|                 | Precision        | Recall           | Precision | Recall |
| k = 1           | 0.75             | 0.03             | 1         | 0.05   |
| k = 5           | 0.70             | 0.02             | 0.89      | 0.27   |
| k = 10          | 0.61             | 0.34             | 0.86      | 0.48   |
| k = 15          | 0.57             | 0.46             | 0.75      | 0.61   |
| k = 20          | 0.51             | 0.55             | 0.59      | 0.64   |
| k = 25          | 0.45             | 0.60             | 0.50      | 0.68   |
| k = 30          | 0.41             | 0.66             | 0.43      | 0.70   |
| k = 35          | 0.39             | 0.72             | 0.39      | 0.72   |
| k = 40          | 0.36             | 0.76             | 0.36      | 0.76   |
| k = 45          | 0.33             | 0.79             | 0.33      | 0.80   |
| k = 50          | 0.31             | 0.80             | 0.31      | 0.80   |

Figure 13. The precision and recall curve of three methods.

8. Conclusions and Future Work

A personalized mobile video recommendation method is proposed based on user preference modeling by deep features and social tags in this paper. Firstly, in order to compactly and accurately represent the content of mobile videos, the deep features are extracted by ELU-3DCNN. Then, user preference is modeled by maximum likelihood estimation and exponential moving average method to have a trade-off between user preference for deep features and user preference for social tags. Finally, a personalized mobile video recommendation system is built based on user preference modeling after detecting key frames with differential evolution optimization algorithm. The experiments are performed in precision and recall, proving the effectiveness of our method. Experimental results have shown our method can significantly improve the precision and recall of recommendation results.

In further works, the following aspects will be considered: a) a more suitable deep neural network will be used to extract deep features; b) the natural language processing algorithm can be utilized to solve the problems of understanding social tags; c) more information such as rating matrix and user comments will be considered to generate a more accurate user preference modeling. Through designing a better user preference modeling to represent the user interest, a more accurate recommendation system will be built in future works.

Author Contributions: J.Z. and J.-F.L. proposed the research direction and gave the conceptualization. J.-F.L., C.-H.L. and J.-H.L. conceived and designed the experiments, analyzed and interpreted the data. J.-F.L., C.-H.L., J.-H.L. and J.Z. wrote the paper. L.Z. and M.W. supervised the study and reviewed this paper. All authors read and approved the final manuscript.
Funding: This work in this paper was supported by the Beijing Municipal Natural Science Foundation Cooperation Beijing Education Committee (No. KZ 201910005007, No. KZ 201810005002) and the National Natural Science Foundation of China (No. 61971016, No. 61531006, No.61602018, and No.61701011).

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

References
1. Jiang, L.; Fu, X. Research and implementation of algorithm for short videos recommendation. In Proceedings of the IEEE International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Beijing, China, 17–20 August 2014; pp. 796–801.
2. Jones, Q.; Ravid, G.; Rafaeli, S. Information overload and the message dynamics of online Interaction spaces: A theoretical model and empirical exploration. *Inf. Syst. Res.* 2004, 15, 194–210. [CrossRef]
3. Qian, X.; Feng, H.; Zhao, G.; Mei, T. Personalized recommendation combining user interest and social circle. *IEEE T Knowl. Data. Eng.* 2014, 26, 1763–1777. [CrossRef]
4. Min, W.; Bao, B.; Xu, C.; Hossain, M.S. Cross-platform multi-modal topic modeling for personalized inter-platform recommendation. *IEEE T Multimed.* 2015, 17, 1787–1801. [CrossRef]
5. Zhang, J.; Yang, Y.; Tian, Q.; Zhuo, L.; Liu, X. Personalized social image recommendation method based on User-Image-Tag model. *IEEE T Multimed.* 2017, 19, 2439–2449. [CrossRef]
6. Cheng, G.; Wan, Y.; Saudagar, A.N.; Namuduri, K.; Buckles, B.P. Advances in human action recognition: A survey. *arXiv* 2016, arXiv:1501.05964.
7. Zhang, Z.; Zhou, T.; Zhang, Y. Tag-aware recommender systems: A state-of-the-art survey. *J. Comput. Sci. Technol.* 2011, 26, 767–777. [CrossRef]
8. Zhang, J.; Yang, Y.; Zhuo, L.; Tian, Q.; Liang, X. Personalized recommendation of social images by constructing a user interest tree with deep features and tag trees. *IEEE T Multimed.* 2019. [CrossRef]
9. Wang, X.; Weng, Z. Scene abrupt change detection. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Halifax, NS, Canada, 7–10 May 2000; pp. 880–883.
10. Liu, G.; Wen, X.; Zheng, W.; He, P. Shot boundary detection and key frame extraction based on scale invariant feature transform. In Proceedings of the International Conference on Computer and Information Science, Shanghai, China, 1–3 June 2009; pp. 1126–1130.
11. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* 2014, arXiv:1409.1556.
12. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 111–116.
13. 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.
14. Girshick, R.B.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE T Pattern Anal.* 2016, 1, 142–158. [CrossRef]
15. Lecun, Y.; Bengio, Y.; Hinton, G.E. Deep learning. *Nature* 2015, 521, 436–444. [CrossRef]
16. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Feifei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
17. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE T Pattern Anal.* 2013, 35, 221–231. [CrossRef] [PubMed]
18. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. *arXiv* 2014, arXiv:1406.2199.
19. Srivastava, N.; Mansimov, E.; Salakhudinov, R. Unsupervised learning of video representations using LSTMs. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 843–852.
20. Donahue, J.; Hendricks, L.A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Darrell, T.; Saenko, K. Long-term recurrent convolutional networks for visual recognition and description. *IEEE T Pattern Anal.* 2017, *39*, 677–691. [CrossRef] [PubMed]

21. Liu, J.; Zhang, J.; Zhang, H.; Liang, X.; Zhuo, L. Extracting deep video feature for mobile video classification with ELU-3DCNN. In Proceedings of the International Conference on Internet Multimedia Computing and Service, Qingdao, China, 23–25 August 2017; pp. 151–159.

22. Zhu, Q.; Shyu, M.; Wang, H. VideoTopic: Content-based video recommendation using a topic model. In Proceedings of the International Symposium on Multimedia, Anaheim, CA, USA, 9–11 December 2013; pp. 219–222.

23. Deldjoo, Y.; Elahi, M.; Cremonesi, P.; Garzotto, F.; Piazzolla, P.; Quadrana, M. Content-based video recommendation system based on stylistic visual features. *Lect. Notes Comput. Sci.* 2016, *5*, 99–113. [CrossRef]

24. Yoshida, T.; Irie, G.; Satou, T.; Kojima, A.; Higashino, S. Improving item recommendation based on social tag ranking. In Proceedings of the Conference on Multimedia Modeling, Klagenfurt, Austria, 4–6 January 2012; pp. 161–172.

25. Guy, I.; Zwerdling, N.; Ronen, I.; Carmel, D.; Uziel, E. Social media recommendation based on people and tags. In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 194–201.

26. Li, L.; Zheng, L.; Yang, F.; Li, T. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Syst. Appl.* 2014, *41*, 3168–3177. [CrossRef]

27. Saia, R.; Boratto, L.; Carta, S. A semantic approach to remove incoherent items from a user profile and improve the accuracy of a recommender system. *J. Intell. Inf. Syst.* 2016, *47*, 111–134. [CrossRef]

28. Said, A.; Bellolli, A. Coherence and inconsistencies in rating behavior: Estimating the magic barrier of recommender systems. *User Model. User Adapt. Interact.* 2018, *28*, 97–125. [CrossRef]

29. Saia, R.; Boratto, L.; Carta, S. Semantic Coherence-based User Profile Modeling in the Recommender Systems Context. In Proceedings of the 6th International Conference on Knowledge Discovery and Information Retrieval (KDIR), Rome, Italy, 22–24 October 2014; pp. 154–161.

30. Saia, R.; Boratto, L.; Carta, S.; Fenu, G. Binary sieves: Toward a semantic approach to user segmentation for behavioral targeting. *Future. Gener. Comput. Syst.* 2016, *64*, 186–197. [CrossRef]

31. Poirson, E.; Da Cunha, C. A recommender approach based on customer emotions. *Expert Syst. Appl.* 2019, *122*, 281–288. [CrossRef]

32. Yan, H.; Yang, C.; Yu, D.; Li, Y.; Jin, D.; Chiu, D.M. Multi-site user behavior modeling and its application in video recommendation. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017.

33. Filho, R.J.; Wehrmann, J.; Barros, R.C. Leveraging deep visual features for content-based movie recommender systems. In Proceedings of the International Joint Conference on Neural Networks, Anchorage, AK, USA, 14–19 May 2017; pp. 604–611.

34. Wang, H.; Wang, N.; Yeung, D.Y. Collaborative deep learning for recommender systems. In Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 1235–1244.

35. Wang, H.; Shi, X.; Yeung, D.Y. Relational stacked denoising autoencoder for tag recommendation. In Proceedings of the National Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 3052–3058.

36. Cheng, H.T.; Koc, L.; Harmsen, J. Wide & Deep learning for recommender systems. *arXiv* 2016, arXiv:1606.07792.

37. Lucas, J.M.; Saccucci, M.S.; Baxley, R.V.; Woodall, W.H.; Maragh, H.D.; Faltin, F.W.; Harris, T.J. Exponentially weighted moving average control schemes: Properties and enhancements. *Technometrics* 1999, *32*, 1–12. [CrossRef]

38. Aloise, D.; Deshpande, A.; Hansen, P.; Popat, P. NP-hardness of Euclidean sum-of-squares clustering. *Mach. Learn.* 2009, *75*, 245–248. [CrossRef]
39. Dhanalakshmy, D.M.; Pranav, P.; Jeyakumar, G. A survey on adaptation strategies for mutation and crossover rates of differential evolution algorithm. *Int. J. Adv. Sci.* **2016**, *6*, 613–623. [CrossRef]

40. Abraham, K.T.; Ashwin, M.; Sundar, D. Empirical comparison of different key frame extraction approaches with differential evolution based algorithms. In Proceedings of the International Symposium on Intelligent Systems Technologies and Applications, Udupi, India, 13–16 September 2017; pp. 317–326.

© 2019 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 (http://creativecommons.org/licenses/by/4.0/).