

METHOD AND SYSTEM FOR THE AUTOMATIC IDENTIFICATION OF FUZZY COPIES OF VIDEO CONTENT

Field

This technique lies within the field of identifying violations of intellectual property rights; in particular, methods and systems to automatically identify copies of video content.

Background

Every year the film industry loses billions of dollars due to the distribution of pirated copies that are published online and become instantly available to a large number of users. Piracy means copying and distributing works, in particular films, cartoons, clips, TV series, and other video content protected by copyright, without the permission of the author or owner, or in violation of the conditions of an agreement on the use of such works. According to the data of the company Group-IB (<https://www.group-ib.ru/>) the average pirate movie theater makes its owners an annual income of approximately \$90,000, target audience outflow to pirate sites is approximately 60%, and loss of revenue suffered by media companies as a result of the actions of pirates may be as high as 45%.

The improvement of distribution tools, concealment of pirated copies online, and increase in the scale of violations combine to make the development of mechanisms to detect and suppress piracy a priority. The requirements for modern antipiracy solutions are speed, low search cost, and reduced involvement of the human analyst.

Traditional solutions to search for pirated video content use parsing of web pages. However, many web pages that contain pirated copies may be indistinguishable from pages with trailers that can be legally distributed. Also, for example, the first few episodes of a TV series may be freely distributed, so it is important to not only identify the TV series, but also to identify the specific episode whose duplicate is posted on a particular web resource.

Another method is metadata analysis, which allows a comparison of the length, frame size, recording quality, and other technical information. However, the search for a pirated copy of the sought film might be successful only when the copy is identical to the original, or not significantly different from it. When the original video content of the file is recoded, a fuzzy duplicate may be created, where the majority of its metadata may be different from the metadata of the original video content. A change of metadata is possible, for example, as a result of recoding

or compressing the original video content, changing its length, by deleting or adding frames or entire sections (in particular, embedding advertisements), etc. Furthermore, it is possible to edit the metadata of a file separately.

To identify fuzzy copies of video content, an automated review algorithm that simulates user behavior may be used. An automated web browser algorithm that can test the content of web pages is known in the art (see, e.g., project Selenium).

There are known technical solutions that can extract various characteristics from the original video and compare them with the characteristics of the potential fuzzy copies that have been found. The disadvantages of these methods include an increased load on computer resources and data transmission networks, which is caused by the need to download the entire file of a potential fuzzy copy for analysis. Global experience in the battle against cybercrime shows that digital pirates use reliable tools to bypass traditional algorithms for the detection of pirated content. Metadata and the content of materials are changed: video content is cropped, noise is introduced, the color range is changed, etc. For this reason, developing new methods to uncover pirated content that are substantially insensitive to video content changes is a very high priority in the art of identifying pirated copies of video content.

This technique is designed to automate and improve the quality of video content analysis in order to quickly and accurately identify fuzzy copies of video content.

Disclosure

The technical result of this technique is the elimination of at least some shortcomings inherent in the known state of the art.

The first object is a method to automatically identify fuzzy video content copies, the method performed on a computer device that has access to a previously generated database of original video content; the method comprising:

Receiving an identifier for video content;

Extracting each Nth frame from the video stream of received video content to obtain a sequence of frames, where $N > 1$;

Identifying the sequence of scenes by comparing at least fragments of adjacent frames from the obtained sequence of frames;

Creating a vector of the sequence of scenes of the received video content;

Comparing the vector of the sequence of scenes of obtained video content with the vector of the sequence of scenes of at least one template of original video content from the previously generated database of original video content;

Identifying a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold.

There is a possible embodiment, wherein, prior to the step of receiving the identifier, a search is performed for video content on a data transmission network, the search is performed according to one or several of the following parameters: keywords in the file name, keywords in the description, accompanying images, file metadata.

There is a possible embodiment, wherein the value of N is preset as a function of the length of the video content.

There is a possible embodiment, wherein, during the step of identifying scenes, a comparison of extracted hash sums of at least fragments of adjacent frames is performed in response to the Hamming distance between hash sums of adjacent frames exceeds a predefined threshold; the boundary of the scene (scene change) is identified.

There is a possible embodiment, wherein the vector of the sequence of scenes of received video content is created based on the total number of frames in the scene.

There is a possible embodiment, wherein the vector of the sequence of scenes of received video content is a vector of at least two disjoint subsets of scenes of received video content.

There is a possible embodiment, wherein a fuzzy copy of the original video content is identified in response to coincidence of the vector of at least two disjoint subsets of scenes of received video content and at least part of the vector of the sequence of scenes of original video content.

There is a possible embodiment, wherein, during the step of comparing the vector of the sequence of scenes of received video content with the vector of the sequence of scenes of at least one template of original video content, a comparison is also performed of at least one frame

from the scene of received video content with at least one frame from the corresponding scene of original video content, forming at least one comparative pair of frames.

There is a possible embodiment, wherein hash sums are extracted and the hash sums of a comparative pair of frames are compared.

There is a possible embodiment, wherein an additional search for fuzzy copies of video content containing metadata identical to the metadata of the identified fuzzy copy of video content is performed when the metadata of an identified fuzzy copy of video content differs at least partially from the metadata of the original video content.

There is a possible embodiment, wherein the length of the fuzzy copy differs from the length of the original video content.

There is a possible embodiment, wherein, after identifying a fuzzy copy of video content, the identifier for the video content is saved in a database of fuzzy copies.

Another object is **a method for creating a template of original video content** performed on a computer device, the method comprising:

Receiving identifiers for at least part of the original video content;

Extracting at least part of the metadata of the original video content;

Extracting at least part of the frames from the sequence of frames of original video content;

Identifying a sequence of scenes;

Creating a vector of the sequence of scenes;

Generating and saving in a database a template of the original video content, the template including at least part of the metadata, and the vector of the sequence of scenes of the original video content.

There is a possible embodiment, wherein at least one frame from each scene is saved in the template of the original video content.

Yet another object is a **computer device that automatically identifies fuzzy copies of video content**, has access to a previously generated database of original video content, and is connected to a data transmission network; this computer device includes a communication

interface and at least one computer processor that is functionally connected to the communication interface, the computer processor is configured to:

Receiving an identifier for video content;

Extracting each Nth frame from the video stream of received video content to obtain a sequence of frames, where $N > 1$;

Identifying the sequence of scenes by comparing at least fragments of adjacent frames from the obtained sequence of frames;

Creating a vector of the sequence of scenes of the received video content;

Comparing the vector of the sequence of scenes of obtained video content with the vector of the sequence of scenes of at least one template of original video content from the previously generated database of original video content;

Identifying a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold.

There is a possible embodiment of this device, wherein the processor is configured to perform a search for video content on a data transmission network, where the search is performed according to at least one of the following parameters: keywords in the file name, keywords in the description, accompanying images, file metadata.

There is a possible embodiment, wherein the processor is configured to perform a comparison of extracted hash sums of at least fragments of adjacent frames and to identify the boundary of the scene in response to when the Hamming distance between hash sums of adjacent frames exceeds a predefined threshold.

There is a possible embodiment, wherein the processor is configured to create a vector of the sequences of scenes of received video content based on the total number of frames in the scene.

There is a possible embodiment, wherein the processor is configured to create a vector of the sequence of scenes of received video content, which is a vector of at least two disjoint subsets of scenes of received video content.

There is a possible embodiment, wherein the processor is configured to identify a fuzzy copy of the original video content in response to the vector of at least two disjoint subsets of scenes of received video content coincides with at least part of the vector of the sequence of scenes of the original video content.

There is a possible embodiment, wherein the processor is configured to perform a comparison of at least one frame from the scene of received video content and at least one frame from the corresponding scene of original video content and generate at least one comparative pair of frames.

There is a possible embodiment, wherein the processor is configured to extract hash sums and compare the hash sums of a comparative pair of frames.

There is a possible embodiment, wherein the processor is configured to perform a supplemental search for fuzzy copies of video content that contain metadata identical to the metadata of an identified fuzzy copy of video content in response to the metadata of the identified fuzzy copy of video content differs at least partially from the metadata of the original video content.

There is a possible embodiment of device, the device also has access to a database of fuzzy copies and the processor is configured to save identifiers for video content in the database of fuzzy copies.

There is a possible embodiment, wherein the processor is also configured to:

Receive identifiers for at least part of the original video content;

Extract at least part of the metadata of the original video content;

Extract at least part of the frames from the sequence of frames of original video content;

Identify a sequence of scenes;

Create a vector of the sequence of scenes;

Generate and saving in a database a template of the original video content that includes at least part of the metadata, and the vector of the sequence of scenes of the original video content.

There is a possible embodiment, wherein the processor is also configured to save at least one frame from each scene in a template of the original video content.

In the context of this description "fuzzy copy of video content" means a video that is approximately the same as the original video content, but may differ from the original video content in one or several of the following ways in particular: coding parameters, photometric parameters (chromaticity, brightness), use of editing (frame, titles, logo, etc.). A "fuzzy copy of video content" may also be an identical video with relevant additional information or a lack of at least part of the information, for example, additional advertisements, specific added and/or deleted frames or scenes, etc.

In the context of this description, "original video content" means any source video file concerning which a search and identification of fuzzy copies is performed. For example, original video content may be a film, pirated copies of which must be found and identified in order to block and suppress copyright violation.

In the context of this description, a "scene" of video content means a set of related frames that, for example, were filmed in one shot, contain the same objects, actors, etc. Visual differences between a sequence of two or more adjacent frames in one scene are insignificant, and conversely, visual differences between the last frame of one scene and the first frame of the next scene differ substantially. The degree of similarity and difference may be predefined. For example, a coincidence of 20% or more between two frames may be considered significant.

In the context of this description, "vector of sequence of scenes" means an expression of the quantity of frames in scenes or the length of scenes (scenes), a sequence of which forms video content.

In the context of this description, "computer device" means a computer program operating on relevant hardware, which is able to receive requests (for example, from client devices) via a network and to execute these requests or initiate the execution of these requests. The equipment may consist of one physical computer or one physical computer system, but neither is required for this invention. In the context of this description, the use of the expression "computer device" does not mean that each request (for example, received commands or requests) or any particular request will be received, executed, or have its execution initiated by the same "computer device" (that is, the same software and/or hardware); this means that any number of elements of software or hardware may be involved in receiving/transmitting, executing or initiating the execution of any

request or consequence of any request related to the client device, and all this software and hardware may consist of a single computer device or several computer devices. Both options are included in the expression "at least one computer device."

In the context of this description, "database" means any structured set of data, regardless of the specific structure, the software used to manage the database, or the computer hardware on which the data is stored, used, or otherwise made accessible for use. The database may be located on the same hardware that runs the process that stores or uses the information stored in the database, or it may be located on separate hardware, for example, a dedicated server or set of servers.

In the context of this description, unless expressly indicated otherwise, the term "video content" or "content" means any visual or audio-visual data, which could be, for example, video files.

In the context of this description, unless expressly indicated otherwise, the "locator" or "identifier" of an information element may be the information element itself or a locator, reference, link, or other indirect method that allows the recipient of the identifier to find the network, memory, database, or other machine-readable medium from which the information element may be extracted. For example, an identifier for video content may include a video content file itself, or a set of video content files, or it may even be a uniform resource locator (for example, URL www.webpage.com/videofile.avi) that identifies the video content in relation to a specific network (in particular, the Internet), or some other means to transfer the identifier to the recipient for a network folder, memory address, table in a database, or other place in which the video content or its individual fragments can be accessed. As will be understood by those skilled in the art, the degree of specificity required for such an identifier depends on the degree of originally understanding how the information exchanged by the sender and receiver of the locator should be interpreted. For example, if it is understood prior to transmitting data between the sender and recipient that the information element locator takes the form of a Universal Resource Locator (URL), transmitting the locator and link to this video content is all that is necessary to effectively transmit the video content to the recipient, despite the fact that the information element itself (for example, a film or a fragment of it) was not transmitted between the sender and recipient of the identifier.

In the context of this description, unless expressly indicated otherwise, the words "first," "second," "third," etc. are used as adjectives solely in order to distinguish the nouns they modify from one another, and not for purposes of describing any specific relation between these nouns. For example, it should be borne in mind that the use of the terms "first scene" and "third scene" do not imply any sort of order, assignment to a specific category, chronology, hierarchy, or ranking (for example) of scenes within the set of scenes, similarly their use (by itself) does not imply that a certain "second scene" must exist in any given situation. Furthermore, as specified here in other contexts, mentioning a "first" element and a "second" element does not rule out the fact that they could actually be the same element. For example, in some cases, the "first" server and the "second" server may be the same software and/or hardware, and in other cases they may be different software and/or hardware.

In the context of this description "machine-readable medium" means a medium of absolutely any type and nature, including RAM, ROM, disks (CDs, DVDs, floppy disks, hard disks, etc.), USB flash drives, solid state drives, tape drives, etc.

Additional and/or alternative features, aspects, and advantages of different embodiments of this technique will be apparent from the following description, drawings, and claims.

Brief Description of the Drawings

In order to better understand this invention, as well as its other aspects and characteristic features, reference is made to the following description, which should be used in conjunction with the accompanying drawings, where:

Fig. 1 is a simplified schematic representation of one of the non-limiting implementations of a system to embody this invention.

Fig. 2 is a flowchart that illustrates the aspect of this method for creating a template of the original video content.

Fig. 3 is a flowchart that illustrates the aspect of this method for automatically identifying fuzzy copies of video content.

Embodiment

The following description serves only as a description of an illustrative example of this invention. This description is not intended to define the scope or establish limitations on this invention.

Some useful examples of modifications to the described method and system for identifying malicious web resources may also be covered by the following description. The goal of this is also exclusively to assist in understanding, and not to define the scope or limitations to this invention. These modifications do not constitute an exhaustive list, and persons skilled in the art will recognize that other modifications are possible. Furthermore, it should not be interpreted in such a way that no modifications are possible where no examples of modifications have been stated, and/or that something described is the only option for implementing a certain element of this invention. As will be clear to a person skilled in the art, this is most likely not the case. Furthermore, it should be borne in mind that, for several specific manifestations, the method and system for automatically identifying fuzzy copies of video content consist of simple different embodiments of this invention, which in such cases are presented here in order to facilitate understanding. As a person skilled in the art will understand, many different embodiments of this technique will possess much greater complexity.

Fig. 1 is a simplified schematic representation of one of the non-limiting implementations of system 100 to embody this invention.

In some non-limiting embodiments of this invention, system 100, which implements the method, may include data transfer network 102, at least one computer device 104 to automatically identify fuzzy copies of video content (hereinafter computer device 104), at least one processor 106 included in computer device 104, and database 108 of original video content. System 100 may also include at least one search engine 110, which has access to video content database 112. System 100 may also include database 114 of fuzzy copies of video content, where fuzzy copies of original video content that have been identified during the performance of the described method may be saved.

Data transmission network 102 may be, for example, the Internet or any other computer network.

Computer device 104 may be an ordinary computer server. In the example embodiment of this invention, computer device 104 may be a Dell™ PowerEdge™ server, which uses the

Ubuntu Server operating system. Computer device 104 may be any other suitable hardware and/or application software and/or system software, or a combination thereof. In this non-limiting embodiment of this invention, computer device 104 is a single server. In other non-limiting embodiments of this invention, the functionality of the computer device may be divided among several computer devices, and, for example, may be provided by several servers.

In some non-limiting embodiments of this technique computer device 104 may be a personal computer (desktop, laptop, etc.).

Embodiments of computer device 104 are well known. Thus, it is sufficient to note that computer device 104 includes, among other things, a data transmission network interface (for example, a modem, network map, etc.) for two-way communication over a data transmission network (not shown); and at least one processor 106, connected to the data transmission network interface, where processor 106 is configured to perform various procedures, including those described below. With this goal, processor 106 can have access to machine-readable instructions stored on machine-readable medium (not shown), the execution of which initiates processor 106's implementation of various procedures described here. There is also a possible embodiment of this technique wherein at least part of the operations are performed by a graphics-processing unit (GPU, not shown).

In some embodiments of this invention, computer device 104 includes one or more databases 108, which are configured to store previously generated templates of original video content, where each template includes at least part of metadata, the vector of the sequence of scenes of original video content. There is a possible embodiment wherein the template also includes at least one frame from each scene. There is a possible embodiment wherein the identifier for the original video content file and/or the original video content file itself is stored in database 108 of original video content.

Although database 108 of original video content is represented as separate from computer device 104, to which it is connected using the corresponding line (not numbered), database 108 may be implemented as part of computer device 104 or will be available to computer device 104 by other means. Similarly, database 114 of fuzzy copies of video content, shown in Fig. 1 as separate from computer device 104, with which it is connected using an appropriate line (not numbered), may be implemented as part of computer device 104 or will be available to computer device 104 by other means. There is also a possible embodiment wherein database 108 of original

video content and database 114 of fuzzy copies of video content may be implemented on the same physical device or on separate devices.

There is a possible embodiment of system 100 wherein computer device 104 includes machine-readable medium (not shown), on which machine-readable instructions and/or database 108 and/or database 114 of fuzzy copies of video content may be stored in advance. Alternatively or additionally, computer device 104 may have access to machine-readable instructions and/or database 108 and/or database 114 of fuzzy copies of video content, which may be located on other hardware. Thus, as implemented, processor 106 of computer device 104's access to machine-readable instructions, database 108, and database 114 of fuzzy copies of video content, is not a limiting condition of this invention.

Processor 106 is configured to receive identifiers for video content. As a non-limiting example, receiving identifiers for video content may entail receiving a single uniform resource locator (URL). The source or sources of identifiers for video content may be diverse.

There is a possible embodiment of this technique wherein processor 106 is configured to obtain identifiers for video content at least partially from an external source via data network 102. For example, an external source might be a search engine 110 configured to perform vertical searches of video content that has access to video content database 112 and/or a video hosting server (not shown) on which video content is posted, etc. Moreover, computer device 104 may have access to video content database 112 via data transmission network 102. Alternatively or additionally, an external source may be a web resource, including identifiers for video content.

Thus, exactly how and by what principle identifiers for video content are grouped on an external source, in particular, on video content database 112, are not limiting conditions. There is a possible embodiment wherein database 112 is preloaded with classifications of video content according to one or more parameters.

In the example embodiment of this invention, search engine 110 may be a Dell™ PowerEdge™ server, which uses the Ubuntu Server operating system. Search engine 110 may be any other suitable hardware and/or application software and/or system software, or a combination thereof. In this non-limiting embodiment of this invention, search engine 110 is a single server. In other non-limiting embodiments of this invention, the functionality of search engine 110 may be divided, and may be provided by several servers.

Embodiments of search engine 110 are well known. Thus, search engine 110 includes, among other things, a data transmission network interface (for example, a modem, network card, etc.) for two-way communication over data transmission network 102; and a processor (not shown), connected to the data transmission network interface, where the processor is configured to search video content on data transmission network 102 and store it in database 112. With this goal, the search engine processor can have access to relevant machine-readable instructions stored on a machine-readable medium (not shown).

The process of populating and saving video content database 112 is generally known as "data crawling". The search engine 110 is configured to "visit" various web resources, websites, and webpages, including search engines, video hosting sites, social networks, etc. that are accessible via data transmission network 102, and save video content identifiers in video content database 112 based on one or more predefined parameters. As a non-limiting example, server 110 can scan data transmission network 102 and save identifiers for video content from, for example, web pages that include certain keywords in one or more languages, or other accompanying content, such as, for example, movie posters, trailers, film descriptions, etc. The search and selection parameters for video content, as well as specific web resources on which the search may be performed, are not specifically limited. As an example, a search may be conducted on search engines, in particular, Yandex (www.yandex.ru), Google (www.google.com), etc. In this regard, a search query to search for video content that potentially violates copyrights may be "free download in high quality" or "watch for free." Search queries may also include names or parts of names of specific original video content, for example, names of recently released films or TV shows; examples of such search queries might be "title_of_movie stream," "title_of_movie download" etc. Algorithms to search for information on the Internet are well known; one or several different approaches may be used for the purposes of this invention.

The found and selected identifiers for video content that potentially may violate copyrights are saved in video content database 112/ Then they may be received by the processor of computer device 104 via data transmission network 102 from search engine 110 in order to perform the method of automatically identifying fuzzy copies of video content, which will be described in more detail below with reference to Fig. 3.

There is a possible embodiment of this invention, wherein processor 106 of computer device 104 is configured to obtain identifiers for video content directly from video content database 112.

In this regard, video content database 112 may be implemented as part of computer device 104 or be accessible to computer device 104 by other means.

There is a possible embodiment of this technique wherein processor 106 is configured to scan network 102 in order to obtain identifiers for video content and in this way computer device 104 can perform the functionality of search engine 110 described above.

There is a possible embodiment of this technique wherein processor 106 of computer device 104 is configured to create templates of original video content and thereby generate and populate database 108 of original video content. Alternatively or additionally, the creation of templates of original video content and generation of database 108 of original video content may also be executed by one or more different computer devices (not shown).

With reference to Fig. 2, method 200 for creating a template of original video content in database 108 of original video content will be discussed in more detail. Method 200 may be performed on computer device 104 and, specifically, by processor 106 of computer device 104 as in the non-limiting embodiment of system 100 shown in Fig. 1.

Stage 202 - Receiving identifiers for at least part of the original video content

Method 200 begins at step 202, where computer device 104 receives identifiers for at least part of the original video content. Identifiers for at least part of the original video content may be video content files themselves, or, for example, links to video content, URLs that are accessible to computer device 104 via data transmission network 102 or another network (not shown). The source or sources of original video content may be diverse, as an example, but this is not a limitation; computer device 104 may receive identifiers for at least part of the original video content via data transmission network 102 from another computer device (not shown) connected to data transmission network 102 or via machine-readable medium (not shown), which may be connected to computer device 104. The range of persons from whom identifiers for at least part of the original video content may be received is not specifically limited. For example, this may be the copyright owner, a representative of the copyright owner, law enforcement agencies, or other organizations involved in protecting copyrights, etc. who wish to detect and suppress the distribution of pirated copies of original video content.

At least part of the original video content may be, for example, full video content or a fragment thereof (in particular a fragment of a movie, TV show, etc.). In this case, the minimum

length of such a fragment may be preset for video content of various types to ensure the most accurate identification of fuzzy copies of original video content. For a music video or other video content that features frequent scene changes, the minimum length may be, for example, 1 minute, for a film 10 to 15 minutes, for a TV show 5 to 8 minutes, etc. These minimum lengths are not limiting and are used only for purposes of examples. It will be obvious to a person skilled in the art, considering the following description of stages of method 300 for automatically identifying fuzzy copies of video content, that the more complete the received (at least) portion of original video content, the greater the quantity of different fuzzy copies of video content that may be discovered.

Then method 200 transits to step 204.

Stage 204 - Extracting at least part of the metadata of the original video content

During step 204, processor 106 of computer device 104 extracts at least part of the metadata of the original video content. Video content metadata may include, in particular, name, description, tags, length, language, comments, file format, file size, frame size, frame rate, subtitles, date of creation, type of video content, place and date of filming, GPS coordinates, etc. Methods to automatically extract metadata from a video file are well known and are not the subject of this patent.

Then method 200 transits to step 206.

Stage 206 - Extracting at least part of the frames from the sequence of frames of original video content

During step 206, processor 106 of computer device 104 extracts at least part of the frames from the sequence of frames of the original video content. The video stream of the original video content is broken down into separate frames. In order to decrease the load on computing resources and increase processing speed, there is a possible embodiment wherein frames are extracted at a sampling rate of D frames per minute, where $D > 1$. As a non-limiting example, D may be equal to 60, in which case, when the frame rate is 30 frames per second, every 30th frame would be extracted. A sequence of frames with step 1 or greater than 1 is obtained depending on the value of D . The value of D may be preset, or set automatically by a function, for example, depending on the size and length of video content and/or other parameters that can be obtained during step 204, extracting original video content metadata.

There is a possible embodiment of method 200 wherein step 204, extracting at least part of the metadata of the original video content, and step 206, extracting at least part of the frames of the sequence of frames of the original video content, can be performed in parallel after step 202, receiving identifiers for at least part of the original video content. In another non-limiting embodiment of method 200, step 206, extracting at least part of the metadata of the original video content, can be performed after step 204, extracting at least part of the frames of the sequence of frames of the original video content.

Then method 200 transits to step 208.

Stage 208 - Identifying a sequence of scenes

During step 208, processor 106 of computer device 104 identifies a sequence of scenes of the original video content. Identifying a sequence of scenes could be performed, for example, by comparing adjacent frames from the received sequence of frames. In the case where the sampling rate $D > 1$, the comparison of frames is performed with the appropriate sampling rate. For example, if D is equal to 60 frames per minute and the frame rate is 30 frames per second, then the sequence of frames from the video stream to be compared is a sampling of every 30th frame (the 30th frame is compared with the 60th, 60th with 90th, 90th with 120th, etc.). There is a possible embodiment wherein, in order to lower the burden on computing power, the comparison is made not of the whole frame, but one or several parts of it, for example, one or several fields in a central part of the frame.

In order to compare adjacent frames, a hash sum of the comparative frames or one or several parts of the comparative frames may be extracted from the received sequence of frames. Then processor 106 computes the Hamming distance $P(x)$ between hash sums of comparative frames. When the Hamming distance between hash sums of comparative frames or parts of frames exceeds a predefined threshold, then it is defined as a scene change. The predefined threshold may be, for example, 15-30%. If the similarity of hash sums exceeds the threshold value, then they are considered to be frames in the same scene.

Thus, comparing pairs of frames or pairs of parts of frames allows the identification of sequences of scenes in original video content and the length of each of the scenes in the sequence of scenes.

Then method 200 transits to step 210.

Stage 210 - Creating a vector of the sequence of scenes

During step 210, processor 106 of the computer device creates a vector of the sequence of scenes of original video content, the vector can contain data on the length of each scene that was identified during step 208 of method 200. The length of scenes can be expressed, for example, in seconds or minutes, or in absolute or relative quantity of frames per scene. Absolute quantity of frames per scene means the number of frames of the video stream, and relative quantity of frames per scene means the number of extracted frames, taking into consideration the sampling rate. For example, if every 30th frame is extracted, then the relative values of the vector might contain the quantity of extracted frames (the relative value would be 20 when the absolute value is 600).

In a non-limiting example of the embodiment, the vector of original video content containing nine scenes may look as follows:

$V_1=(20; 230; 430; 150; 340; 230; 340; 560; 105)$

Then method 200 transits to step 212.

Stage 212 - Generating and saving in a database a template of the original video content, which includes at least part of the metadata, and the vector of the sequence of scenes of the original video content

During step 212, processor 106 of the computer device generates a template of original video content, the template includes at least part of the metadata of the original video content obtained during step 204, as well as the vector of the sequence of scenes obtained at step 210. Then the generated template of original video content is saved in database 108 of original video content.

There is a possible embodiment of the method wherein at least one frame or part of a frame of each scene is extracted and saved in the template of original video content.

The template of video content is, in a general sense, a structured set of data associated with the specific original video content. Creating a template helps to reduce the burden on the system's computing resources substantially when searching for found video content and comparing it with the original video content to identify fuzzy copies, which will be shown in more detail in the description of method 300.

After step 212, method 200 may end.

Database 108 of original video content may be populated according to the claimed method 200 described above by one or several different computer devices.

Next, referring to Fig.1 and Fig.2, method 300 for automatically identifying fuzzy copies of video content will be described. Method 300 may be performed on computer device 104 and, specifically, by processor 106 of computer device 104 as in the non-limiting embodiment of system 100 shown in Fig. 1.

Stage 302 - Receiving an identifier for video content

Method 300 begins at step 302, where computer device 104 receives an identifier for video content. The identifier may be a video content file itself, or, for example, links to video content, URLs that are accessible to computer device 104 via data transmission network 102 or another network (not shown).

The identifier may be received in particular from search engine 110 or directly from database 112 of video content.

There is a possible embodiment of method 300 wherein, in parallel before step 302, receiving the identifier, processor 106 of computer device 104 performs a search of video content on a data transmission network, where the search is performed according to one or more of the following parameters: keywords in the file name, keywords in the description, accompanying images, file metadata.

The search criteria are not specifically limited. There is a possible embodiment of this technique wherein the search is performed according to specific parameters of a given original video content, fuzzy copies of which (including identical copies) are to be found. Alternatively or additionally, the search may be performed over all video content that is accessible via data transmission network 102 or on a specific web resource. So, as a non-limiting example, during step 302 processor 106 of computer device 104 may receive identifiers for any new video content published on the social network VKontakte (www.vk.com) in order to identify fuzzy copies of original video content, templates of which were previously generated and saved in database 108 of original video content through method 200 described above. A further verification may be performed in relation to one or more different templates of original video content simultaneously.

Then method 300 transits to step 304.

Stage 304 - Extracting each Nth frame from the video stream of received video content to obtain a sequence of frames, where $N > 1$;

During step 304, processor 106 of computer device 104 extracts each Nth frame of the video stream of received video content and obtains a sequence of frames. The video stream of the original video content is broken down into separate frames. In order to decrease the load on computing resources and increase the speed of automatically identifying fuzzy copies of video content, frames are extracted with a sampling rate of N frames per minute, where $N > 1$. As a non-limiting example, N may be equal to 60, in which case, when the frame rate is 30 frames per second, every 30th frame would be extracted. A sequence of frames with step greater than 1 is obtained depending on the value of N. The value of N may be preset, or set automatically by a function, for example, depending on the size and length of video content and/or other parameters.

For video content characterized by frequent scene changes, for example, video clips, the value of N may be set as equal to, for example, 120. For less dynamic video content characterized by rare scene changes (for example, video lectures), the value of N can be set, for example, to 2000. The converse embodiment is also possible, where the value of N for less dynamic video content is less than the value of N for more dynamic content, or generally does not depend on the type of video content and other parameters, but has a constant value, for example, 60. The above-described values of N are not limiting and given only as illustrative examples to facilitate understanding.

Then method 300 transits to step 306.

Stage 306 - Identifying the sequence of scenes by comparing at least fragments of adjacent frames from the obtained sequence of frames;

During step 306, processor 106 of computer device 104 identifies a sequence of scenes of video content. Identifying a sequence of scenes could be done, for example, by comparing adjacent frames from the received sequence of frames with step N. So, for example, if N is equal to 60 frames per minute and the frame rate is 30 frames per second, then the sequence of frames from the video stream to be compared is a sampling of every 30th frame (the 30th frame is compared with the 60th, 60th with 90th, 90th with 120th, etc.). There is a possible embodiment wherein, in order to lower the burden on computing resources, a comparison is made of, not the whole frame, but one or more parts thereof (fragments), for example, one or more fields, for example in the central part of the frame, the upper right portion of the frame, the lower left portion

of the frame, etc. Embodiments of identifying fragments are not specifically limited and may depend on the type of video content and the most dynamic fragments of the frame. They may be identified in advance on the basis of statistical data or basic principles of frame construction (composition) that are used by operators during shooting.

There is a possible embodiment wherein, during the step of identifying scene changes (scene boundaries), a comparison of extracted hash sums of at least fragments of adjacent frames is performed in response to the Hamming distance between hash sums of adjacent frames exceeds a predefined threshold; the boundary of the scene (scene change) is identified. In order to compare adjacent frames, a hash sum of the comparative frames or one or several parts of the comparative frames may be extracted from the received sequence of frames. Then processor 106 computes the Hamming distance $P(x)$ between hash sums of comparative frames. In response to the Hamming distance between hash sums of comparative frames or parts of frames exceeds a predefined threshold, then it is defined as a scene change. The predefined threshold may be, for example, 15-30%. If the similarity of hash sums exceeds the threshold value, then they are considered to be frames in the same scene.

Thus, comparing pairs of frames or pairs of parts (fragments) of frames allows the identification of sequences of scenes in received video content and the length of each of the scenes in the sequence of scenes.

There is a possible embodiment wherein steps 304 and 306 of the method are performed in parallel.

Then method 300 transits to step 308.

Stage 308 - Creating a vector of the sequence of scenes of the received video content;

During step 308 processor 106 of computer device 104 creates a vector of the sequence of scenes of received video content, which can contain data on the length of each scene that was identified during step 306 of method 300. The length of scenes can be expressed, for example, in seconds or minutes, or in absolute or relative quantity of frames per scene. Absolute quantity of frames per scene means the number of frames of the video stream, and relative quantity of frames per scene means the number of extracted frames, taking into consideration step N. For

example, if every 30th frame is extracted, then the relative values of the vector might contain the quantity of extracted frames (the relative value would be 20 when the absolute value is 600).

There is a possible embodiment of method 30 wherein the vector of the sequence of scenes of received video content is a vector of at least two disjoint subsets of scenes of received video content.

Some illustrative examples of vectors of received video content, for the purpose of further clarifying the description of method 300, are given below:

Example 1: $V_2=(20; 230; 430; 150; 120; 340; 230; 180; 340; 560; 105)$

Example 2: $V_3=(230; 430; 150; 120; 340; 230; 180; 340; 560)$

Example 3: $V_4=(33; 230; 430; 150; 340; 230; 340; 560; 105; 700; 388; 222; 250; 92; 170; 800; 448; 590; 75; 33; 620)$

Then method 300 transits to step 310.

Stage 310 - Comparing the vector of the sequence of scenes of obtained video content with the vector of the sequence of scenes of at least one template of original video content from the previously generated database of original video content

During step 310, processor 106 of computer device 104 compares the vectors of the sequence of scenes of received video content with the vector of the sequence of scenes of at least one template of original video content from a previously generated database 108 of original video content.

Before comparison, the vectors may be converted into a common format, for example, if the vector of original video content was created on the basis of the total number of frames in the video stream of original video content, but the vector of received video content was created on the basis of the number of frames of a sequence with step N.

In these illustrative examples the vector of original video content $V_1=(20; 230; 430; 150; 340; 230; 340; 560; 105)$ and examples of vectors of received video content (V_2 , V_3 , V_4) are converted to a common format.

Vector differences are determined from the results of the comparison (differences are shown in bold):

Example 1: $V_2=(20; 230; 430; 150; \mathbf{120}; 340; 230; \mathbf{180}; 340; 560; 105)$

Example 2: $V_3=(230; 430; 150; \mathbf{120}; 340; 230; \mathbf{180}; 340; 560)$

Example 3: $V_4=(\mathbf{33}; 230; 430; 150; 340; 230; 340; 560; 105; \mathbf{700; 388; 222; 250; 92; 170; 800; 448; 590; 75; 33; 620})$

There is a possible embodiment wherein the permissible error rate for the comparison of vectors is preset. For example, the error rate could be 5-10%. Accordingly, when comparing vectors with scene lengths, for example, of 200 and 220, they may be equated to one another and considered equal. This error may arise, for example, by deleting part of the frames in the fuzzy copy, or by selecting step N of received video content greater or lower than step D of the sequence of frames of original video content. As will be understood by persons skilled in the art, other factors can also affect the error rate, which can be offset by the predefined acceptable error rate.

There is a possible embodiment of method 300 wherein, during step 310, when comparing the vector of the sequence of scenes of received video content with the vector of the sequence of scenes of at least one template of original video content, a comparison is also performed of at least one frame from the scene of received video content with at least one frame from the corresponding scene of original video content, forming at least one comparative pair of frames. There is a possible embodiment of method 300 wherein hash sums are extracted and the hash sums of a comparative pair of frames are compared. In this case, the consistency of the comparison result is complicated by the fact that hashes of comparative images differ by a small number of bits as compared with the size of the image, which facilitates finding the Hamming distance between image data hashes.

Comparing separate frames facilitates carrying out a spot check that one or several scenes match, in order to increase the accuracy of the claimed method.

Then method 300 transits to step 312.

Stage 312 - Identifying a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold

During step 310, processor 106 of computer device 104 identifies a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold.

There is a possible embodiment of method 300 wherein, the vector of at least two disjoint subsets of scenes of received video content is identified as the vector of a fuzzy copy of the original video content in response to it coincides with at least part of the vector of the sequence of scenes of original video content. As shown in the illustrative example of vectors, $V_2=(20; 230; 430; 150; \mathbf{120}; 340; 230; \mathbf{180}; 340; 560; 105)$ and $V_3=(230; 430; 150; \mathbf{120}; 340; 230; 340; 560)$

In Example 2 of vector V_3 , between the sequence of scenes with length 230; 430; 150; and scenes of length 340; 230; 340; 560 that match the length of scenes of original video content there is a scene with length 120 that is missing in the vector of original video content.

In Example 1 vector $V_2=(20; 230; 430; 150; \mathbf{120}; 340; 230; \mathbf{180}; 340; 560; 105)$ has two scenes of length 120 and 180 that are missing in the vector of the original video content $V_1=(20; 230; 430; 150; 340; 230; 340; 560; 105)$.

In response to some different sequences of scenes coincide, received video content with vector $V_2=(20; 230; 430; 150; \mathbf{120}; 340; 230; \mathbf{180}; 340; 560; 105)$ and received video content with vector $V_3=(230; 430; 150; \mathbf{120}; 340; 230; 340; 560)$ can be identified as a fuzzy copy of the original video content with vector $V_1=(20; 230; 430; 150; 340; 230; 340; 560; 105)$. The scenes of length 120 and 180, in this illustrative example, can be, for example, an advertisement that was embedded into the original video content when the fuzzy copy was created.

There is a possible embodiment of method 300 wherein the length of the fuzzy copy differs from the length of the original video content.

In illustrative Examples 1 and 2 (vectors V_2 V_3), the lengths of fuzzy copies differ from the length of the original video content due to embedded advertising, as well as in Example 2 (vector V_3) due to deleting the first scene with length 20 and the last scene (for example, credits) with length 105 relative to the original video content. In Example 3 (vector $V_4=(\mathbf{33}; 230; 430; 150; 340; 230; 340; 560; 105; \mathbf{700; 388; 222; 250; 92; 170; 800; 448; 590; 75; 33; 620;})$), the length of the fuzzy copy is substantially increased and can match, for example, two combined vectors of the original video content. This is possible, for example, when two episodes of original video content are combined in a single fuzzy copy.

As will be understood to persons skilled in the art, other installation techniques are also possible, which can be used to create fuzzy copies of video content based on one or more copies of original video content.

After step 312, method 300 may end.

There is a possible embodiment of method 300 wherein, after identifying a fuzzy copy of video content, processor 106 saves the identifier for the video content in database 114 of fuzzy copies. Then the obtained data can be sent to the copyright holder, and measures can also be taken to automatically block the detected fuzzy copy.

There is a possible embodiment of this method wherein processor 106 is configured to perform an additional search for fuzzy copies of video content containing metadata identical to the metadata of the identified fuzzy copy of video content when the metadata of an identified fuzzy copy of video content differs at least partially from the metadata of the original video content. Next, the identified additional fuzzy copies can be checked using method 300 described above, or immediately saved in database 114 of fuzzy copies.

The claimed method to automatically identify fuzzy copies of video content ensures high speed and accuracy, as well as a substantial reduction in the burden on computing resources.

Modifications and improvements to the above-described embodiments of this technique will be clear to persons skilled in the art. The preceding description was given only as an example and does not impose any limitations. Thus, the scope of this technique is limited only by the scope of the appended claims.

CLAIMS

1. A method to automatically identify fuzzy video content copies, which is performed on a computer device that has access to a previously generated database of original video content; the method comprising:
 - receiving an identifier for video content;
 - extracting each Nth frame from the video stream of received video content to obtain a sequence of frames, where $N > 1$;
 - identifying the sequence of scenes by comparing at least fragments of adjacent frames from the obtained sequence of frames;
 - creating a vector of the sequence of scenes of the received video content;
 - comparing the vector of the sequence of scenes of obtained video content with the vector of the sequence of scenes of at least one template of original video content from the previously generated database of original video content;
 - identifying a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold.
2. The method of claim 1, wherein prior to the step of receiving the identifier, a search is performed for video content on a data transmission network, the search is performed according to at least one of the following parameters: keywords in the file name, keywords in the description, accompanying images, file metadata.
3. The method of claim 1, wherein the value N is predefined as a function of the length of video content.
4. The method of claim 1, wherein, during the step of identifying scenes, a comparison of extracted hash sums of at least fragments of adjacent frames is performed in response to the Hamming distance between hash sums of adjacent frames exceeds a predefined threshold; the scene change is identified.
5. The method of claim 1, wherein a vector of the sequence of scenes of received video content is created based on the total number of frames in the scene.

6. The method of claim 1, wherein the vector of the sequence of scenes of received video content is a vector of at least two disjoint subsets of scenes of received video content.
7. The method of claim 6, wherein, in response to a coincidence of the vector of at least two disjoint subsets of scenes of received video content and at least part of the vector of the sequence of scenes of original video content, the received video content is identified as a fuzzy copy of the original video content.
8. The method of claim 1, wherein, during the step of comparing the vector of the sequence of scenes of received video content with the vector of the sequence of scenes of at least one template of original video content, a comparison is also performed of at least one frame from the scene of received video content with at least one frame from the corresponding scene of original video content, forming at least one comparative pair of frames.
9. The method of claim 8, wherein hash sums are extracted and hash sums of comparative pairs of frames are compared.
10. The method of claim 1, wherein, in response to the metadata of an identified fuzzy copy of video content differs at least partially from the metadata of the original video content, an additional search for fuzzy copies of video content containing metadata identical to the metadata of the identified fuzzy copy of video content is performed.
11. The method of claim 1, wherein the length of the fuzzy copy differs from the length of original video content.
12. The method of claim 1, wherein, after identifying a fuzzy copy of video content, the identifier for the video content is saved in a database of fuzzy copies.
13. **The method for creating a template of original video content** performed on a computer device, the method comprising:
 - receiving identifiers for at least part of the original video content;
 - extracting at least part of the metadata of the original video content;
 - extracting at least part of the frames from the sequence of frames of original video content;
 - identifying a sequence of scenes;

creating a vector of the sequence of scenes;

generating and saving in a database a template of the original video content that includes at least part of the metadata, and the vector of the sequence of scenes of the original video content.

14. The method of claim 13, wherein at least one frame from each scene is saved in the template of the original video content.

15. **A computer device that automatically identifies fuzzy copies of video content**, has access to a previously generated database of original video content, and is connected to a data transmission network; the computer device includes a communication interface and at least one computer processor that is functionally connected to the communication interface, configured to:

receive an identifier for video content;

extract each Nth frame from the video stream of received video content to obtain a sequence of frames, where $N > 1$;

identify the sequence of scenes by comparing at least fragments of adjacent frames from the obtained sequence of frames;

create a vector of the sequence of scenes of the received video content;

compare the vector of the sequence of scenes of obtained video content with the vector of the sequence of scenes of at least one template of original video content from the previously generated database of original video content;

identify a fuzzy copy of the original video content in response to the sequence of scenes of original video content coincides with the scenes of received video content to a degree that exceeds a predefined threshold.

16. The device of claim 15, wherein the processor is configured to perform a search for video content on a data transmission network, the search is performed according to at least one of the following parameters: keywords in the file name, keywords in the description, accompanying images, file metadata.

17. The device of claim 15, wherein the processor is configured to perform a comparison of extracted hash sums of at least fragments of adjacent frames and to identify the boundary of the scene in response to the Hamming distance between hash sums of adjacent frames exceeds a predefined threshold.
18. The device of claim 15, wherein the processor is configured to create a vector of the sequences of scenes of received video content based on the total number of frames in the scene.
19. The device of claim 15, wherein the processor is configured to create a vector of the sequence of scenes of received video content, the vector contains of at least two disjoint subsets of scenes of received video content.
20. The device of claim 19, wherein the processor is configured to identify a fuzzy copy of the original video content in response to the vector of at least two disjoint subsets of scenes of received video content coincides with at least part of the vector of the sequence of scenes of the original video content.
21. The device of claim 15, wherein the processor is configured to perform a comparison of at least one frame from the scene of received video content and at least one frame from the corresponding scene of original video content and generate at least one comparative pair of frames.
22. The device of claim 21, wherein the processor is configured to extract hash sums and compare the hash sums of a comparative pair of frames.
23. The device of claim 15, wherein the processor is configured to perform a supplemental search for fuzzy copies of video content that contain metadata identical to the metadata of an identified fuzzy copy of video content in response to the metadata of the identified fuzzy copy of video content differs at least partially from the metadata of the original video content.
24. The device of claim 15, wherein the device also has access to a database of fuzzy copies, where the processor is configured to save identifiers for video content in the database of fuzzy copies.

25. The device of claim 15, wherein the processor is also configured to:
- receive identifiers for at least part of the original video content;
 - extract at least part of the metadata of the original video content;
 - extract at least part of the frames from the sequence of frames of original video content;
 - identify a sequence of scenes;
 - create a vector of the sequence of scenes;
 - generate and save in a database a template of the original video content, including at least part of the metadata, and the vector of the sequence of scenes of the original video content.
26. The device of claim 25, wherein the processor is also configured to save at least one frame from each scene in a template of the original video content.