Analysis of Streaming Video Content and Generation Relevant Contextual Advertising

Tetiana Kovaliuk[1][0000-0002-1383-1589], Nataliya Kobets[2][0000-0003-4266-9741],
Grigorii Shekhet[3] and Tamara Tielysheva[3'][0000-0001-5254-3371]

[1]Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska str, Kyiv, 01601 Ukraine
[2]Borys Grinchenko Kyiv University, 18/2 Bulvarno-Kudriavska Str, Kyiv, 04053, Ukraine
[3]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",
37, Prospekt Peremohy, Kyiv 03056, Ukraine
tetyana.kovalyuk@gmail.com, nmkobets@gmail.com,
a.gambit.gregory@gmail.com, telyshevatamara@gmail.com

Abstract. The article discusses the task of searching and selecting advertisements that match the content of the video. The problem is urgent due to the large amount of video and advertising content on the Internet. The authors set the task to develop information technology to solve this problem. The methods of recognizing the contours of objects using the Prewitt operator and searching for similar objects in streaming video based on perceptual hash functions are considered. An algorithm for recognizing speech from an audio track based on Mel-Frequency Cepstral Coefficients and an algorithm for searching for keywords in the text from a sound track based on TF-IDF are developed. The considered algorithms are used to determine contextual advertising for video content. The presented control example, the results of which indicate the efficiency of the algorithm.

Keywords: streaming video content, computer vision, classifications and recognition speech, contextual advertising.

1 Introduction

Every now and then, terabytes of digital information, including photos and video content, appear in the world. Video is becoming more widespread and video collections of public services and personal archives are growing. Cisco predicts that by 2020, 82% of all consumer web traffic will be video. Video is the most popular trend in the presentation of information, as evidenced by such statistics [1]:

- almost 82% of Twitter users watch video content on this platform;
- people spend over 500 million hours watching videos on YouTube around the world;
- users watch about 10 billion videos daily on Snap chat;
- 92% of mobile users are actively involved in sharing videos with others;
- 87% of online marketers use video content;

The volume of video content proves that video search technology will become the core technology of the new digital world. Video data must be processed and their content must be indexed for information retrieval. For processing video data, computer vision methods are used.

There are a variety of online movie theaters and Internet video services. While watching videos, online movie theaters and video services offer users a variety of advertising that does not correspond to the content of the video and therefore not relevant. Such advertising is not interesting to the viewer, and therefore does not generate expected revenue for advertisers, which in turn casts doubt on its appropriateness. Therefore, search advertising, the content of which corresponds to the content of the video and may interest the viewer, is an urgent task.

The purpose of this work is to increase the efficiency and usability of algorithms for real-time video stream analysis and finding advertisements in accordance with the video stream content.

2 Analysis of Related Research and Publications

The task of analyzing streaming video content is one of the most pressing real-time data analytics tasks. A variety of approaches and algorithms are used to solve it, many of which are based on basic data processing and analysis methods, while others can be highly specialized in video and graphic information analysis. Among them are algorithms that analyze video through semantic analysis [2, 3], by complicating the encoding format or exploring its metadata [4], searching for major video objects by tracking key frames [5], or using computer vision methods [6], trying to find content by analyzing audio video tracks [7]. In [8], the authors describe the semantics of multimodal video and segmental interest indexing at a conceptual level. In [9] on basic semantic analysis, the author argues that knowledge about video structure can be used not only as a means to improve the performance of content analysis, but also to extract features that convey semantic information about the content of video.

Instead, most of the approaches considered are slow, non-scalable, and unable to work steadily in real time. A study of the literature devoted to the semantic analysis of the video stream, the search for key fragments, the classification and recognition of various objects in the video stream showed that most of the considered algorithms do not work well in real time or are not capable of it. The proposed algorithms cannot be scaled on problems of greater dimension or in real practice. Computer vision methods considered in the literature require large data sets, and often lose in speed to statistical algorithms on large images in real time. Most of the approaches considered in the literature neglect the analysis of the audio track and speech in the video stream, or do not take into account that the audio information does not always completely coincide with the image on the screen.

The ideas embedded in the approaches of searching for key objects, analyzing metadata, audio and video content are very interesting and require refinement to solve the problem.

3 Problem Statement

It is necessary to create algorithmic software that accepts streaming video and finds advertisements that match the content of the video. To do this, you need to solve the problem of finding key objects in the video stream to determine its content.

The main tasks to solve this problem are as follows:

1. Consider the theoretical aspects of the problem of video analysis. For this purpose, it is necessary to analyze the current state and features of streaming video analysis, evaluate the feasibility of using these approaches.
2. Solve the problem of real-time video stream analysis and advertising search based on video content. For this purpose, it is necessary to develop a formal statement of the problem; to be acquainted with modern approaches to solving this problem; to evaluate the potential advantages and disadvantages of the proposed methods.
3. Based on a detailed analysis of the main algorithms, develop an algorithm that takes into account the features and limitations of the task of analyzing video stream in real time and finding advertisements in accordance with video content.
4. Implement the developed algorithm in the form of software to evaluate the reliability of its results

4 Stages of Problem Solving

Real-time video content analysis takes into account the following factors: metadata, video frames, and video audio. The algorithm developed divides the video stream into parts, after which the advertisement is displayed, and in parallel analyzes the content for each of them in four stages.

In the first stage, the video is storyboarded and the frames obtained are analyzed by searching for the main objects using image and computer vision algorithms.

The second stage consists of speech analyzing and recognizing on the audio track and searching for keywords in the resulting text.

In the third step, the algorithm analyzes the video metadata and finds the keywords in them.

In the fourth stage, we search for ads that are as close as possible to the content of the video stream, based on the data received about the objects.

5 Pattern Recognition Algorithm from Video Stream Frames

Search and recognition of the main objects in the video frames most often consist of two stages. At the first stage, image filtering is performed, and at the second, its direct analysis. There are computer vision methods, which facilitate these steps, but they work slowly and require optimization.

That is why the key idea of the developed algorithm is to optimize the analysis process using computer vision methods. To do this, a search for common objects is performed and only the fragments found by frames are analyzed.

Pattern recognition algorithm from video stream frames is based on the following methods:

1. Object contour recognition using Prewitt algorithm.
2. Find perceptual hashes of objects in frames using the pHash algorithm.
3. Find common objects by comparing hashes using the Hamming distance.
4. Recognize common objects using the computer vision method.

5.1 Object Contour Recognition Method Using Prewitt Operator

To optimize the recognition of objects in frames from video, you should find the fragments of objects in images by defining their contour.

Methods for determining the contours of objects are based on the property of the luminance signal discontinuity. The procedure for determine contours includes two stages. Initially, the image is detected brightness variations that form the contours, and then the results are compared with a threshold. If the detection result exceeds the threshold, then it is considered that this image pixel belongs to the contour and it is assigned the brightness value of the contour, otherwise it is assigned the brightness value of the background. As a result of such image processing, its contour analogue is obtained. The most common way to find luminance signal discontinuity is to image processing using a sliding mask that acts as a filter. The mask is a square matrix 3×3 corresponding to the pixel group of the original image [10].

The process is based on moving the filter mask from one image point to another. At each point (x, y), the filter response is calculated. In the case of linear spatial filtering, the response is given by the sum of the product of the filter coefficients by the corresponding pixel values in the area covered by the filter mask. For 3 × 3 mask, the linear filtration result R at the image point (x, y) is defined as the sum of the products of the mask coefficients $w(i, j), i, j \in [-1, 0, 1]$ per pixel value $f(x, y)$ directly below the mask (1):

$$\begin{aligned} R = w(-1,-1)\, f(x-1, y-1) = w(-1,0)\, f(x-1, y) + ... + w(0,0)\, f(x, y) + ... \\ + w(1,0)\, f(x+1, y) + w(1,1)\, f(x+1, y+1) \end{aligned} \tag{1}$$

The mask coefficients are set in relative coordinates, the coefficient $w(0,0)$ at value $f(x, y)$ indicates that the mask is centered at a point (x, y).

Discrete analogs of first and second order derivatives are used to detect brightness differences. We will consider one-dimensional derivatives. The first derivative of a one-dimensional function $f(x, y)$ is defined as the difference of values of adjacent image elements (2):

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{2}$$

The second derivative is defined as the difference of adjacent values of the first derivative:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{3}$$

The calculation of the first derivative of the digital image is based on discrete approximations of the two-dimensional gradient. By definition, the image gradient $f(x,y)$ at a point (x,y) is a vector (4):

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \Big/ \frac{\partial f}{\partial y} \end{bmatrix} \tag{4}$$

It is known that the direction of the gradient vector coincides with the direction of the maximum rate of change of the function $f(x,y)$ at a point (x,y). Consider the module of this vector (5):

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} \tag{5}$$

This value is equal to the value of the maximum rate of change of function $f(x,y)$ at a point (x,y). The maximum is reached in the direction of the vector ∇f, which is called the gradient. The direction of the gradient vector is determined by the angle between the vector ∇f at the point (x,y) and the axis X and is equal (6):

$$\alpha(x,y) = arctg\left(G_y / G_x\right) \tag{6}$$

From here it is easy to find the direction of the contour at a point (x,y) that is perpendicular to the direction of the gradient vector at this point. The gradient of the image can be calculated by determining the values of the partial derivatives $\partial f / \partial x$ и $\partial f / \partial y$ for each point.

Let the 3 × 3 region represent the brightness values in the vicinity of some image element (7):

$$\begin{pmatrix} z1 & z2 & z3 \\ z4 & z5 & z6 \\ z7 & z8 & z9 \end{pmatrix} \tag{7}$$

One of the simplest ways to find the first partial derivatives at a point z_5 is to use the Prewitt cross gradient operator.

$$\begin{aligned} G_x &= (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \\ G_y &= (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \end{aligned} \tag{8}$$

In these formulas, the difference between the sums over the upper and lower rows of the 3 × 3 neighborhood is the approximate value of the axis X derivative, and the difference between the sums over the first and last columns of this neighborhood is the axis y derivative. To implement these formulas, we use the Prewitt operator described by masks (9):

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix},\ G_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{9}$$

Using the Prewitt operator in the gradient formula, we get images from frames that show only the contours of the objects. These contours allow highlighting the areas in the frames where the desired objects are located.

5.2 Search Similar Objects in Streaming Video Based on Perceptual Hash Functions

To search similar objects on the found contours, you can use frequency analysis of images. Given the dependence of image brightness on horizontal and vertical coordinates, the image can be represented as a two-dimensional non-periodic signal. In images, high frequencies provide detail, while low frequencies determine the structure of the image. Thus, a high-resolution detailed image contains a large number of high frequencies. A smaller copy of such an image almost entirely consists of low frequencies

Let's consider use of perceptual hash functions at the decision of a problem of search of similar images. Perceptive hash functions extract certain features from a multimedia object and calculate a hash based on these features. The obtained hash can be considered as an image imprint. The task of comparing images is to calculate the hash values of these images and calculate the Hamming distance between them. The smaller is the Hamming distance, the more similar are the image data.

Consider a hash based on Discrete Cosine Transform (DCT), which is a type of Fourier transform. The graphic image can be considered as set of spatial waves, and axes X and Y coincide with width and height of a picture, and on an axis Z value of color of corresponding pixel of the image is postponed. The Discrete Cosine Transformation allows to pass from spatial representation of the image to its spectral representation and back.

Let's consider the most widespread variant of DCT-2 according [11]:

$$X[k]-\sum_{n=0}^{N-1} x[n]\cos\left(\frac{(2n+1)k\pi}{2N}\right), k = \overline{0, N-1}, \tag{10}$$

where $x[n], n = \overline{0, N-1}$ - the sequence of points of the signal, N - the length of the signal.

Denote the elements of the DCT matrix by $C[k,n], k,n = \overline{0, N-1}$ whose values are determined by expression (11):

$$C[k,n] = \cos\left(\frac{(2n+1)k\pi}{2N}\right), n,k = \overline{0, N-1} \tag{11}$$

Given (11), we transform expression (10):

$$X[k] = \sum_{n=0}^{N-1} C[k,n]x[k], k = 0, \overline{N-1} \tag{12}$$

Then, having the image I presented in the form of a square matrix, we can obtain its two-dimensional cosine transformation as follows:

$$DCT(I) = C \times I \times C^T , \tag{13}$$

where C is the DCT matrix calculated by formula (11), C^T is the transposed matrix, I is the square size image that is a transformation matrix.

To build a perceptual hash using DCT, follow these steps:

1. Zoom out the image to 32 × 32 pixels to get rid of high frequency.
2. Convert small image to grayscale, so that the hash is reduced three times.
3. Perform a Discrete Cosine Transform for the resulting image. DCT splits a picture into a set of frequencies and vectors.
4. Select upper left block 8x8 of matrix DCT. It contains the lowest frequencies from the picture.
5. Calculate the average for all 64 colors.
6. Construct a hash from the resulting matrix.

After the hash value is calculated for each image in the collection, these values must be compared. For this purpose, a comparison can be made on the basis of Hamming distance, which is a metric of difference of objects of the same dimension [12].

Denote the Hamming distance for the hash values of the images X_i and X_j of length p by $d[X_i, X_j]$ (14):

$$d[X_i, X_j] = \sum_{m=1}^{p} | x_{im} - x_{jm} | \tag{14}$$

The smaller the Hamming distance, the more similar the images.

5.3 Classification of Objects Found on Video

To search for objects that will determine the content of an advertising message relevant to the video stream, the operation of classifying the found common objects in frames is performed. To solve this problem, we will use computer vision methods to identify the objects received namely the classification of objects.

In machine learning, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing instances whose category membership is known. Classification is an example of pattern recognition. A finite set of objects is given for which it is known which classes they belong to. This set is called the training set. The class affiliation of the remaining objects is not known. It is required to construct an algorithm capable of classifying an arbitrary object from the original set.

Each of the test images can be considered as a point in the feature space. Its coordinates are the weight of each of the features of the image. Let the features are specific objects of images. All signs can be seen for the help of detectors, which can be found on the most suitable objects. For the classification of video sources, the method of supporting vectors is considered, which is the basis for the OpenCV library [13]. This method relates to learning methods with a teacher. This means that the system is "trained" in the training set in the form of a multitude of examples and the corresponding answers, after which, based on the established dependencies between the examples and the answers, the system predicts the answers already for the test data. The method belongs to the category of discriminatory linear classifiers, which means that it does not matter how the data was generated, and the classifier makes decisions based on the value of a linear combination of features of objects that are stored in feature vectors

Let us consider the formal statement of the problem of constructing a linear classifier according to the Support Vector Machine (SVM) method [14, 15]. There are training data from n points: $(\overline{x_1}, y_1)....(\overline{x_n}, y_n)$, where y_i can be equal to 1 or −1, pointing to the class that belongs to one of the points $\overline{x_i}$ that make up the real vector with dimension equal to p. It is necessary to find the optimal hyperplane that separating the set of points, for which $y_i = 1$ and the set of points, for which $y_i = -1$. The distance from optimal separating hyperplane to the nearest points from both classes should be maximum.

The hyperplane can be expressed by the following equation (15):

$$\overline{w} \times \overline{x} - b = 0\,, \tag{15}$$

where the vector $\overline{w}$ is the normal vector to the hyperplane, and the value $b/\|\overline{w}\|$ is the distance from the hyperplane to the origin.

In the case of linear separability of training data, two parallel hyperplanes can be constructed. The optimal hyperplane is parallel to them and lies in the middle between them. These two hyperplanes can be described by the following equations (16):

$$\overline{w} \times \overline{x} - b = 1, \ \overline{w} \times \overline{x} - b = -1 \tag{16}$$

In order for each point to be on the right side with respect to the hyperplane, it is necessary to add a restriction:

$$\begin{cases} \overline{w} \times \overline{x_i} - b > 1, \text{if } y_i = 1 \\ \overline{w} \times \overline{x_i} - b \leq 1, \text{if } y_i = -1 \end{cases}. \quad (17)$$

Combining (16) and (17), we obtain the expression $y_i(\overline{w} \times \overline{x_i} - b) \geq 1, \forall i : 1 \leq i \leq n$.

The problem of finding the optimal separating hyperplane can be formulated as follows: to minimize $\| \overline{w} \|$, provided that the classifier is defined $y_i(\overline{w} \times \overline{x_i} - b) \geq 1, \forall i : 1 \leq i \leq n$, where the $\overline{w}$ and b determines the classifier $x \Rightarrow sign(\overline{w} \times \overline{x_i} - b)$, the points $\overline{x_i}$ are called support vectors.

The support vector method is able to effectively solve the nonlinear classification problem using the kernel trick method. In this case, the data is implicitly mapped into the space of features of a larger dimension, under the assumption that there the data will be linearly separable. The idea of the method is that instead of the scalar product of points, a nonlinear kernel function is used. The function corresponding to the optimal decision function in the original space will be nonlinear due to the nonlinearity of its mapping into a space of higher dimension.

5.4 Video Audio Track Analysis Algorithm

The analysis and recognition of the language from the audio track of the video and the search for keywords in the resulting text are necessary to search or create an advertising message that matches the context of the video. The first step of the algorithm is to recognize the audio language of the video using the Mel-Frequency Cepstral Coefficients (MFCC) vocabulary. The second step is to search for keywords in the resulting text based on the TF-IDF algorithm.

Speech Recognition Algorithm from Audio Track Video. When recognizing speech, first of all necessary to select the words in the text. Let the speech contain pauses that will separate the words. [16].

An audio signal is input to the speech recognition system. Sound is divided into frames that represent sections of 25 ms with overlapping frames of 10 ms. To process an audio signal, it should be converted either in the form of a signal spectrum or in the form of a logarithmic spectrum, followed by scaling using the Mel-scale. Then the signal is presented as MFCC (Mel Cepstral Coefficients) by applying a Discrete Cosine Transform. MFCC is usually a vector of thirteen real numbers; it represents the energy of the signal spectrum. This method takes into account the wave nature of the signal, the Mel-scale identifies the most significant frequencies perceived by humans, and any number, which allows you to compress the frame and reduce the amount of processed information, can set the number of MFCC coefficients. The calculation of MFCC occurs according to such an algorithm [17, 18].

The initial speech signal can be written in a discrete form as $x[n], 0 \leq n < N$, where N is the frame size or window length, $x_j[n]$ is the j - th frame. Apply the Fourier transform to it:

$$X_{\alpha}[k] = \sum_{n=0}^{N-1} x_j[n] e^{-j2\pi nk/N}, 0 \le k < N \ . \tag{18}$$

We define a filter bank with $M = \{m \mid m = \overline{1, M}\}$ filters, where filter m is triangular filter by

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])}, & f[m] \le k \le f[m+1] \\ 0, & k > f[m+1] \end{cases} \tag{19}$$

where $f[m]$ is determined according to [17]. We then compute the log-energy at the output of each filter as (20):

$$S[m] = \ln\left[\sum_{k=0}^{N-1} | X_{\alpha}[k]^2 H_m[k] |\right], \ 0 \le m < M, \tag{20}$$

The Mel Frequency Cepstrum is then the Discrete Cosine Transform of the M filter outputs:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos(\pi n (m + 1/2)/M), \ 0 \le n < M \ , \tag{21}$$

We get a set of MFCC.

Word recognition is reduced to comparing the set of numerical values of a digital signal and words from a dictionary. We have L words, to each of them we associate a set of MFCC coefficients. This correspondence is called a model. For each model, we find the average (Euclidean) distance between the identified MFCC-vector and the model vectors. We choose as the correct model, the average distance to which will be the smallest.

Keyword Searching Algorithm in Soundtrack Text Based on TF-IDF. The keywords that will be used to search for advertising that correspond to the video content should be identified in the recognized text from the audio track. To find keywords, we use the algorithm based on TF-IDF [19]. The TF-IDF statistic is used to estimate the meaning of terms in a body of text. The meaning of the term is directly proportional to the number of uses of the term in the text, and inversely proportional to the number of uses of the term in other body texts (documents). So we have (22):

$$TF(d, t) = \frac{n_{dt}}{n_d}, \ IDF(t) = \frac{N}{| d_j |} \tag{22}$$

where n_{dt} is the number of occurrences of the term t in the document d ; n_d is the number of terms in the document d ; $| d_j |, d_j \supset t$ is he number of documents in which the term t occurs.

In modern search engines, the calculation of document relevance is based on the Okapi BM25 function based on a probabilistic model [20].

$$W(TF) = IDF(t) \frac{TF(d,t)(k_1+1)}{TF(d,t) + k_1(1-b+b\frac{|d|}{avgl})} \quad (23)$$

where $avgl$ is the average length of documents in a collection. The constant k_1 determines how much the weight reacts to increasing TF . If $k_1 = 0$, the weight reduces to the term-presence weight only; if k_1 is large, the weight is almost linear in TF . b is the tuning constant, the simple normalization factor. According to [20] a value of around b = 0.75 is good.

Before applying the TF-IDF algorithm, the text is pre-processed: "stop words" are deleted, as well as the remaining words are brought to a common basis by dropping the suffix and prefix using stemming. The number of keywords that are the result is 1/10 of the total number of words because the density of the words in the resulting text from the video audio track is very high.

6 Contextual Advertising Design Algorithm

The general scheme of the algorithm for solving the problem of finding relevant contextual advertising for an arbitrary interval of video duration.

Step 1: Select a set of objects from the video stream frames.

Step 1.1. Get frames from video stream.

Step 1.2. Find outlines of objects in an image using Prewitt.

Step 1.3. Calculate the areas of objects in the image.

Step 1.4. Calculate the perceptual hashes of image objects using the pHash method.

Step 1.4.1. Resize image to a size of 32 × 32.

Step 1.4.2. Perform an image discoloration operation.

Step 1.4.3. Determine the average color value of the matrix.

Step 1.4.4. Perform a discrete cosine transform for the resulting matrix.

Step 1.4.5. Construct a hash for the resulting matrix.

Step 1.5. Find shared objects using the Hamming feature.

Step 1.6. Play pictures of shared objects.

Step 1.7. Classify many common objects by computer vision.

Step 2: Find a set of audio content objects for time you want your ads to show.

Step 2.1. Retrieve text from audio stream using MFCC coefficients.

Step 2.2. Find a set of terms referred to in the audio stream of text using TF-IDF.

Step 2.2.1. Break the text into words, remove the stop words, and bring the words to the common ground with stinging.

Step 2.2.2. Add text to a collection of texts, and list for each word the number of texts in which the term occurs.

Step 2.2.3. For each term in the text, calculate the frequency of its occurrence (formula 2.2).

Step 2.2.4. For each term, determine its significance (estimate) based on Okapi BM25 function (formula 2.16).

Step 2.2.5. Arrange words in descending order of importance.

Step 2.2.6. Return 1/10 of a third of the phrases in an ordered list.

Step 3: Find a set of keywords from the video stream metadata.

Step 4: Find the ads that are as close as possible to the content of the video stream, considering the many possible advertisements, subjects from video frames, audio stream, and video metadata.

The end of algorithm.

7 Analysis of Results

Consider video as input (figure 1):

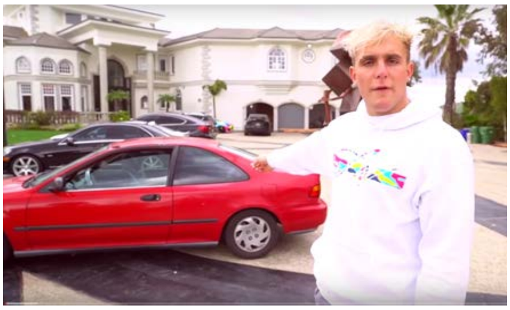

Fig. 1. Video frame being analyzed.

The algorithm developed consists of analyzing the meta data of the video, its frames and the audio track and finding the appropriate advertisement.

This video has the following meta data: JakePaul; CarInPool; TakeFunWithFriends; MyFriendsPool. After processing, they received the following ordered set of tags with the frequency of their appearance: pool – 2; friends – 2; jake – 1; paul – 1; car – 1; take – 1; fun – 1; my – 1.

To analyze the video image, the video stream fragment is split into frames at one-second intervals (figure 2). The frames are analyzed using the developed algorithm and methods of computer vision.

Fig. 2. Frame from the video stream.

The result of the frame analysis is an ordered set of objects with their coefficient of significance in the video stream (table 1):

Table 1. Frame objects and Significance factors

Frame objects	Significance factors	Frame objects	Significance factors
vehicle	22.74494457244873	windshield	5.518191874027252
car	18.89864546060562	tree	5.253252983093262
vehicle door	12.314380288124084	glass	3.742722451686859
window part	6.391701281070709	swimming pool	1.5801533460617065
grass	5.551468133926392	road	1.05413019657135

The analysis of an audio track consists of two stages. The first step is speech recognition. The result of this operation for the video stream snippet is the following text:

"Scarlet Starlet recipe Sky broke yo we almost died less than you need a better car by Scarlett so wait any car okay not like well there's limitations like a better car than this one but not like a Lamborghini. it's sorry lol Pro subscribe seriously I still got subscribe button for this after have you done this before now why do you guys keep saying that huh".

The second step is the operation of finding keywords for a given text and determining their weights (table 2):

Table 2. Keywords and weights

Keyword	Weight	Keyword	Weight	Keyword	Weight
car	6	wait	2	starlet	2
subscribe	6	limitations	2	scarlett	2
scarlet	2	lamborghini	2	lol	2
recipe	2	died	2	broke	2
sky	2	beauty	2	pie	2

The result of selecting contextual advertising from the set of all advertisements according to the criterion of maximizing the percentage of contextual objects to a plurality of objects from video frames, sound stream and video data is to advertise a Lincoln machine (figure 3).

Fig. 3. Ad Search Result

8 Conclusion

The theoretical aspects of the problem of streaming video analysis are considered in the article. Algorithm for video stream analysis that takes into account the features and limitations of the task of analyzing video stream in real time and finding advertisements in accordance with video content is presented.

Real-time video content analysis takes into account the following factors: metadata, video frames, and video audio. The algorithm developed breaks down the video stream into snippets, after which the advertisement is displayed, and the content for each of them is analyzed in parallel.

In the first stage, the video is storyboarded and the frames obtained are analyzed by searching for the main objects using image and computer vision algorithms. The second stage consists of analyzing and recognizing the audio language of the video track and searching for keywords in the resulting text. In the third step, the algorithm analyzes the video metadata and finds the keywords in them. In the fourth stage, we search for ads that are as close as possible to the content of the video stream, based on the data received about the objects.

A reference example is given, the results of which testify to the operability of the algorithm. Instead, most of the considered approaches are slow, non-scalable, and unable to work steadily in real time.

References

1. YouTube for press. Landline 10.11.2019. http://www.youtube.com/yt/press/uk/statistics.html/
2. Chong-Wah Ngo, Ting-Chuen Pong, and HongJiang Zhang. Recent advances in content-based video analysis. In: International Journal of Image and Graphics 1(3), 445-468. (2011).

3. Ballan L., Bertini, M., Del Bimbo A., Seidenari L., Serra G. Event Detection and Recognition for Semantic Annotation of Video. In: Multimedia Tools and Applications 51, 279-302. (2011).
4. Zhao L., He Z., Cao W., Zhao D. Real-Time Moving Object Segmentation and Classification From HEVC Compressed Surveillance Video. In: IEEE Transactions on Circuits and Systems for Video Technology, Vol. 28(6), 1346 –1357. (2018).
5. Lee Y. J., Kim J., Grauman K. Key-segments for video object segmentation. In: 13th International Conference on Computer Vision, 1995-2002. IEEE Computer Society, USA. (2011).
6. Jackson S., Miranda-Moreno L., St-Aubin P., Saunier N. A Flexible, Mobile Video Camera System and Open Source Video Analysis Software for Road Safety and Behavioural Analysis. In: Transportation Research Record: Journal of the Transportation Research Board. 2365, 90-98. (2013).
7. Jiang W., Cotton C., Loui A. Automatic consumer video summarization by audio and visual analysis. In: 2011 IEEE International Conference on Multimedia and Expo (ICME 2011), 1-6. (2011).
8. Snoek C., Worring M., Smeulders A. Early versus late fusion in semantic video analysis. In: 13th ACM International Conference on Multimedia (MM05), 399-402. (2005).
9. Vasconcelos N., Lippman A. Statistical models of video structure for content analysis and characterization. In: IEEE Transactions on Image Processing, Vol. 9(1), 3-19. (2000).
10. Martyanova A.V., Mukhamatnurov V.E. Analysis of Edge Detection Algorithms Based on Gradient and Aggregation Operators. Part 1. The algorithm of differentiation based on aggregation operators. In: 1th International Conference on Computer Image Analysis and Intelligent Solutions in Industrial Networks (CAI-2016), 74-77 (2016).
11. Rudakov I.V., Vasiutovich I.M. Analysis of Perceptual Image Hash Functions. In: Science and Education of the Bauman MSTU. 08, 269–280. (2015).
12. Mohammad Norouzi, David J. Fleet, Russ R. Salakhutdinov. Hamming Distance Metric Learning. In: 25 Annual Conference on Neural Information Processing Systems (NIPS-2012), 1071-1080. (2012)
13. OpenCV. Landline - 05/11/2019. https://docs.opencv.org/master/
14. Cristianini N., Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press. Cambridge. (2000).
15. Srivastava D., Bhambhu L. Data Classification using Support Vectors Machine. In: Journal of Theoretical and Applied Information Technology. 12(1), 1-7. (2010).
16. Alyunov D.Yu., Sergeev E.S., Pigachev P.V., Mytnikov A. N. Implementation of the algorithm processing and speech recognition. In: Modern high technology, 3, 225-230. (2016).
17. Huang X., Acero A., Hon Hsiao-Wuen. Spoken Language Processing. A Guide to Theory, Algorithm and System Development. 1st edn. Prentice-Hall, Inc. New Jersey. (2001).
18. Mikhaylov V.G., Zlatoustov L.V. Speech Measurement. Radio and Communication. Moscow. (1987).
19. TF–IDF. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. (2011)
20. Jones K. S., Walker S., Robertson S. E. A probabilistic model of information retrieval: development and comparative experiments. Part 2. In: Information Processing and Management, Vol 36 (6), 809-840. (2000)