mathematics

Article

CNN-Based Temporal Video Segmentation Using a Nonlinear Hyperbolic PDE-Based Multi-Scale Analysis

Tudor Barbu [1,2]

[1] Institute of Computer Science of Romanian Academy—Iasi Branch, 2 T. Codrescu Street, 700481 Iași, Romania; tudor.barbu@iit.academiaromana-is.ro
[2] The Academy of Romanian Scientists, 3 Ilfov Street, Sector 5, 050663 Bucharest, Romania

Abstract: An automatic temporal video segmentation framework is introduced in this article. The proposed cut detection technique performs a high-level feature extraction on the video frames, by applying a multi-scale image analysis approach combining nonlinear partial differential equations (PDE) to convolutional neural networks (CNN). A nonlinear second-order hyperbolic PDE model is proposed and its well-posedness is then investigated rigorously here. Its weak and unique solution is determined numerically applying a finite difference method-based numerical approximation algorithm that quickly converges to it. A scale-space representation is then created using that iterative discretization scheme. A CNN-based feature extraction is performed at each scale and the feature vectors obtained at multiple scales are concatenated into a final frame descriptor. The feature vector distance values between any two successive frames are then determined and the video transitions are identified next, by applying an automatic clustering scheme on these values. The proposed PDE model, its mathematical investigation and discretization, and the multi-scale analysis based on it represent the major contributions of this work. Some temporal segmentation experiments and method comparisons that illustrate the effectiveness of the proposed framework are finally described in this research paper.

Keywords: temporal video segmentation; cut detection; multi-scale analysis; nonlinear hyperbolic PDE model; finite difference method; numerical approximation algorithm; mathematical treatment; deep learning; convolutional neural networks; frame clustering

MSC: 35Axx; 35A01; 35L51; 62H35; 65D18; 68W25; 68Txx; 68T07; 68T45; 94A08; 97N80

check for updates

Citation: Barbu, T. CNN-Based Temporal Video Segmentation Using a Nonlinear Hyperbolic PDE-Based Multi-Scale Analysis. *Mathematics* **2023**, *11*, 245. https://doi.org/10.3390/math11010245

Academic Editor: Christopher Goodrich

Received: 6 December 2022
Revised: 21 December 2022
Accepted: 30 December 2022
Published: 3 January 2023

1. Introduction

The temporal video segmentation field represents an important computer vision subdomain [1]. It has been applied in a large variety of video analysis fields, such as the video compression, the video sequence indexing and retrieval, or the object detection and tracking.

This type of video segmentation consists of dividing the movie sequence into a number of temporal segments, such as shots and scenes. While the video shot represents a continuous sequence of frames that are shot uninterruptedly by a single camera, the scene of a video is a succession of semantically correlated shots [2].

The video transitions, which represent the mechanism used to change from one shot to the next one in a video sequence, could be grouped into three classes: hard cuts, soft cuts and digital effects [1–3]. Hard cuts, which are called simply cuts, represent the most common transitions and constitute sudden transitions between consecutive shots. The soft cuts are gradual transitions between successive shots, meaning a sequence of frames belonging to both of them, and may represent fades or dissolves. Digital effects that are used for shot transitions include animated effects, wipes, color replacement, lighting effects, pixelization, focus drops and others.

We consider only the hard cut detection task in this research paper. Many cut detection approaches have been developed in the last 30 years and are grouped into several main categories. The pixel difference-based techniques measure the discontinuity of the visual content comparing the corresponding pixel intensities between two successive frames. Sum of absolute differences (SAD) [1–3] or pair-wise pixel comparisons [4] are examples of these methods. The histogram comparison-based video cut detection approaches measure the similarity of the grayscale/color histograms that correspond to adjacent video frames using metrics such as histogram difference, histogram quadratic distance and histogram intersection [2,3]. Edge-based shot detection methods include approaches based on edge change ratio (ECR), edge tracking and edge histograms [1–3,5]. The video motion-based shot detection techniques apply the motion estimation to determine the motion breaks, that may indicate the presence of abrupt video transitions [5–7]. The cut detection algorithms using statistical features break the video frames into regions and compare the statistical measures (such as those based on mean or the standard deviation) of the pixels in those zones [1–4,7]. A recently developed fast statistical measure-based shot detection technique is based on separable moments and SVM classifiers [8].

Other temporal segmentation algorithms use the concept of visual rhythm, which represents a simplification of the video sequence into a static image [9]. Fuzzy logic-based video approaches that could detect properly both the abrupt and the gradual cuts were also introduced [10]. Shot boundary detection solutions based on Principal Component Analysis (PCA) and deep learning were also developed [11,12]. Some effective detection methods combine multiple invariant features, such as edge change ratio, color descriptors and SIFT features [13].

We have also developed some video shot segmentation approaches. The most important of them performs a 2D Gabor filtering-based frame feature extraction [14]. Now, we propose a new video cut detection technique that overcomes some disadvantages of the existing shot detection methods. Some of them, such as those based on pixel differences, pair-wise pixel comparisons or edges, are quite sensitive to object and camera motion and produce a lot of false hits. Other methods, such as those using histograms, may generate many missed hits, since they disregard the spatial distribution. The detection models based on video motion and statistical features are characterized by a high computational cost and running time. Additionally, many methods are not automatic or rely on error-producing thresholds. The proposed CNN-based framework performs a high-level feature extraction that generates powerful frame descriptors ensuring a successful shot-transition detection. It is fully automatic, does not use threshold values and has a motion-insensitive character.

The main contribution of this research work is the novel nonlinear hyperbolic second-order PDE-based model that is proposed, mathematically treated and solved numerically in Section 2. This work aims to illustrate how such a well-posed PDE model can be used in combination to deep learning models to perform an effective multi-scale analysis of a movie sequence, which leads to a successful temporal segmentation. Its mathematical validity (well-posedness) is rigorously investigated and demonstrated here. Thus, this PDE admits a weak and unique solution under some certain conditions, which is then computed numerically applying a finite difference method-based approximation algorithm. The proposed iterative fast-converging numerical approximation scheme is stable and consistent to the PDE model it solves and it is used successfully to create a scale–space representation. The multi-scale analysis of the video frames, which uses this scale space, is described in Section 3. It extracts the high-level content features at each scale by applying a combination of two pre-trained convolutional neural networks. Next, the CNN-based feature vectors obtained at multiple scales are concatenated into the final frame descriptor. The values of the feature vector distances between adjacent frames are determined, then grouped automatically using a hierarchical clustering approach, in the cut detection process presented in Section 4. The video segmentation experiments and method comparisons described in Section 5 illustrate the effectiveness of the proposed technique. The conclusions of this research are drawn in Section 6.

2. Nonlinear PDE-Based Filtering Model for Scale-Space Representation

We have performed a large quantity of research in the PDE-based image processing and analysis domain, and developed many PDE and variational models for image denoising [15,16], inpainting [17], compression [18], edge detection [19] and segmentation [20], in the last 15 years. Since the partial differential equations provide more effective scale spaces than the well-known 2D Gaussian filter [21], we consider here a novel hyperbolic PDE model for multi-scale image analysis. The proposed model is described in Section 2.1 and a mathematical treatment of its well-posedness is performed in the Section 2.2. Then, a numerical approximation algorithm that solves it is provided in Section 2.3.

2.1. A Nonlinear Second-Order Hyperbolic PDE Model

Here, we introduce a nonlinear second-order PDE-based filtering model that is composed of the following hyperbolic partial differential equation and its boundary conditions:

$$\begin{cases} \alpha\frac{\partial^2 u}{\partial t^2} + \beta\frac{\partial u}{\partial t} - \eta\nabla\cdot(\psi(\|\nabla u_\sigma\|)\nabla u) + \lambda(u-u_0) = 0 \\ u(x,y,0) = u_0(x,y), \forall(x,y) \in \Omega \subseteq R^2 \\ u_t(x,y,0) = u_1(x,y), \forall(x,y) \in \Omega \\ u(x,y,t) = 0, \forall\ (x,y) \in \partial\Omega \\ \frac{\partial u}{\partial \vec{n}}(x,y,t) = 0, \forall(x,y) \in \partial\Omega \end{cases} \tag{1}$$

where the parameters $\alpha, \beta, \eta, \lambda \in (0,1]$; $\partial\Omega$ is the frontier of the image domain $\Omega \subseteq R^2$; the observed image $u_0 \in L^2(\Omega)$, $u = u(x,y,t)$ represents the evolving image function, $u_\sigma = u * G_\sigma$, where the 2D Gaussian filter kernel $G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$.

The following diffusivity function of the model has been properly selected for an effective filtering process, since it is positive, monotonically decreasing and converges to zero [15,22]:

$$\psi : [0,\infty) \to [0,\infty), \psi(s) = \left(\frac{\xi}{|\delta s^k + \gamma|}\right)^{\frac{1}{k-1}}, \tag{2}$$

where $\delta \in (0,1], \xi \geq 4$, $\gamma \geq 5$ and $k \in \{2,3,4,5\}$.

This PDE-based filtering model provides an effective detail-preserving restoration of any image corrupted by the additive white Gaussian noise (AWGN) and overcomes the unintended side effects, such as blurring and staircasing. Its second-time derivative, $\frac{\partial^2 u}{\partial t^2}$, that provides the hyperbolic character of this nonlinear PDE, sharpens the image's edges, thus enhancing its essential details.

The second-order nonlinear PDE model provided by (1) is non-variational, since it cannot be derived from the minimization of any energy cost functional. Additionally, we demonstrate that the proposed mathematical model is well-posed, which means it exists as a unique and weak (variational) solution, for it, under certain conditions. This mathematical validity (well-posedness) of the hyperbolic PDE-based model will be covered in Section 2.2.

Then, its solution will be determined numerically by applying a numerical approximation scheme for (1), which is created by using the finite difference method [23]. This stable and fast-converging iterative discretization algorithm that is consistent and solves numerically the proposed hyperbolic differential model will be described in Section 2.3.

2.2. Mathematical Treatment of PDE Model's Validity

A mathematical investigation is performed on the nonlinear second-order hyperbolic PDE model given by (1)–(3), in order to demonstrate its well-posedness, or mathematical validity. So, a PDE model is well-posed if it admits a unique weak, or variational, solution, that is also unique.

One performs an integration operation on the equation in (1) and obtains the next integral model:

$$\begin{cases} \frac{\partial u}{\partial t}(x,y,t) - \frac{1}{\alpha}\int\limits_0^t e^{\frac{\beta^2}{\alpha}(t-s)}(\eta div(\psi(\|\nabla u_\sigma(x,y,s)\|)\nabla u(x,y,s)) + \lambda(u(x,y,s) - u_0(x,y)))ds = u_1(x,y) \\ u(0,x,y) = u_0(x,y) \\ u(x,y,t) = 0, \text{ on } \partial\Omega \times (0,T) \end{cases} \tag{3}$$

A modified version of this integral model could be well-posed under some certain conditions [24]. Therefore, we may replace (3) by the next model:

$$\begin{cases} \frac{\partial u}{\partial t} - \varphi\Delta u - \int\limits_0^t (\eta div(\psi(\|\nabla u_\sigma\|)\nabla u) + \lambda(u - u_0))ds = 0 \\ u(x,y,0) = u_0(x,y) \\ u(x,y,t) = 0 \text{ on } \partial\Omega \times (0,T) \end{cases}. \tag{4}$$

The integral Equation (4) admits a solution for $\varphi > 0$ if the next conditions hold [9,18]:

$$\begin{cases} (\psi(\|v\|)v - \psi(w)w)(v-w) \geq 0,\ \forall v,w \in R^2 \\ \exists a,b : a \geq \psi(s) \geq b > 0, \forall r \geq 0 \\ \psi - continuous \end{cases} \tag{5}$$

Since $\psi\prime(s) \geq 0$, we have:

$$\frac{\partial(\psi(s)s)}{\partial s} = \psi\prime(s)s + \psi(s) \geq 0, \forall s \in R^+, \tag{6}$$

which means the function $s \to \psi(s)s$ is monotone in R^2, so the first condition in (5) holds.

The second condition of (5) also holds, because the function ψ is bounded. Since it is also continuous on the interval $[0,\infty)$, the third condition in (5) also holds.

Under these conditions the integral problem (4) admits a unique and weak solution $u* : \Omega \times (0,T) \to \mathbb{R}$ in sense of distributions [24], where parameter $T > 0$. That means:

$$u* \in L^\infty\Big(0,T;H_0^1(\Omega)\Big), \frac{\partial u^*}{\partial t} \in L^2\Big(0,T;L^2(\Omega)\Big), \tag{7}$$

$$\nabla \cdot (\psi(\|\nabla u_\sigma^*\|)\nabla u^*) \in L^\infty\Big(0,T;L^2(\Omega)\Big) \tag{8}$$

and

$$\begin{aligned} &\int\limits_\Omega \frac{\partial}{\partial t}u^*(x,y,t)\chi(x,y)dxdy + \varphi\int\limits_\Omega \nabla u^*(x,y,t) \cdot \nabla\chi(x,y)dxdy + \\ &\int_0^t ds\int\limits_\Omega \eta\psi(\|\nabla u_\sigma^*(x,y,t)\|)\nabla u^*(x,y,t) \cdot \nabla\chi(x,y) + \lambda(u^*(x,y,s) - u_0(x,y))\chi(x,y)dxdy = 0, \\ &u^*(x,y,0) = u_0(x,y), \forall x,y \in \Omega, \forall\chi \in H_0^1(\Omega) \end{aligned} \tag{9}$$

where the Sobolev space $H_0^1(\Omega)$=$\Big\{u \in L^2(\Omega); \nabla u \in \big(L^2(\Omega)\big)^2, u = 0 \text{ on } \partial\Omega\Big\}$.

The integral model (4) represents a very good approximation of the model (3). Additionally, for $\varphi \to 0$, the solution of (4) converges in a certain weak sense to a solution of the model (3).

2.3. Numerical Approximation Algorithm

The well-posed nonlinear second-order hyperbolic model is solved numerically by creating a finite difference method-based numerical approximation scheme that converges to its weak solution [23]. Thus, a grid of space size h and the time step Δt is used for this discretization task.

So, the space and time coordinates are quantized, for the $[Ih \times Jh]$ support image, as: $x = ih, y = jh, i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}$ and $t = n\Delta t, n \in \{0, \ldots, N\}$. The Equation in (1) could be written as:

$$\alpha \frac{\partial^2 u}{\partial t^2} + \beta \frac{\partial u}{\partial t} + \lambda(u - u_0) = \eta div(\psi(\|\nabla u_\sigma\|)\nabla u). \tag{10}$$

The left term of the above equation is then discretized, applying central differences [15,23], as:

$$\alpha \frac{u_{i,j}^{n+\Delta t} + u_{i,j}^{n-\Delta t} - 2u_{i,j}^{n}}{\Delta t^2} + \beta \frac{u_{i,j}^{n+\Delta t} - u_{i,j}^{n-\Delta t}}{2\Delta t} + \lambda\left(u_{i,j}^{n} - u_{i,j}^{0}\right) = u_{i,j}^{n+\Delta t}\left(\frac{\alpha}{\Delta t^2} + \frac{\beta}{2\Delta t}\right) + u_{i,j}^{n-\Delta t}\left(\frac{\alpha}{\Delta t^2} - \frac{\beta}{2\Delta t}\right) + u_{i,j}^{n}\left(\lambda - \frac{2\alpha}{\Delta t^2}\right) - u_{i,j}^{0}\lambda, \tag{11}$$

which leads to $u_{i,j}^{n+1}\left(\frac{2\alpha+\beta}{2}\right) + u_{i,j}^{n-1}\left(\frac{2\alpha-\beta}{2}\right) + u_{i,j}^{n}(\alpha - 2\lambda) - u_{i,j}^{0}\lambda$ for $\Delta t = 1$.

Next, the right component of (10) is approximated. The term $div(\psi(\|\nabla u_\sigma\|)\nabla u)$ could be approximated as $\varsigma \sum\limits_{q \in N_p} \psi\left(\left|\nabla u_{p,q}^{n}\right|\right)\nabla u_{p,q}^{n}$, where $\varsigma \in (0,1)$, the set of pixels N_p represents the 4-neighborhood of the pixel p, given as a pair of coordinates (i, j), and the gradient magnitude in a particular direction at iteration n is:

$$\nabla u_{p,q}^{n} = u(q,n) - u(p,n) \tag{12}$$

One may consider h = 1 and the next explicit numerical approximation algorithm is obtained:

$$u_{i,j}^{n+1} = u_{i,j}^{n}\left(\frac{4\lambda - 2\alpha}{\beta + 2\alpha}\right) + u_{i,j}^{n-1}\left(\frac{\beta - 2\alpha}{\beta + 2\alpha}\right) + u_{i,j}^{0}\frac{2\lambda}{\beta + 2\alpha} + \eta\varsigma \sum_{q \in N_p} \psi\left(\left|\nabla(u * G_\sigma)_{p,q}^{n}\right|\right)\nabla u_{p,q}^{n}. \tag{13}$$

The explicit iterative finite difference-based numerical approximation scheme in (13) is stable and consistent to the hyperbolic PDE model (1) and converges fast to its variational solution representing the filtered image u^{N+1}. This numerical solving algorithm is next used successfully to create an effective scale–space representation for the multi-scale analysis.

3. Multi-Scale Deep Learning-Based High-Level Frame Feature Extraction

A multi-scale analysis for high-level video frame feature extraction is performed in this section. The creation of a scale space using the numerical approximation scheme of the PDE model introduced in Section 2 is described in Section 3.1. Then, the proposed deep learning-based video feature extraction is presented in Section 3.2.

3.1. PDE-Based Scale Space

One represents the RGB color video sequence as $V = [F_1, \ldots, F_M]$, where F_i represent its frames, $i \in \{1, \ldots, M\}$ and M is large enough. An effective high-level feature extraction is performed on all the frames of this movie, as a first step of the temporal video-segmentation procedure.

The PDE-based scale–space used by this feature extraction is created by applying the numerical approximation scheme (10) on the current frame and considering the images obtained at various iteration moments.

Since our discretization algorithm works for gray-level images only, we consider a solution that works for the RGB frames. So, one may apply (10) on each of the three color channels of a frame, but this approach may not work properly, since the R, G and B channels could have high levels of correlation. Therefore, a much better solution is to convert the RGB frame F_i to the de-correlated color space CIE $L^*a^*b^*$, then to filter its luminance channel $L(F_i)$ by applying (13). The filtered L^*a^*b image $\left[(L(F_i))^n, a(F_i), b(F_i)\right]$, where $(L(F_i))^n$ is the luminance channel-filtering result after n iterations, is then converted back to the RGB form. So, one achieves the next multi-scale representation of K scales for that frame:

$$S(i) = \left\{F_i, RGB\left(\left[(L(F_i))^r, a(F_i), b(F_i)\right]\right), \ldots, RGB\left(\left[(L(F_i))^{r(K-1)}, a(F_i), b(F_i)\right]\right)\right\}, \quad (14)$$

where $r \in [3, 10]$ represents the iteration step, $K \geq 3$ and RGB ( ) converts the argument image to the RGB form. The obtained scale space S is next used by the high-level image feature extraction described in Section 3.2.

3.2. Deep Learning-Based Feature Extraction

The high-level characteristics of the video frames are extracted by applying a deep learning-based technique. Thus, a multi-scale content feature extraction using the scale-space representation S given by (14) is performed for all the frames of V. A deep learning-based feature extraction is applied on the current frame F_i at each scale $k \in \{0, \ldots, K-1\}$, where its image has the form $S(i)\{k\} = RGB\left(\left[(L(F_i))^{rk}, a(F_i), b(F_i)\right]\right)$.

The Convolutional Neural Networks (CNNs or ConvNets) represent deep neural networks which are able to learn high-level feature representations for various types of images when trained on voluminous databases storing a large variety of digital images [25]. These deep learning models generate powerful content characteristics which outperform the image features produced by the classic descriptors.

Here, we create a combination of two convolutional neural networks to determine the high-level content features of each video frame. The two pre-trained CNNs considered by us are Inception-ResNet-V2 and DenseNet-201. These ConvNets outperform other pre-trained CNN models, such as GoogleNet, VGGNet-16, VGGNet-19 or AlexNet, which have fewer convolutional layers [26].

The first one, Inception-ResNet-V2, represents a deep convolutional neural architecture which builds on the Inception family while incorporating residual connections [27]. It is 164 layers deep and it has been trained on more than a million images of the voluminous ImageNet database containing 1000 object categories [28]. This deep network constitutes an effective image classification tool. Its architecture is described in Figure 1 [27].

The second model, DenseNet-201, is a DenseNet convolutional neural network whose architecture has 201 layers [29]. A DenseNet represents a ConvNet characterized by dense connections between layers, through Dense Blocks, where one connects all layers directly with each other, in a feed-forward fashion. So, each layer achieves feature maps as inputs from all preceding layers and passes on its own feature maps as inputs to all subsequent layers, in order to preserve the feed-forward character.

It has been trained on 1.2 million training images from the ImageNet collection [26,28]. Since DenseNet-201 has learned rich feature representations for a large variety of images, it is successfully used to recognize new digital images and image objects [29]. The design of its architecture is displayed in Figure 2 [29].

Thus, network activations are computed by forward propagating the input image through each CNN up to the specified layer. While the layers of these pre-trained deep neural networks generate activations on the input images, not all of them have the same capacity of feature extraction. The first convolutional layers of these CNNs can compute only low-level image characteristics that are next processed by the deeper layers which mix them in order to achieve higher level image-feature representations.

So, the proposed method extracts the required content characteristics from the deep layers of Inception-ResNet-V2 and DenseNet-201 and then combines those features. It uses for this process the fully connected layers of the two networks, more precisely the *predictions* layer of Inception-ResNet-v2, which is located before the final classification layer, and FC 1000 which is the fully connected layer with 1000 neurons of DenseNet-201.

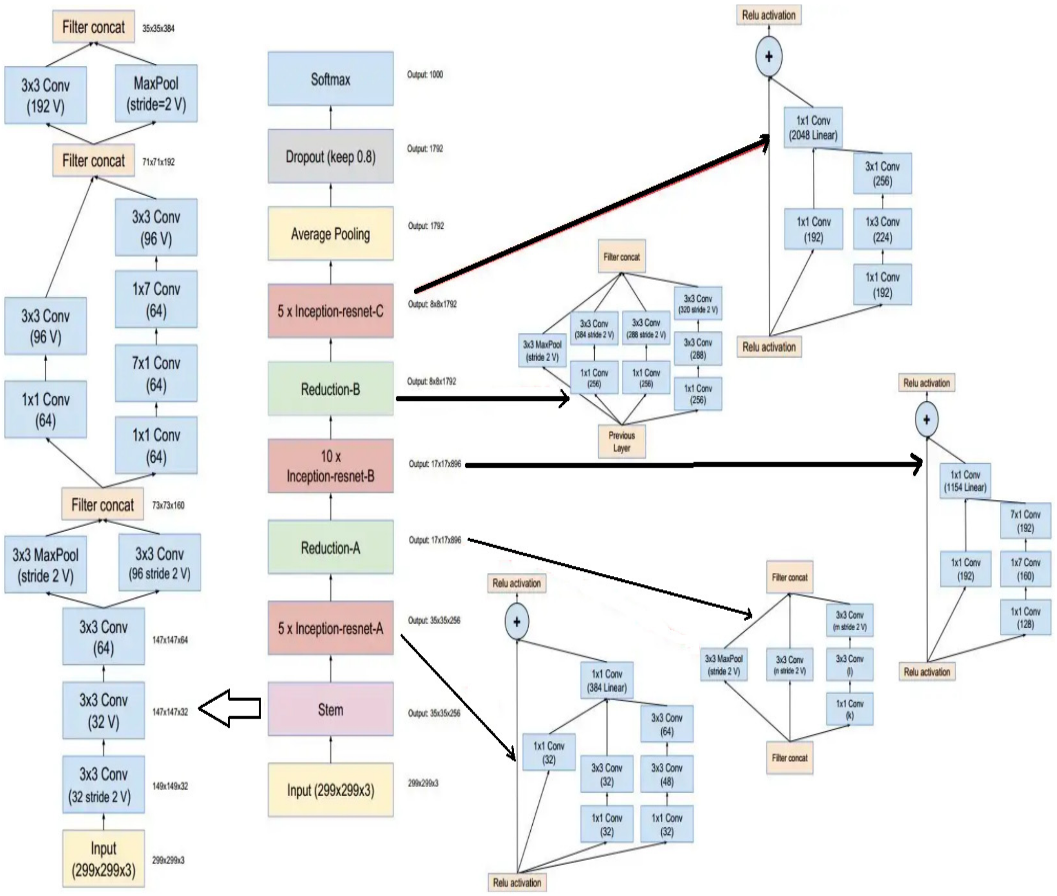

Figure 1. Inception-ResNet-V2 architecture.

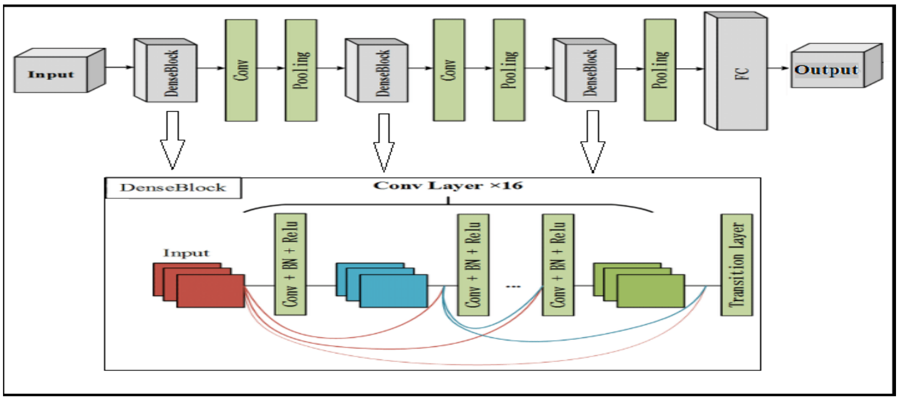

Figure 2. DenseNet-201 architecture.

First, each $S(i)\{k\}$ image of the F_i frame is pre-processed according to the specifications of the input layer of the Inception-ResNet-V2 model. Thus, it is resized at the $[299 \times 299 \times 3]$ format required by the network and the next 0-center normalization is also performed on it:

$$S(i)\{k\} := \frac{S(i)\{k\} - \mu(S(i)\{k\})}{\sigma(S(i)\{k\})}, \ \forall k \in \{0, \ldots, K-1\}. \tag{15}$$

The processed $S(i)\{k\}$ is then fed into the Inception-ResNet-V2 model. The predictions layer of the network generates an activation on $S(i)\{k\}$ that determines its high-level characteristics in the form of the feature vector $V_{IRN}(S(i)\{k\})$ with 1000 coefficients.

DenseNet-201 is next applied on the initial version of $S(i)\{k\}$. It is resized at $[224 \times 224 \times 3]$, which is the input image-size format of this CNN, and the 0-center normalization given by (15) is then applied. Next, the pre-processed image is fed into DenseNet-201 whose FC 1000 layer produces an activation generating the high-level feature vector $V_{DN}(S(i)\{k\})$, having also 1000 coefficients. A 2D feature vector is achieved at this scale by combining these two vectors through the following concatenation:

$$V_i(k) := [V_{IRN}(S(i)\{k\});\ V_{DN}(S(i)\{k\})]. \tag{16}$$

All these deep learning-based 2D feature vectors computed at multiple scales are then combined into a final two-dimension descriptor of the analyzed video frame, as follows:

$$Fv(i) := [V_i(0) \dots V_i(K-1)],\ \forall i \in \{1, \dots, M\}. \tag{17}$$

The final $[2 \times 1000K]$ high-level feature vectors $Fv(i)$ represent some powerful content descriptor of the frames F_i. These optimal high-level frame characterizations provided by these 2D feature vectors lead to successful discriminations between the video frames of V. This means that similar frames correspond to very close feature vectors, while dissimilar frames have very different feature vectors. Therefore, the proposed multi-scale CNN-based feature extraction leads to the optimal frame grouping process presented in the next section.

4. Automatic Video Frame Clustering Technique

The video frames of V, which are characterized by the high-level CNN-based content descriptors described in the previous section, must now be grouped in shots, by using these feature vectors. Other segmentation approaches perform the cut detection process using thresholding operations. They detect the shot breaks as the locations where the inter-frame difference metric exceeds a certain threshold value. However, they generate many detection errors, since an optimal threshold selection still constitutes a difficult task [7].

In order to solve this drawback, we consider here a temporal segmentation solution that does not rely on any threshold for transition detection, using an automatic video frame clustering scheme instead of thresholding. In addition, unlike other segmentation methods that use an a priori known number of shots, our cut detection technique uses no knowledge about it, being completely automatic.

The inter-frame difference metric used by the proposed technique represent the distances between the 2D feature vectors corresponding to adjacent frames. So, one determines the frame feature vector distance value set $\{d_1, \dots, d_{M-1}\}$ corresponding to V, where:

$$d_i = d(Fv(i), Fv(i+1)),\ i \in \{1, \dots, M-1\}, \tag{18}$$

and d computes a properly selected distance working for these vectors, for example, the 2D Euclidean metric.

Obviously, any inter-shot distance value has to be much higher than any intra-shot distance value. The metric values computed by (18) satisfy this shot segmentation condition, since the deep network-based feature vectors provide strong high-level frame characterizations. Therefore, we have:

$$\min_{i \in C} d_i \gg \max_{j \in \{1,\dots,M-1\} \setminus \mathrm{C}} d_j, \tag{19}$$

where the set $C \subseteq \{1, \dots, M-1\}$ containing the indices of the video cuts must be determined.

Since the number of optimal temporal segments is unknown, one applies an inter-frame distance clustering technique to identify the break point indices in C. So, one must group the values of $\{d_1, \dots, d_{M-1}\}$ into *high distances* and *low distances*, due to the property (19). The high (inter-shot) feature vector distance values indicate the abrupt transitions of the video sequence.

So, an automatic unsupervised classification (clustering) operation is applied on the distance set $\{d_i\}_{i\in\{1,\ldots,M-1\}}$. An agglomerative hierarchical clustering algorithm is used for this purpose [30,31]. Since that set has to be partitioned into two categories of distance values, the number of clusters used by this hierarchical clustering scheme is c = 2. The metric considered for this hierarchical clustering process is the average-linkage clustering. Our clustering approach labels each distance value with either 1, for *low*, or 2, for *high*. Since we have

$$label(d_i) = 2 \Leftrightarrow i \in C,\ \forall i \in \{1,\ldots,M-1\}, \tag{20}$$

the locations of the video cuts in V are thus detected. Obviously, the obtained set of cut locations, C, determines all the movie frame clusters representing temporal video segments (shots).

An example of temporal video segmentation that is based on the presented detection technique is described in Figure 3. Thus, the 10 shot transitions of a video sequence composed of 658 frames of size $[720 \times 1280 \times 3]$ have been identified at the locations of the eight high (inter-frame) feature vector distance values displayed in (i). These detected movie cuts are represented as pairs of frames in (a) to (h).

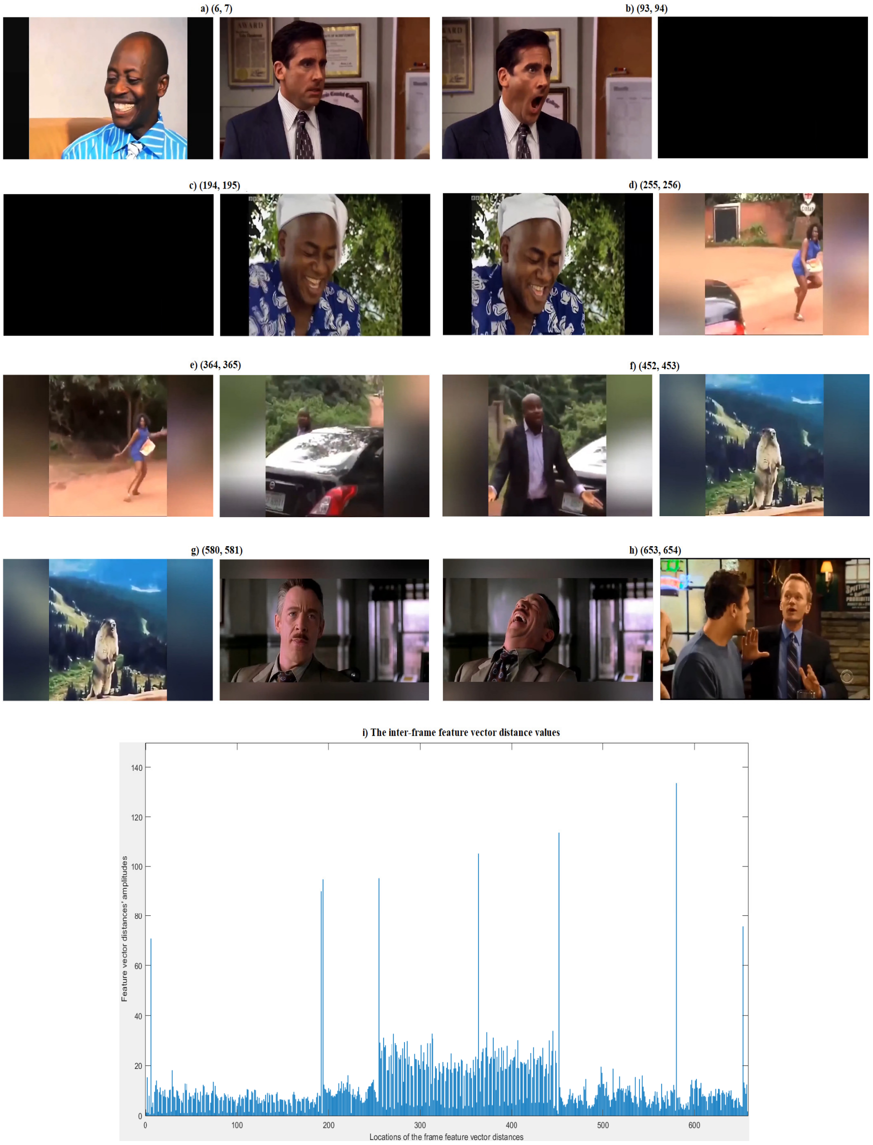

Figure 3. A temporal video segmentation example: (**a**–**h**) the pairs of frames related to the shot cuts; (**i**) the frame feature vector distance values (the *highest* ones indicate the cuts).

5. Discussion

The described CNN-based multi-scale video segmentation framework has been tested successfully on numerous movie sequences. The testing video dataset consisted of 14 Mp4 RGB clips containing over 30,000 frames of $[720 \times 1280 \times 3]$ and $[360 \times 640 \times 3]$ dimensions. The temporal segmentation experiments have been performed on an Intel (R) Core (TM) i7-6700HQ CPU 2.60 GHz processor on 64 bits, operating Windows 10, by using MATLAB software.

The video shot detection technique introduced here has achieved a high detection rate, based on the performed simulations. The proposed method generates only few missed hits (undetected cuts) and false hits (falsely detected cuts), thus producing very high scores for all the performance measures used to assess the detection and recognition quality: Precision, Recall and F_1 [32,33].

Video segmentation-method comparisons have also been performed. Due to its deep learning-based high-level feature extraction component, our cut detection approach outperforms the temporal segmentation schemes based on pixel differences, color/grayscale or edge histograms, and various statistical image features, which produce more detection errors (missed or false hits), and even the segmentation methods using SIFT, LBP, SURF and 2D Gabor filter-based descriptors [14].

The performance metrics' average scores obtained by our cut detection method and other techniques are described in Table 1. One can see that the proposed DL-based solution achieves higher Precision, Recall and F_1 values than the other movie segmentation approaches [32,33].

Table 1. Video segmentation method comparison results.

Technique	Precision	Recall	F1
The proposed technique	0.984	0.991	0.987
Color histograms	0.745	0.724	0.734
Edge histograms	0.815	0.752	0.782
Pixel differences (SAD)	0.775	0.763	0.769
Gabor 2D filter-based model	0.941	0.840	0.887
Statistical features with LHR	0.645	0.618	0.631
Pairwise pixel comparisons	0.741	0.723	0.731

The proposed framework has a high computational complexity in any event, given the number of procedures executed by its multi-scale deep neural network-based feature extraction that generates big-sized frame feature vectors which raise the cost of the inter-frame distance clustering. That means it does not run fast, its execution time being also influenced by the video's dimensions. This may represent a disadvantage compared to other segmentation models, such as those using the sum of absolute differences or various histograms, which may operate faster than the described approach, although being less performant in terms of the quality metrics [33].

The technique proposed here also works properly in both clean and noisy image conditions, because of its nonlinear PDE-based filtering process. However, as another drawback, while this shot detection method provides a very effective hard-cut identification, it achieves weaker segmentation results for other types of shot transition types, such as the gradual transitions or the digital effects.

6. Conclusions

A novel temporal movie-sequence segmentation technique has been described in this research work. The effectiveness of the proposed automatic video-shot detection framework is provided mainly by its high-level frame feature extraction component that is based on a multi-scale analysis combining successfully the deep learning- and partial differential equation-based models.

The nonlinear second-order hyperbolic PDE model introduced here, the mathematical investigation of its validity, its stable and consistent numerical approximation and the effective scale-space representation created by using its iterative numerical solving algorithm represent the main contributions of our research. The powerful frame content descriptors computed by applying a combination of the activations of two convolutional neural networks at the multiple scales of the obtained PDE-based scale space have determined a successful frame discrimination leading to an effective cut-detection result.

Unlike many other video segmentation approaches, the proposed shot identification technique is fully automatic. More precisely, it is not based on either a priori knowledge of the number of movie shots or inter-frame distance thresholds, using an automatic inter-frame feature vector distance-clustering algorithm instead.

The obtained temporal segmentation results prove that the deep neural networks can be applied more successfully in this computer vision field, when integrated into nonlinear PDE-based multi-scale frame analysis. The scale space created by applying the proposed hyperbolic PDE model provides a more effective multi-scale image analysis than the scale–space representations generated by the 2D Gaussian kernels. However, the performance of the obtained PDE-based scale–space representation has still not been compared to that of the scale spaces produced by the wavelet transformations [34].

Besides the mentioned benefits, the proposed framework has also some limitations. As already mentioned, it clearly outperforms the cut-detection schemes using lower level image features, in terms of quality metric scores, but may execute slower than some of them, because of the higher computational cost of its high-level multi-scale feature extraction component. So, improving the running time of this video segmentation framework will represent the focus of our future research. In addition, since our shot detection method does not work efficiently for the video transitions other than the abrupt ones, improving it in the direction of the detection of gradual transitions [35], such as those based on soft cuts or digital effects, will also represent a future research focus in this area. We also intend to develop some effective computer vision applications, which perform video indexing and retrieval and video object detection and tracking tasks, by using this segmentation framework.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Koprinska, I.; Carrato, S. Temporal video segmentation: A survey. *Signal Process Image Commun.* **2001**, *16*, 477–500. [CrossRef]
2. Zhu, X.; Aref, W.G.; Fan, J.; Catlin, A.C.; Elmagarmid, A.K. Medical video mining for efficient database indexing, management and access. In Proceedings of the 19th IEEE international conference on data engineering (ICDE 2003), Bangalore, India, 5–8 March 2003.
3. Boreczky, J.S.; Rowe, L.A. Comparison of video shot boundary detection techniques. *J. Electron. Imaging* **1996**, *5*, 122–128. [CrossRef]
4. Patel, U.; Shah, P.; Panchal, P. Shot detection using pixel wise difference with adaptive threshold and color histogram method in compressed and uncompressed video. *Int. J. Comput. Appl.* **2013**, *64*, 38–44. [CrossRef]
5. Jacobs, A.; Miene, A.; Ioannidis, G.T.; Herzog, O. Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In *TRECVID 2004 Workshop Notebook Papers*; NIST: Gaithersburg, MD, USA, 2004; pp. 197–206.
6. Tekalp, A.M. *Digital Video Processing*; Prentice-Hall: Hoboken, NJ, USA, 1995.
7. Yusoff, Y.; Christmas, W.; Kittler, J. Video shot cut detection using adaptive thresholding. In Proceedings of the 11th British Machine Vision Conference University of Bristol, Bristol, UK, 11–14 September 2000.
8. Idan, Z.N.; Abdulhussain, S.H.; Mahmmod, B.M.; Al-Utaibi, K.A.; Al-Hadad, S.A.R.; Sait, S.M. Fast Shot Boundary Detection Based on Separable Moments and Support Vector Machine. *IEEE Access* **2021**, *9*, 106412–106427. [CrossRef]
9. Guimarães, S.J.F.; Couprie, M. Video segmentation based on 2d image analysis. *Pattern Recognit. Lett.* **2003**, *24*, 947–957. [CrossRef]
10. Fang, H.; Jiang, J.; Feng, Y. A fuzzy logic approach for detection of video shot boundaries. *Pattern Recognit.* **2006**, *39*, 2092–2100. [CrossRef]

11. Chakraborty, D.; Chiracharit, W.; Chamnongthai, K. Video Shot Boundary Detection Using Principal Component Analysis (PCA) and Deep Learning. In Proceedings of the 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Virtual Conference, 19–22 May 2021. [CrossRef]
12. Xu, J.; Song, L.; Xie, R. Shot Boundary Detection Using Convolutional Neural Networks. In Proceedings of the 2016 Visual Communications and Image Processing (VCIP), Chengdu, China, 27–30 November 2016.
13. Jose, J.T.; Rajkumar, S.; Ghalib, M.R.; Shankar, A.; Sharma, P.; Khosravi, M.R. Efficient Shot Boundary Detection with Multiple Visual Representations. *Mob. Inf. Syst.* **2022**, *2022*, 4195905. [CrossRef]
14. Barbu, T. Novel automatic video cut detection technique using Gabor filtering. *Comput. Electr. Eng.* **2009**, *35*, 712–721. [CrossRef]
15. Barbu, T. *Novel Diffusion-Based Models for Image Restoration and Interpolation*; Book Series: Signals and Communication Technology; Springer International Publishing: Berlin/Heidelberg, Germany, 2019.
16. Barbu, T.; Miranville, A.; Moroșanu, C. A Qualitative Analysis and Numerical Simulations of a Nonlinear Second-order Anisotropic Diffusion Problem with Non-homogeneous Cauchy-Neumann boundary conditions. *Appl. Math. Comput.* **2019**, *350*, 170–180. [CrossRef]
17. Barbu, T. Second-order anisotropic diffusion-based framework for structural inpainting. *Proc. Rom. Acad. Ser. A* **2018**, *19*, 329–336.
18. Barbu, T. Feature Keypoint-Based Image Compression Technique Using a Well-Posed Nonlinear Fourth-Order PDE-Based Model. *Mathematics* **2020**, *8*, 930. [CrossRef]
19. Barbu, T. Automatic Edge Detection Solution using Anisotropic Diffusion-based Multi-scale Image Analysis and Fine-to-coarse Tracking. *Proc. Rom. Acad. Ser. A* **2021**, *22*, 267–274.
20. Barbu, T. Robust contour tracking model using a variational level-set algorithm. In *Numerical Functional Analysis and Optimization*; Taylor & Francis: Abingdon, UK, 2014; Volume 35, pp. 263–274.
21. Ren, X. *Multi-Scale Improves Boundary Detection in Natural Images. European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 533–545.
22. Weickert, J. *Anisotropic Diffusion in Image Processing*; European Consortium for Mathematics in Industry; B.G. Teubner: Stuttgart, Germany, 1998.
23. Johnson, P. *Finite Difference for PDEs*; School of Mathematics, University of Manchester, Semester I: Manchester, UK, 2008.
24. Barbu, V. *Nonlinear Semigroups and Differential Equations in Banach Spaces*; Noordhoff International Publishing: Groningen, The Netherlands, 1976.
25. Murphy, J. *An Overview of Convolutional Neural Network Architectures for Deep Learning*; Microway Inc.: Plymouth, MA, USA, 2016.
26. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
27. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *AAAI* **2017**, *4*, 12. [CrossRef]
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 84–90. [CrossRef]
29. Gao, H.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *CVPR* **2017**, *1*, 3.
30. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
31. Barbu, T. An automatic unsupervised pattern recognition approach. *Proc. Rom. Acad. Ser. A* **2006**, *7*, 73–78.
32. Powers, D.M. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
33. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2021**, *17*(1), 168–192. [CrossRef]
34. Kumar, P. A wavelet based methodology for scale-space anisotropic analysis. *Geophys. Res. Lett.* **1995**, *22*, 2777–2780. [CrossRef]
35. Bescos, J.; Martinez, J.M.; Cabrera, J.; Menendez, J.M.; Cisneros, G. Gradual shot transition detection based on multidimensional clustering. In Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation, Austin, TX, USA, 2–4 April 2000; pp. 53–57. [CrossRef]

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