Date of publication JUN-30, 2022, date of current version JUN-11, 2022.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.21.2.2590

Development of an Effective Bootleg Videos Retrieval System as a Part of Content-Based Video Search Engine

AHMAD SEDKY ADLY[1], ISLAM HEGAZY[2], TAHA ELARIF[2], M. S. ABDELWAHAB[1]

[1]Computer Science Dept., Faculty of Info. Technology, Misr University for Science & Technology, sedky@must.edu, mswahab@must.edu.eg
[2]Computer Science Dept., Faculty of Computer & Information Sciences, Ain Shams University, islheg@cis.asu.edu.eg, taha_elarif@cis.asu.edu.eg

Corresponding author: Ahmad Sedky Adly (e-mail: sedky@must.edu).

ABSTRACT Many research studies in content-based video search engines are concerned with content-based video queries retrieval where a query by example is sent to retrieve a list of visually similar videos. However, minor research is concerned with indexing and searching public video streaming services such as YouTube, where there is a dilemma for misusing copyrighted video materials and detecting bootleg manipulated videos before being uploaded. In this paper, a novel and effective technique for a content-based video search engine with effective detection of bootleg videos is evaluated on a large-scale video index dataset of 1088 video records. A novel feature vector is introduced using video shots temporal and key-object/concept features applying combinational-based matching algorithms, using various similarity metrics for evaluation. The retrieval system was evaluated using more than 200 non-semantic-based video queries evaluating both normal and bootleg videos, with retrieval precision for normal videos of 97.9% and retrieval recall of 100% combined by the F1 measure to be 98.3%. Bootleg videos retrieval precision scored 99.2% and retrieval recall was of 96.7% combined by the F1 measure to be 97.9%. This allows making a conclusion that this technique can help in enhancing both traditional text-based search engines and commonly used bootleg detection techniques.

KEYWORDS Content-based video search engine; bootleg video detection; content-based video indexing and retrieval; bootleg videos; CBVSE; CBVIR; feature extraction; search engine; video search; video retrieval.

I. INTRODUCTION

CONTENT-based video indexing and retrieval in the application of content-based video search engines CBVSE is a topic of a great need to resolve main problems in CBVSE as searching online video databases using video queries. Another important issue is providing a searching method that is both transparent and more accurate than traditional text-based search engines used. Moreover, text-based search engines mostly rely on English as their primary search language with a non-transparent searching mechanism instead of actual visual contents of the video. Additionally, indices for such search engines are built on text-based classification of video records such as file names, video descriptions, tags, captions, narration text, etc. This compels video search to be less accurate, unsatisfactory and more manipulated [1].

Another drawback of text-based video indexing is the problem of video content-stealing and copyrighted video material misusing, as almost all video material contents are copyrighted and available for streaming and downloading on online video streaming services and databases such as YouTube, giving a chance for pirates to copy such contents with false information and a new record without the YouTube even noticing the violation, until found by the video copyright holder, who compels a complaint to get his copyrighted material removed from publishing. One of the most difficult procedures in the online video streaming services is keeping track of the content of video files being uploaded daily to web services by numerous users as in YouTube which receives an average of 300 hours of video per minute with more than 2 billion users according to YouTube Press [2], at the time of writing this paper. They distribute video files amongst multiple servers in many countries, which makes it impossible to verify the copyright misuse until a complaint is filed. In [3], Balouch et al. introduced a detection algorithm for copyrighted videos on YouTube using high-level objects or concepts extracted from video shots. A video clip of the original copyrighted material is used as a query to be uploaded and compared to each video in the video database. This approach has a drawback of complex computation and a vast number of video records changing by the minute if applied on the YouTube service.

This paper is organized into five sections, the second section discusses the related work on content-based video

indexing and retrieval search engines. The third section shows our method and techniques used in this research. The fourth section is all about experiments and results, showing our retrieval system, matching algorithm, and its performance compared to state-of-the-art work in content-based images and videos retrieval and showing the process of acquiring query videos, extracting features, and retrieving relevant videos including bootleg videos. Finally, the fifth section presents conclusions and our future experiments and research work.

II. RELATED WORK

There has been a major leap in the production of digital video and video streaming services in the past decade or so. Video databases and repositories are one of the main issues with great need to comprehend as they lack organization and searching accuracy, which leads to time consumption and high computation cost as well as to the need for human laboring assistance. The text-based video search engines failed due to inaccuracy and non-transparency in solving video searching problems using content queries rather than text annotations described by the user. As a result, the urge to find a better video retrieving methodology drove the research towards content-based video search engines. Thus, increased video content analysis demands pushed content-based video indexing and retrieving in the active researching areas worldwide. In [1], a survey was introduced reviewing issues and challenges facing content-based video search engines.

In [4], Mazaheri, et al., introduced principles of a video ranking model based on scoring ranks computed for each video file. These ranks are created out of basic concepts provided to the system in two different ways, a direct way from the user by selecting one or multiple concepts from a list, or an indirect method of analyzing the user text queries for concept detection. The concepts are averagely weighted according to the scores of the concept's detectors. The method applies a latent ranking SVM algorithm and latent variables to tag and labels mutual complementary shots. As for their feature vector, they used object-based multi-concepts in form of 50 pairs of concepts used for training, and 50 triplets of concepts. As for indexing a dataset of 230 videos from YouTube was gathered and clipped to pairs of shots in the database, ranked by pairs of concepts, and retrieved using queries that include a checklist of 9 of the concept pairs used. The retrieval algorithm was based on solving the ranking problem of the provided multi-concepts extracted from the text query and/or concepts checklist. In [5], Garcia, et al., implemented a visual feature temporal aggregation technique to retrieve videos based on image queries. They used two different models of temporal redundant aggregation, local binary temporal tracking (LBTT) and deep feature temporal aggregation (DFTA), performing large scale retrieval by reducing the amount of processed data that exploits highly correlated image redundancy based on local and standard features like pixel intensity and other deep learning representation features such as temporal encoding tracking binary vectors through time. LBTT uses standard local features for image representation as values of pixel intensity regions of 256-dimensional binary vectors. As for temporal encoding, binary features are traced along with time by matching sequential frames descriptors using Hamming distance, filtering out frames far apart in pixel space. The shot boundary detection then takes place between sequential frames sharing visual similarities and packed into shots, and the boundary between two shots is detected if nonsimilar two sequential frames are traced, with each shot being represented by a set of key features. Indexing for key features is carried out using a kd-tree. Retrieving based on image query is carried out by extracting pixel intensity and other standard local features performing a nearest neighbor (NN) for each feature against the indexed key features using brute force. The second model DFTA is a method based on deep learning of visual features on temporal aggregation using an image representation vector called RMAC to encode visual contents of each of the frames. Obtained from the last layer of the CNN, the RMAC extracts local features using a max polling process which activates the feature map's various regions. The temporal encoding is performed by representing each video segment into one feature vector to reduce data redundancy, using two different approaches to aggregate RMAC global vectors into video shots, DLTA-Max and DLTA-Mean, which encodes shots by computing the max value of RMAC and average of RMAC feature in the frame respectively. Indexing was performed using MoviesDB dataset [6]. The experiment showed that LBTT using frames of 720-pixel wide lead to 76% to 97% accuracy, DLTA using frames of 1024 pixels wide lead to 12% to 22%, and DLTA-Max along with DLTA-Mean registered 47% to 69% accuracy. In [7], Mühling, et al., introduced a novel algorithm for a media and television production video retrieval using deep learning for concept detection combined with face detection, recognition, and clustering, as an efficient retrieval and inspection for video records approach. They used a combination of concept detection and similarity searching as a multi-tasking learning analysis algorithm with half the time of calculations and video retrieval by applying weights sharing of the network, a lexicon of visual concepts, and a new visualization of components. Results introduced showed a mean average precision for concepts detection of 90% on the first 100 videos of the Movie Trailers Face dataset [8]. Gabriel de Oliveira Barra, et al. [9] introduced the large-scale content-based video retrieval system or LIvRE, which is usable by interfaces with lightweight, portable specifications and built to assist users in searching for video content. Features vector includes multiple global and local features that allow users to get less complicated comparisons between features that are new and the existing features. Joint Color Descriptor (JCD), Pyramid Histogram of Oriented Gradients (PHOG), MPEG-7 descriptors Edge Histogram, and Color Layout and Scalable Color are global features used, along with a bag of visual words and VLAD aggregation as local features used for image retrieval. Local features are mainly based on the OpenCV employments of the SIFT and SURF techniques. Moreover, the Lucene Image Retrieval Engine (LIRE) fully employs the original SIMPLE descriptor in applying the global features on local image patches with configurable key point detectors. The system has three main components: video parsing, indexing, and retrieval. The video parsing component accomplishes indexing by first acquiring an input containing a dataset of video sequences and performing keyframes extraction along with obtaining features form the keyframes' images in containing documents. Subsequently, these resulting documents from the parsing process are then uploaded and indexed by the indexing component. This component ensures the search engine's organization of video segments and the keyframes to be listed in the returned results. As for the retrieval component it is combined with a responsive web app interface feature that allows the user to send a query to the search engine and then returns a list of results with the ability

to work on mobile and desktop devices browsers. As for the experiment, the implementation process was evaluated utilizing a huge video dataset with over 1000 hours of video recordings. With a performing result on the runtime using a Solr core, with a single mid-range server, running for an uncashed image requests of 136, with a 3,808,760 population of video keyframes indexed, this resulted in 19.5 seconds average request time with 16.3 seconds of median request time. Luca Rossetto, et al. [10] proposed the IMOTION content-based video search engine, which is built upon Cineast system, with an original design for a video search engine using sketch-based queries along with other query types. The system utilizes a variety of low-level features of both images and videos, with high-level features consisting of temporal and spatial types which can be combined and used. This means that the system supports query types of sketching, motion, by image example, and any mixture of all types, also providing a relevance feedback from the users. Low-level feature vector includes global features like (average/median color, dominant shot colors, chroma/saturation, color histogram, shot position), regional color features (color moments, registered color grid, color layout descriptor, color element grids, subdivided color histogram), regional edge features (partitioned edge image, edge histogram descriptor, dominant edge grid), and motion features (directional motion histograms, regional motion sums). High-level features vector includes relevant descriptors extraction including techniques using machine learning utilizing deep neural networks as spatial keyframe appearance (neural network architecture, and temporal (using video shot motion) data. The dataset used was ImageNet which contains 1000 categories and about 1.2 million images. Retrieval modes support the search of known items with three various types such as: query-by-sketching (prompting the user to enter a drawn sketch with either line or colored sketch drawing), query-by-example (prompting the user to drag and drop a query object of a previously retrieved sequence to find similar video sequences, and motion queries (allowing the user to signify the objects in motion crosswise frames in consequent order using a partial flow field). All of which are accompanied by relevance feedbacks to refine the query results. This resulted in producing a similar relevant set and far from non-relevant ones. Lastly, there were no performance results published.

There is a lot of research considering content-based indexing and retrieval where precision enhancement is the major concern as in [11-14], some of them considered enhancing speed and accuracy [15, 16]. However, a very large number of state-of-the-art researches on content-based indexing systems in the applications of content-based video search engines, are still far from acquiring fast retrieval for video records from index database due to matching and classification procedures, which affected the interactive use of such systems [17].

III. METHODOLOGIES & TECHNIQUES

Video sequences are constructed of subsets of scenes, furtherly dissected into shots, and lastly into fixed images called frames. A video shot is a set of frames captured without interruption for a continuous action using a single camera operation.

Let us consider E as a set of video files on a public web video streaming service, where $E = \{v_i, \ldots, v_n \mid n$ is an integer, $0 \le n < \infty\}$, where n is the number of video files and vi represents each video file and $i \in n$. At the same time, each video vi has a set of shots s_{ij}, where each video $v_i = \{s_{ij}, \ldots, s_{im} \mid m$ is an integer, $0 \le m < \infty\}$, and $j \in m$. Shots are furtherly segmented to a set of frames f_{ijk}, where each video shot $s_{ij} = \{f_{ijk}, \ldots, f_{ijq} \mid q$ is an integer, $0 \le q < \infty\}$, and $k \in q$. Our goal is to build a content-based video search engine with a video index set I crawled from video set E to extract features, classify, and store each video's URL, keyframes and feature vectors in a record r_i for each member in E, where $I = \{r_i, \ldots, r_n\}$, and $i \in n$, the same size of E. This is done to benefit the search for a given video file query Q_v to find the visually similar video(s) records set $R_v \subset I$, with similar video shots temporal combination similarity vector (T_v), video shots combination concepts similarity vector (C_v), and video classification, which are all extracted from Q_v. This is implemented using two-phased search criteria by the means of following:

The first phase involves applying a cosine similarity measure SC for the temporal feature vectors T_v extracted from Q_v, against vectors Ti in all records or a certain video class, where $r_i \in I$, and $i \in n$, such as:

$$\mathrm{SC}(T) = \frac{\sum_{k=0}^{K}(Tv)_k \cdot \sum_{k=0}^{K}(Ti)_k}{\sqrt{\sum_{k=0}^{K}(Tv)_k^2} \cdot \sqrt{\sum_{k=0}^{K}(Ti)_k^2}}, \quad (1)$$

where Tv is the temporal and concepts visual representation respectively of the given query Q_v and $k = \{0, \ldots, K\}$, where K is the elements count for each vector. Each record's comparison results are then equated against a temporal similarity threshold th_t to produce a temporal similar set TS that contains only video records that are temporal combinational related to the given query Q_v.

Secondly, we applied and compared three different similarity metrics on concepts features vector C_v extracted from Q_v, against all concept's vectors tied to TS temporal similar set called CS concept similar list from all records $r_i \in CS$, and $i \in m$, where m is the number of concepts similar records found by first temporal search phase inside index I. In the following a brief demonstration of three different similarity metrics used in the second phase of the concept combinational search is shown:

1) **Cosine similarity algorithm:** applying the same SC similarity measure algorithm on the concepts feature vector C_v extracted from Q_v, against vectors C_i in all records in CS, where $r_i \in CS$, and $i \in m$ is as follows:

$$SC(C_v, C_i) = \frac{\sum_{k=0}^{K}(Cv)_k \cdot \sum_{k=0}^{K}(Ci)_k}{\sqrt{\sum_{k=0}^{K}(Cv)_k^2} \cdot \sqrt{\sum_{k=0}^{K}(Ci)_k^2}}, \quad (2)$$

where C_v is the representation of the concepts of the given query Q_v, and $k = \{0, \ldots, K\}$, and K is the elements count for each vector.

2) **Minkowski distance similarity algorithm** is a generalization algorithm for both Euclidean distancing and Manhattan distancing algorithms using a metric of a normed space vector. The Minkowski distance is also inversely proportional to the distance calculated, the same as the Euclidean distance. Therefore, applying Minkowski distance similarity MD on the concept feature C_v extracted from Q_v, against vectors C_i in all records in CS, where $r_i \in CS$ and $i \in m$, looks as follows:

$$MD(C_v, C_i) = \sqrt[p]{\sum_{k=0}^{K}|(C_v)_i - (C_i)_i|^p}, \; p > 0, \quad (3)$$

where C_v is the representation of the concepts of the given query Q_v, $k = \{0, \ldots, K\}$, K is the elements count for each vector, and p is the order of the Minkowski distance, which might be altered with different values other than 1, 2, or ∞ that represent respectively: Manhattan, Euclidean, and Chebyshev distance measures.

3) **Jaccard similarity/coefficient algorithm** is a vector similarity metric that measures similarities of two vectors by computing the number of the commonly shared elements and dividing it by the total number of the elements in the two vectors [18]. Therefore, applying Jaccard similarity JS on the concept feature C_v extracted from Q_v, against vectors C_i in all records in CS, where $r_i \in CS$ and $i \in m$, is as follows:

$$JS(C_v, C_i) = \frac{|C_v \cap C_i|}{|C_v \cup C_i|} = \frac{|C_v \cap C_i|}{|C_v| + |C_i| - |C_v \cap C_i|}, 0 \leq JS(C_v, C_i) \leq 1 \quad (4)$$

where C_v is the concepts' representation of the given query Q_v, $k = \{0, \ldots, K\}$, and K is the elements count for each vector.

In the next sections, we will introduce the methods of video classification for our content-based video retrieving system and dataset.

A. CONTENT-BASED VIDEO RETRIEVAL SYSTEM

The main target of building a content-based video search engine is to have a video retrieval system with the objective of retrieving users' queries with the most relevant and similar videos from the video index. Moreover, a retrieval system is only functional and can be used if its main video index is assembled and ready for use. This is accomplished by searching the video index dataset for users' queries using a certain search criterion including similarity metrics and video classification techniques to reduce computational cost and increase the accuracy of retrieved similarity lists.

Furthermore, YouTube as a public video streaming web service is the biggest resource of videos recognized worldwide and over the Internet. The covered topics on YouTube have an unknown profundity such as multiple languages and inadequate information describing videos. Most of the videos have neither classification nor categorization and the biggest challenge is the copyrights dilemma and the misuse of copyrighted materials [1].

Fig. 1 shows the steps of how the user submits a query by example video to the search engine's retrieving system and the required two phases and five steps needed to construct a similarity list of URLs pointing to videos located on YouTube's public streaming web service visually similar to the submitted video file query. The first phase contains three steps and all of them are performed and computed on the client-side device to reduce both bandwidth and computational consumption and only send a features vector to the server-side to process and search for similarities inside the video index dataset.

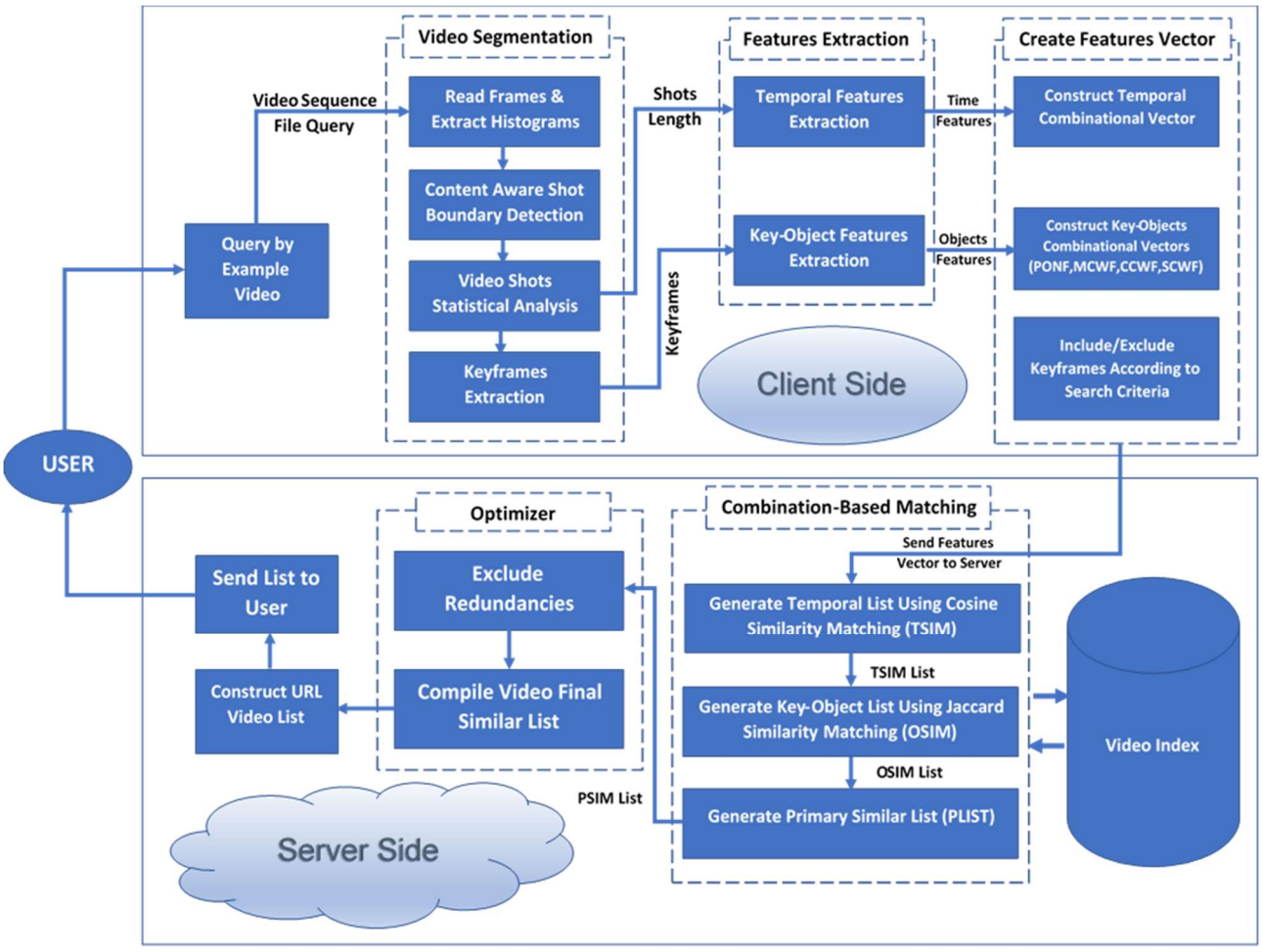

Figure 1. Content-Based Video Retrieval System.

The first step is video segmentation and it has four procedures that ensure preparing and divide the video into keyframes and statistical information from which the two main features are extracted, including reading all video frames to extract histograms to divide the video into shots and measuring shots count, each shot time length, and extracting from each shot a set of three keyframes to be sent to the next step. The second step is using both statistical information and keyframes extracted from the video file to first calculate the temporal vector for all shots in the video sequence, and all keyframes extracted from all shots are then submitted into an object detector to extract object annotations from each keyframe image. After temporal and object features are extracted, the third step is to assemble the main features vector to be sent to the server-side via the network for combinational-based matching, retrieval, and optimization. As for the second phase, the features vector are then acquired by the matching procedure which uses two-phased combinational-based similarity metrics filtering technique that filters first the temporal similarity list (TSIM) out of the video group based on video classification, and then using this list in the second filtering phase matching the object feature vector of the query against TSIM list members only to construct the object similarity list (OSIM) list which will be sent to the optimizer to refine and exclude redundancies if found and construct the final visually similar list of YouTube URLs to be sent back to the user. In the next sections of this paper, further details will be addressed concerning each phase, step, and procedure.

B. NON-SEMANTIC-BASED VIDEO QUERIES

Video queries are categorized into two main types; the first type is the semantic-based video queries which include simple queries with no complex processing to extract features such as keywords queries and natural language queries, all of which are implemented in most commercial applications of video search engines. The second type is the non-semantic video queries which involve much more sophisticated queries such as query by example video, sketch and object, all of which require more processing cost to extract and match features. However, this research considers only the non-semantic-based queries and precise queries by example video to perform the retrieval process to search for similar video files in the video index dataset.

Furthermore, query by example video is given by the user selecting a local video file or URL link for a public video stream, after the query is acquired features vector extraction begins. The features vector is then sent to the server to estimate features' comparability and matching against video records in the video index dataset. Video query processing cost is very high on both bandwidth and server's computational power; therefore, users query video files will be processed on the client-side to extract features vector to prepare and send to the server for matching, which results in optimizing both bandwidths and computation costs. However, the larger the video query is the longer the time necessary to segment, extract features, and construct features vector to be sent to the server for matching and retrieval [1].

C. VIDEO QUERIES SEGMENTATION

Extracting the contents and features automatically from video query files requires numerous technologies, due to the complex and rich contents encoded within these videos. In addition, video contents have a substantial structure of visual, audial, and lingual content like images, tunes/music, narration and more. The following text discusses these techniques.

C.1. VIDEO SHOT BOUNDARY DETECTION

The shot boundary is the edge found between two consecutive video shots called transition and it may be divided into four transition categories cutting, fading, wiping, and dissolving. Cut transitions are the shot's simplest most strict and complete transformation to the next shot, the other three categories are more sophisticated and represent changing gradually involving more than one frame in the transition area between the two consecutive shots [19].

After the user submits the video file query, the process of segmenting the video sequence into shots begins. This study used a histogram-based shot boundary detection recognizing change between two consecutive frames via means of intensity difference between pixels. However, a study by Almousa, et al. [19] showed evaluation of many shot boundary detection tools that uses histogram-based including PySceneDetect [20], FFprobe, and FFmpeg [21], and an experiment was conducted based on performance and speed using diversity values of thresholds. Furthermore, it has been found that based on execution speed, FFmpeg, PySceneDetect, and FFprobe are ordered respectively. As for accuracy, FFmpeg and FFprobe made a better performance than PySceneDetect using a gold standard.

Moreover, in an early experiment conducted in this study to find a threshold value that is optimal to our video index dataset [22], the tools involved in this experiment were PySceneDetect and FFmpeg. A deferential number of video files from the dataset were used and evaluated separately. A set of 6 groups were used, each had 15 randomly chosen videos from the dataset. Increasing the threshold for both tools from 1% to 30% for all 6 groups, each group had a five percent closed intervals threshold values tested on 15 random files, with a total of 90 random videos, with an overall number of frames 6,829,975, and a sum of 4,086 video shots. Henceforth, the experiment was conducted concerning performance and by applying precision and recall as performance parameters. It has been found that the FFmpeg had the best performance in terms of speed and accuracy, which concluded that the optimal threshold ranged between 3.6 and 5.7% resulting in using a fixed threshold for both indexing and retrieval systems of 5.7% with the least computational cost.

C.2. KEYFRAMES EXTRACTION

Extracting feature vectors either visual or semantic from a query video file requires extracting keyframes from every shot in the video sequence. Keyframes are defined as an abstracted signification for video shots as a group of still images that represent shots in one or many numbers. Keyframes are a reputable reduction for the redundant multiple frames that any video sequence or video shot contains to one, two, or three keyframes briefly describing the video stream or shot. It is also crucial for video search with faster processing and less computational cost when it comes to commonly used frame-by-frame matching. However, choosing keyframes is crucial with a maximal representation for every video shot and with no redundancy [22].

Furthermore, a histogram-based approach was used as mentioned earlier for shot boundary detection and keyframes

extraction using tools such PySceneDetect and FFmpeg. By applying FFmpeg, a set of 3 keyframes are extracted from each video shot one from each third of the video shot's length, one representing the first third, another for the middle, and a keyframe for the last third. Which produces 3Ns keyframe images for each video query, where Ns is the number of shots in the video query's sequence. Fig. 2 shows the process of keyframes extraction from a query video.

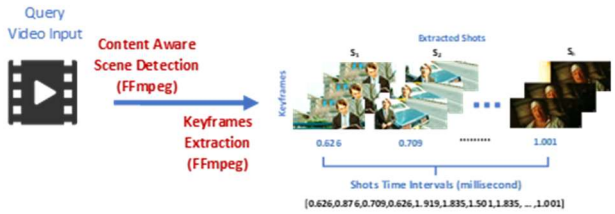

Figure 2. Shot Detection and Keyframes Extraction Using FFmpeg Tool.

D. Features Extraction

D.1. VIDEO SHOTS TEMPORAL ANALYSIS AND EXTRACTION

Temporal relations are one of the ontology techniques used to solve content-based video indexing and retrieval problems, by detecting temporal relations between shots extracted from a video sequence, depending on the fact that shots individually have a temporal relation linking them with other adjacent shots in the video sequence.

Sequence alignment is known as one of the greatest bioinformatics fields that properly exemplify the relations between DNA or RNA proteins and their sequence arrangements to recognize similarities between regions of the protein chains [23]. However, some computer vision researches adopted sequence alignment problems using multimode data with high dimensions applying alignment between videos with untrimmed sequences and video sequences text abiding actions [24].

Another approach used for temporal relations analysis is term-based similarity measurement/function which is a quantification function for measuring similarity between two objects/vectors and reflects the intimacy degree of the targeted object/vector corresponding to features distinguishing clusters indexed in the dataset. Before the clustering process is exploited, similarity measurement/function must be chosen and applied [25]. Selecting a proper similarity measurement/function is vital for relational analysis, which in this case is a temporal relational analysis and matching.

D.2. KEY-OBJECTS EXTRACTION (CONCEPTS)

Convolutional Neural Network (CNN) is one of the fields of machine learning which is a type of artificial intelligence that has many applications in visual recognition, image classification, and objects/concepts recognition. CNNs are built on weighted neurons with learning abilities that enable each neuron to accept an input value to process using a dot product function that forms one single differentiation scoring function for the whole network taking image pixels for classification according to score [26]. The result of applying CNNs on the image as initial inputs has proven to be accurate and provoked more development and implementation for the forward function for efficiency and reducing parameters numbers in the network [27].

Most of the earlier researches in object detection using visual recognition focused on classifying images applying fixed sets of visual segments. Approaches vary depending on different parts of the images or video frames, which makes them suitable for deep neural networks to be used. This was reflected in numerous works, for example, applying deep neural networks using the attention-based recurrent models, machine translation, games and motion tracking, text in images recognition, and caption generation that showed promising results in the past few years [28-31].

Consequently, object detection and extraction from an image or a video frame is the process of bounding all significant areas containing all the relevant objects while ignoring background and irrelevant parts of the image. The bounding process is generally expressed by a surrounding rectangle with coordinates of top-left origin point, width, and height. However, complex shapes and objects endure background encirclement that may result in insignificant wrapping for the object or shape, leading to the classifier's performance reduction when processing such bounding boxes along with some inaccurate detection precision [32].

Object detectors may be divided into single and dual-stage detectors. The dual-stage detectors consist of first proposing regions of interest's stage or RoI, followed by the bounding box stage that will border the proposed region of interest and classify it. As for one-stage detectors, they tend to predict and classify the bounding boxes directly, they are usually faster in processing but less precise than the dual-stage detectors. However, both consist of two neural networks, one for features extraction called a backbone network trained with ImageNet and/or OpenImages, i.e., ResNet and ResNext [23, 33], the other for classification called the head network. Moreover, research showed that in some works unique training experiments were used [34, 35].

In [36], Shaoqing Ren, et al. introduced an R-CNN dual-stage typical example detector, which depends on the types of backbone and head networks that interact differently with the main algorithm called meta-algorithm, for example, the frequent use of Feature Pyramid Networks or FPN as a backbone network that allows the region of interest prediction out of diverse resolution feature maps, which benefits different scaled recognition of objects [37].

Furthermore, single-stage detectors are represented exceedingly by YOLO and SSD [38, 39]. Most single-stage detectors work with images by dividing them into grids for the prediction of classes of objects inside bounding boxes at the same time by using anchors that represent predefined box frames dimensions signifying prior knowledge. In [40], Tsung-Yi Lin, et al., introduced a novel loss focal function in the single-stage detectors that was considered a major improvement since the first phase of the dual-stage detectors produces separate sets of the proposed regions with filtering out most of the negative ones, leaving the second phase with fewer regions. However, single-stage detectors produce larger sets of the proposed regions for inspection to classify boxed objects, which will result in negative regions with incommensurable frequency problems which are solved using a focal loss function altering negative and positive rank in the loss function. In [41], Shifeng Zhang, et al., introduced RefineDet, a single-stage method containing two modules inside, anchor refinement, and object detection. The first module performs

two procedures, filtering out anchors with negative values reducing the classifier's search space, then adjusting indelicately sizes and locations of anchors, providing for the subsequent regressor an improved initialization. As for the second module, it furtherly improves prediction of the classifier for the object labels and the regression accuracy using the inputted refined anchors provided by the first module.

In [42], S. A. Sanchez, et al. conducted a performance matrix to compare object detectors on pre-trained models for speed and performance based on a framework of TensorFlow, including other libraries using free image repositories, COCO of Microsoft, OpenCV, and PASCAL VOC. Object detectors such as YOLO, SSD, R-FCN, R-CNN were used in the study, including various types of extractors, for instance, ResNet, MobileNet, Inception, and VGG16.They concluded that region-based objects detectors such as CNN with both Faster R-CNN and R-R-FCN are faster when it comes to speed precision in real-time processing environments. On the other hand, single-stage object detectors such as SSD and YOLO tend to have difficulties detecting very small objects but faster on average and beats others in precision in size verification and fast extraction of objects. Additionally, YOLO showed an advantage of efficient localization of objects in real-time environments making it a strong competitor detector with high performance.

You Only Look Once or YOLO was first introduced in 2016 by Joseph Redmo, et al. [43], creating a faster detector for objects extracted from processed images on a scale of 45 frames/sec for the basic YOLO model and 155 frames/sec for the fast YOLO. The detector is built out of two basic modules, a CNN backboned from GoogleNet [44] with a twenty-four-layered convolutional neural network subsequent by two layers of the entire connected network. The second module is a uniquely designed loss function. A two-dimensional grid is generated from the neural networks H ×W cells, where H is the vertical height cell number and W is the horizontal width, this partitions the image into small regions with each object present in a part, center or whole-cell region detecting objects by the classifier in these cell regions. In addition, each cell region generates several bounding boxes N with their confidence measurements, giving each box a location based on the base top left point x, y combined with w, h for width and height concerning the size of the image. As for the confidence measurement, it shows object's confidence ratio inside each cell that determines whether an object exists or not. An output probability class map C is generated for each of the cell's regions to indicate the probabilities of an object belonging to a certain object class or multiple object classes. A final prediction output is generated with a H×W×(5N+C) tensor dimensions with 5 being the constant signifying x, y, w, h, and the confidence. Fig. 3 shows the process of detecting objects of 3 frames of a shot in "The Extra Man Trailer" video using YOLO object detector.

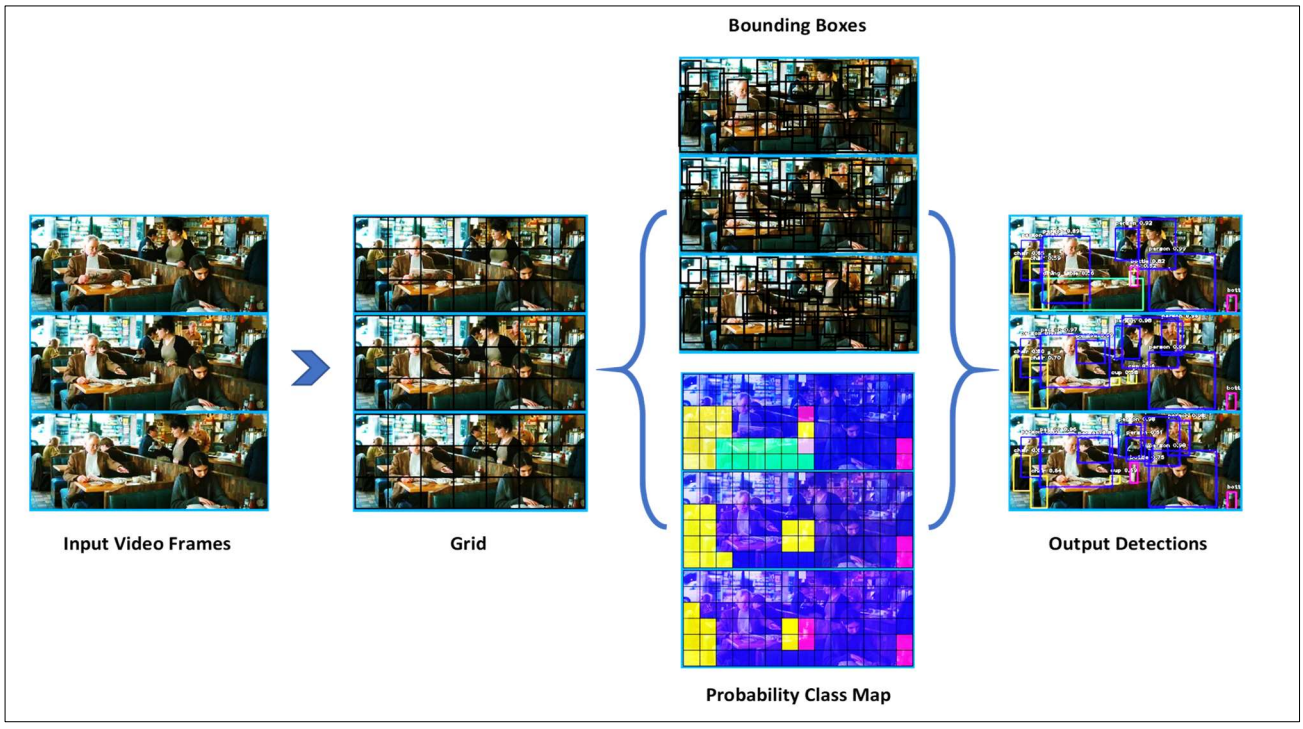

Figure 3. YOLO object detection steps for 3 frames extracted from one video shot in "The Extra Man Trailer" video.

Moreover, YOLO was enhanced in the second version 2017 [45], in some major functions' architecture, one of which was the CNN, it has been restructured to Darknet-19 CNN, using 19 convolutional layers with a batch normalization and 5 pooling layers called max-pooling layers. This enables the network to directly translate and scale the anchoring boxes in the cells rather than predicting the box's base points, width, and height. Therefore, all anchor boxes parameters Aw and Ah are generated using a k-means trained dataset. Another improvement in the second version of YOLOv2 is the usage of fine-graining of features that modify the detections of predicts on a feature map with a size of 13×13, it seems adequate to larger objects, but it also benefits smaller objects localization. This is mainly done by the addition of a layer called passthrough that extracts features via a 26×26 resolution earlier layer and by stacking low- and high-resolution features

adjacently in various channels rather than using special locations, which is correspondent to the ResNet. For example, this transforms a feature map of size 26×26×512 to a feature map with a size of 13×13×2048 that can be concatenated or stacked with the features of origin.

Furthermore, the latest version of YOLOv3 [38] presents a profound layered architecture of Darknet-53 with three feature maps or scales of outputs which gives accuracy higher than YOLOv2 but slower interface speed because of the more dense backbone layers.

However, YOLOv3 which is based on TensorFlow framework was the main object detector for this work and experiments, as it was trained on Microsoft's COCO library using a diversity of objects types that reached 81 objects including added "null object" which represent frames that didn't return any objects by applying the object detector. A state-of-the-art framework called ImageAI [46] was used to implement object detection and keyframes annotations in both indexing and retrieval [22].

E. BOOTLEG VIDEOS RETRIEVAL

A Bootleg video is a term used to describe all kinds of videos pirated from the original copyrighted video materials using several techniques of unauthorized copying as in camcorder piracy from cinema theaters. The copyright of visual materials has been a global issue for a long time, especially for public video streaming services such as YouTube where content-stealing and copyrights misuse is commonly confronted in numerous cases. The public availability of any video stream on the web streaming service is one reason for this problem making it easy for piracy to occur, another major reason is the search engine matching system that most of which relies mainly on the audio content failing to match similar videos upon uploading. This is due to several changes in the video sequence file, making it difficult to detect by audio, or any other non-content-based video technique. Another problem is the high computational cost of most frame-by-frame matching techniques, finally, the sensitivity of these currently used content-based video matching techniques to the bootleg videos and the change in the visual contents of the pirated videos deceiving detectors and matching algorithms [3].

Bootleg videos have many types however, the most common type referred to as a bootleg video is the camcorder recorded videos. Other types include resolution or dimension edited videos with altered video dimensions and resolution. Another type is presented by the speed edited video files, where the video is fast speeded or low speeded, and the third type is flipping the video canvas horizontally to deceive content video detection and matching for the public streaming web services that rely mainly on frame-by-frame matching [47].

In this study, a new approach is introduced and tested on a large video dataset with minimal computational cost and efficient video matching that was examined and showed transparency to the alteration and manipulation of the morphological appearance for any video in the process of matching the manipulated video against its original video source. However, the study only focusses on commonly used manipulations for bootleg videos such as dimensions, speed, flipping, and camcorder altered videos. Fig. 4 shows examples of bootleg videos used in this work.

Figure 4. Example of Manipulated Bootleg Videos Keyframes of an Original Video of "The Extra Man" (Movie trailer on YouTube).

IV. EXPERIMENTAL ANALYSIS

A. VIDEO INDEX DATASET

Many public datasets are in existence for the purpose of video retrieval and building a content-based video search engine most of which include extracted and stored feature vectors, some semantic objects, segment-leveling annotation as in [8, 48-50].

However, processes such as facial recognition and setting multiple categories annotations which signify different human activities are very computationally expensive, requiring exhaustive training, making it very difficult to prepare, other problems include unsuitable all video genders, some datasets only relevant to human involved video genders. Although, datasets of public video streaming services indexe video files for other purposes that include video annotation, learning/training systems, video classification, and additional areas of computer vision with slight attention for public video URL links indexing for searching and retrieval.

Moreover, this study offers a new YouTube crawled large-scaled dataset created and based on the research work of [22], containing 1088 video records representing a sum of 65+ hours of video, extracting video shots of 113,502+ segments, and a total extracted keyframe images of 677,004+ marked/unmarked frames. In addition, object detection and marking the frames was done by using 80 different RetinaNet based objects trained using MS-COCO dataset [51] on top of ImageAI [46] platform, which is a state-of-the-art open-sourced python built library that offers trained models that have been mined from ImageNet-1000 with 1000 diverse objects applied on platforms of ResNet, Inception-v3, DeneseNet, and SqueezeNet, and finally, imposing a video classification criterion on the indexed dataset for the purpose of increasing efficiency and decreasing the time of retrieving queries by example video. Thus, a two-phased technique was introduced classifying video records based, firstly, on object aggregation as the maximal occurred object extracted from the video shots in each video sequence and, secondly, based on event aggregation where the video records from a certain category or dominant object set are divided into groups according to the number of shots extracted from each video. The study shows that 58 different categories were classified and indexed, inside of each set there were 9 event aggregated groups representing number of shots in each video record in closed intervals of [1,100], [101,200], [201,300], [301,400], [401,500],

[501,1000], [1001,2000], [2001,3000], and [3001,4000]. The gathered video categories and various representations of genders are to make sure that the retrieval system outputs and content analysis proposed in this work will be qualitatively evaluated. Fig. 5 shows the steps of the dataset video indexing and video records classification.

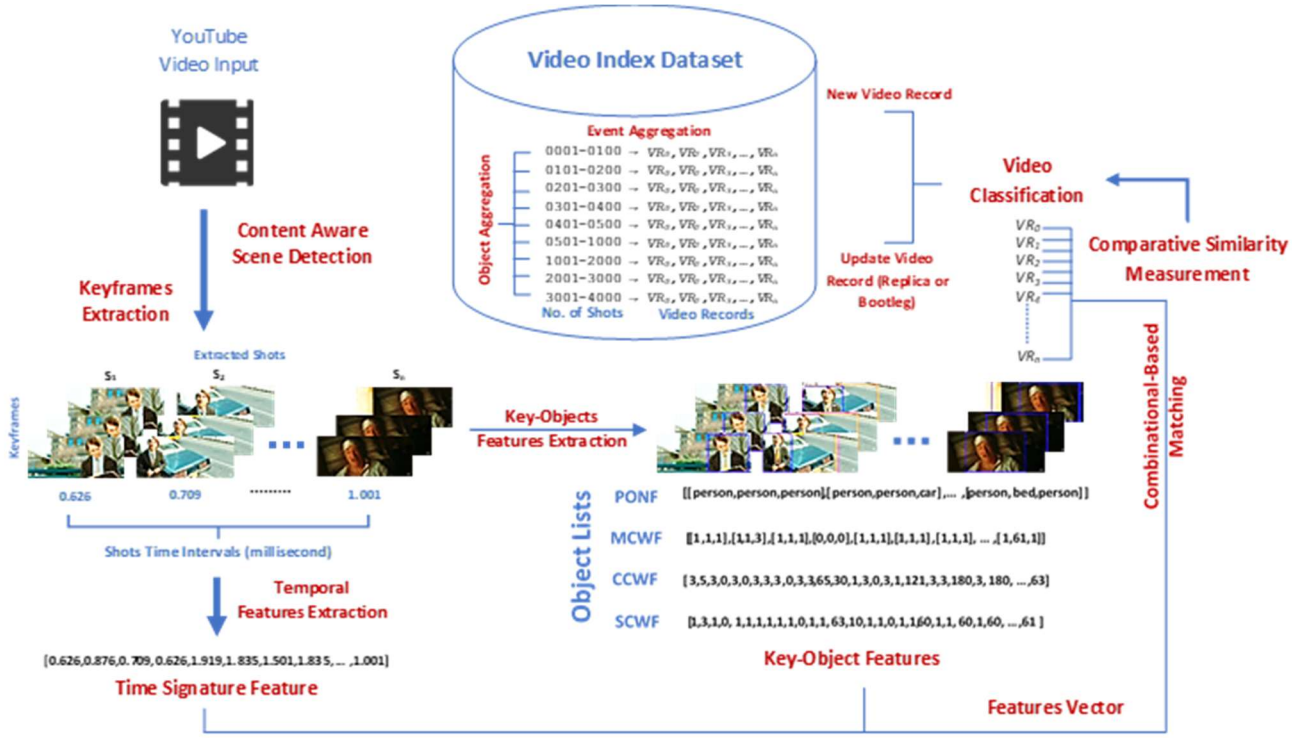

Figure 5. Video Indexing Steps

B. MATCHING & OPTIMIZATION

The matching process is performed on the features vector extracted from the user video file query acquired from the client-side and sent to the retrieval system on the server, features are then processed using two-phased combinational-based matching searching at first the temporal feature set and the resulting output temporal similar list (TSIM) are then matched on the key-objects feature set to generate a key-object similar list (OSIM) from which the primary similar list is generated with binding retrieved video records and URLs for the next step of optimization which removes redundancies. Three different similarity metrics were employed to compare and select the best combination-based search for each phase using Cosine similarity representing angular distance similarities, Minkowski distance similarity which represents distance similarities, and the Jaccard similarity coefficient which is a representation of the Intersection over Union ratio. However, experiments were conducted to evaluate each similarity metric performance on each feature vector, and in the following text detailed description and results are given.

Cosine similarity was selected as best performance for temporal feature vector's combinational-based matching employing an experiment conducted earlier in this research work to evaluate different similarity metrics based on performance using different types of video features such as original YouTube downloaded videos as a gold standard against manipulated bootleg videos created from the originals [22]. All videos were selected and processed from the video index dataset and the manipulation included other YouTube replicas on YouTube for the original gold standard video, video bootleg types included two types of resolution/dimension edited videos (small 320×240 & large 1980×1080), two types of speed edited videos with (75% & 125%) slow and fast motion, a horizontally flipped frames video, and lastly a camcorder video type.

Furthermore, objects/concepts feature vector matching was experimented on three different similarity metrics (Cosine, Minkowski, and Jaccard) to evaluate comparative similarity measurements for the retrieval system according to performance, the experiment was conducted to evaluate and select the best performance key-objects combination similarity metric from the three in terms of accuracy for comparing the similarity between two vectors. Additionally, two groups of queries were used to evaluate the retrieval system's performance, the first experimental group involved non-semantic-based video queries which randomly selected 5 distinct types of video queries: first would be the original rank and it consists of 5 original video files from each gender of the four genders (movie trailers, music clips, news, and sports) in the video index dataset with a total of 20 original video files, the second rank included bootleg videos that involve (dimension edit, speed edit, flipped, and camcorder) videos leading to a total of 100 video results in 20 query ranks. Additionally, each query must return 5 true-positive (TP) results from the video index dataset including the original video and 4 bootleg videos scattered in the dataset.

Moreover, the second group included another 100 randomly selected non-semantic-based video queries, 25 queries from each gender in the video index dataset also representing (movie trailers, music clips, news, and sports) video records.

However, all queries' temporal feature vectors were matched using the cosine similarity metric and the key-object feature vectors were matched using the other similarity metrics represented by (Cosine-Cosine (C-C), Cosine-Minkowski (C-M) and Cosine-Jaccard (C-J)) respectively. The combinational-based matching performance in many related works [52-57] was evaluated in terms of precision, recall, and F measure (F1) parameters, all of which depend on false-negative (FN) and false-positive (FP) query retrievals. Precision, however, is the proportional retrieved results set that is relevant and determined by dividing relevant set or true-positive (TP) over the retrieved set (TP) and false-positives (FP). This determines how many junk results the retrieved list has. The recall is the fraction of all videos found that are relevant in the retrieved set and is determined by dividing the relevant set (TP) over the retrieved set including the number of missed results or false-negatives (FN), this determines how many relevant videos the retrieval system missed in the retrieval process. The combined measure (F) which assesses the tradeoff between precision and recall called weighted harmonic mean that uses the balanced F1 measure computed from precision P and recall R is as follows:

$$F1 = \frac{2PR}{(P+R)} \tag{5}$$

This is due to that on average we find precision dropping and recall increasing which makes it difficult to measure the tradeoff between the two and the only way is to use combined measurement as F measure. However, other measures are used such as accuracy, which is the proportion between the number of correctly retrieved queries with the total number of queries. Yet, accuracy is not taking into consideration the data distribution, which might lead to an incorrect evaluation, for example, if we have a simple binary classifier whose task is to classify 100 data samples 90 of which are negatives and only 10 are positive, and the classifier only predicts the negative values, thus, 10 false-negatives are predicted leading to an accuracy of 90%, which is a false and misleading conclusion, while F1 will score 0 as evaluation for the classifier due to 0 score of recall as a part of F1. Moreover, accuracy is commonly used when classes and their distribution are similar in weight and the true positives and true negatives are more significant than false negatives and false positives for whom in their case F1 score is needed. In this study, we have an imbalanced classification problem in which false negatives and false positives are more important leading to evaluating and comparing the proposed approach using precision, recall, and F1 measure is crucial.

Moreover, as described earlier the experiment for comparative evaluation between similarity metrics including cosine, Minkowski, and Jaccard methods in pursue of estimating the performance of the retrieval system in terms of precision, recall, and F1 measure was carried out. This experiment showed for the first testing group that the average precision for Cosine-Cosine metric was recorded best performance with 99.2%, as well as Cosine-Minkowski with 99.2%, while Cosine-Jaccard performed less with 97.5%. As for the average recall, both Cosine-Cosine and Cosine-Jaccard performed best with 96.7% while Minkowski performed poorly with 71%. However, to eliminate the tradeoff between precision and recall the average F1 measure showed best with Cosine-Jaccard metrics with 97%, while Cosine-Cosine came in second with a small difference of 96.7%, while Cosine-Minkowski came last with 82.7%. Table 1 shows the comparative results and performance for temporal-concept combination matching used in the retrieval system.

Table 1. Comparison of precision, recall, and F1 performances for Temporal – Concept's combination matching using 3 similarity two-phased metrices for 20 sets of videos, each set has 5 video records, one represents original video and 4 represents different types of bootleg videos edited from the original video altering video dimensions, speed, horizontally reversed frame (flipped), and camcorder recorded video using a camera.

#		20 Video Queries from 1088 Index	Original			Dimensions Edit			Speed Edit			Flipped			Camcorder			False (FP)			Evaluation								
			Miss (FN)			Miss (FN)			Miss (FN)			Miss (FN)			Miss (FN)						Precision			Recall			F1		
			C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J	C-C	C-M	C-J
1	Movie Trailers	Extra Man	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
2		Escape Planet Earth	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
3		Horrible Bosses	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
4		Jump Street	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	**1**	0	0	0	0	1.00	1.00	1.00	0.83	0.71	1.00	0.91	0.83	1.00
5		Let Me In	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	**1**	0	0	0	0	1.00	1.00	1.00	0.83	0.71	1.00	0.91	0.83	1.00
6	Music Videos	Bus Song	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	**1**	0	0	0	0	1.00	1.00	1.00	0.83	0.71	1.00	0.91	0.83	1.00
7		Exercise Song	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	**1**	1.00	1.00	0.83	1.00	0.71	0.83	1.00	0.83	0.83
8		Kids Sorry	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	**1**	1.00	1.00	0.83	1.00	0.71	0.83	1.00	0.83	0.83
9		Reino	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
10		Truth Hurts	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
11	News Videos	Bring Dogs	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
12		Spain Lottary	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
13		People Fridge	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	**1**	1.00	1.00	0.83	1.00	0.71	0.83	1.00	0.83	0.83
14		Passenger Trains	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
15		X Rays Vision	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	**1**	**1**	0	0.83	0.83	1.00	0.83	0.63	1.00	0.83	0.71	1.00
16	Sports Videos	World Cup	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
17		Golf	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	**1**	0	0	0	1.00	1.00	1.00	1.00	0.71	0.83	1.00	0.83	0.91
18		Basketball	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
19		Sports Mix	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
20		TUA	0	0	0	0	0	0	0	**1**	0	0	0	0	0	**1**	0	0	0	0	1.00	1.00	1.00	1.00	0.71	1.00	1.00	0.83	1.00
		Average																			**0.992**	0.992	0.975	**0.967**	0.710	**0.967**	**0.979**	0.828	0.970

C-C is **Cosine-Cosine**, **C-M** is **Cosine-Minkowski**, and **C-J** is **Cosine-Jaccard** similarity metrics for two phased Temporal-Concept combination matching using video classification.

Furthermore, the second part of the experiment involved 100 randomly selected non-semantic-based video queries 4 test groups, which included 25 representative queries of each video gender in the video index dataset where the best average precision recorded 97.9% for Cosine-Jaccard followed by Cosine-Cosine with 97.2% and Cosine- Minkowski came in the

last with 96.7%. As for the average recall, all of the three approaches performed best with 100%. The F1 measure was recorded best for Cosine-Jaccard with 98.3% followed by Cosine-Cosine with 98.1%, and Cosine- Minkowski came last with 97.5%. Table 2 shows the comparison of precision, recall, and F1 measure performances for temporal-concept combination matching for the retrieval system.

Table 2. Comparison of precision, recall, and F1 performances for Temporal – Concept's combination matching using 3 two-phased similarity metrics for 100 video queries divided into 4 video groups, 25 queries for each group representing different genders of movie trailers, music, news, and sports videos respectively.

100 Video Queries from 1088 Index	Cosine-Cosine			Cosine-Minkowski			Cosine-Jaccard		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Movie Trailers (25 Video Queries)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Music Videos (25 Video Queries)	0.95	1.00	0.97	0.93	1.00	0.94	0.94	1.00	0.95
News Videos (25 Video Queries)	0.94	1.00	0.96	0.94	1.00	0.96	0.98	1.00	0.99
Sports Videos (25 Video Queries)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Total Average (100 Video Queries)	0.972	1.000	0.981	0.967	1.000	0.975	**0.979**	**1.000**	**0.983**

C. CONTENT-BASED VIDEO INDEXING & RETRIEVING TECHNIQUES COMPARISON

In this section, a comparison for various content-based video indexing and retrieval systems with the proposed approach is introduced. Likewise, a brief description for each technique and evaluation in regards to precision, recall, and F1 measure are given to compare with the proposed approach.

In [52], Kan, et al. introduced a semi-supervised hashing via kernel hyperplane learning technique with accurate retrieval performance of images on a small scale of image dataset which increases matching numerous similar images. However, hashing functions are rested separately which leads to inaccurate results in large-scale datasets. In [53], GuoKehua, et al., introduced an automatic learning and social annotating technique using NoSQL-based semantic storage indexing and MapReduce-based heterogeneous multimedia retrieval with images, video, audio, and text documents queries on large datasets altogether with low cost of input/output. Nevertheless, performance degrades in the case of large data input into the system. In [54], Fernandez-Beltran, et al. introduced a technique of retrieval using manifold ranking and local regression and global alignment with features including K topics and latent topic ranking. Yet, there is an overfitting problem in which multimodel data cannot be supported. In [55], Han, et al. introduced a lossless matching algorithm that accelerates product computation for Fisher Vector high dimensions which does not need discriminatory training that produces good performance in video retrieval. Though, there is a very large time overhead. In [56], Jyothi, et al. introduced a natural flower video retrieval technique with multiclass support vector machine retrieval algorithm employing no indexing only training the DCNN altogether with very good accuracy and reduced complexity. Yet, it requires excessive training and large training samples and special in only flower videos with just 30 flowers classes. In [57], Asha, et al. introduced an online video URL query approach with multiple features technique using color distributions, texture & motion, and binary patterns feature vector with no indexing criteria and a Euclidean distance retrieval algorithm and acceptable performance on a small scale experimental dataset of 40 videos in 4 categories. Table 3 shows the comparison between all mentioned techniques against the proposed technique in this work in terms of precision, recall, and F1 measure performances.

Table 3. Comparison of precision, recall, and F1 performances against different image, video, or both retrieval techniques.

#	Techniques	Features & Algorithms				Results & Evaluation		
		Query Type	Feature Vector	Indexing and/or Retrieval	Dataset	Precision	Recall	F1
1	Semi-Supervised Kernel Hyperplane Learning [55], (**2014**)	• Images (from each dataset 10% randomly selected images)	4 types of features: -320-D GIST -225-D Color Moment -73-D Edge Direction Histogram - 128-D Wavelet Texture	Semi-supervised Hashing via Kernel Hyperplane Learning.	CIFAR-100 (60,000 images on 20 classes) NUS-WIDE dataset (269,648 images and 81 concepts annotations)	0.91	0.923	0.916
2	Semantic-Based Heterogeneous Multimedia Retrieval [56], (**2015**)	• Images • Video • Audio • Text docs	Automatic Learning Social Annotating	NoSQL-based Semantic Storage Indexing MapReduce-based Heterogeneous Multimedia Retrieval	Large scale experimental dataset 20,000 images, 10,000 videos, 10,000 audios, and 10,000 text docs on 10 category classes.	0.927	0.947	0.937
3	Latent Content-Based Video Relevance & Feedback Retrieval Approach [57], (**2016**)	• Query samples inside dataset. • Query samples outside dataset	Extract K topics Latent Topic Ranking	Retrieval using Manifold Ranking and Local Regression and Global Alignment	TRECVID 2007 database only 17 Classes were selected with 8974 samples.	0.946	0.961	0.953
4	Video Retrieval and Fast Fisher Vector Products [58], (**2017**)	50 ranked query videos on 1,000 videos selected dataset.	Fisher Vector built on CNN features.	lossless matching algorithm accelerates product computation for Fisher Vector high dimensions.	TRECVID MED13/14 (around 23,000 videos with 20 complex events). Columbia Consumer Videos (4,658 videos 20 semantic categories). 1,000 videos are used only for retrieval experiment.	0.97	0.988	0.979
5	Deep Learning for Retrieval of Natural Flower Videos [59], (**2018**)	• Natural Flower Video	Keyframes Segmented flowers Flower Gradient	No indexing only training the DCNN Multiclass Support Vector Machine Retrieval Algorithm.	Experimental dataset of 2600 flower videos of 30 flowers classes.	0.989	0.994	0.991
6	Content-Based Video Retrieval System using Multiple Features [60], (**2018**)	• Online Video URL	Color Distributions Texture & Motion Binary Patterns	No indexing criteria Retrieval using Euclidean distance.	Small scale experimental dataset of 40 videos in 4 categories.	0.80	0.80	0.80
7	**Proposed Content-Based Video Search Engine Retrieval System (100 Queries of YouTube Videos)**	• Video	Temporal feature Key-object feature	Each video record in the index has temporal and key-objects features vector + Keyframes representing video shots. Retrieval using Combinational-based matching using Cosine, Minkowski, and Jaccard similarity metrics.	Experimental Dataset of 1088 video records in 58 object categories each of which divided into 9 groups. 65 hours of video. 338,502 keyframes images.	**0.979**	**1.000**	**0.983**
	Original and Bootleg Video Retrieval (20 Ranks Queries 5 Videos/rank)					**0.992**	**0.967**	**0.970**

V. CONCLUSIONS

In this paper a novel and effective technique for content-based video search engine altogether with bootleg videos retrieval system, evaluated on a large-scale video index dataset of 1088 video records. The objective is set to be an enhancement for currently used web and text-based video search engines to enable content-based video search and similarity matching in a large-scale video index crawled and indexed through public video streaming services such as YouTube on the world wide web. Another objective is to service the copyright dilemma for misusing copyrighted video materials, especially bootleg videos, and detecting them before they are uploaded relating them to the original video. There were some challenges regarding content-based video search engines that have been overcome in this research, one of which is indexing a large-scaled video, index dataset gathering a collection of a large representation for major video genders of minimum space consumption and sensible retrieval accuracy via a lessening computational cost and time-efficient extraction for feature vector. Another challenge is providing a retrieval system that supports nonsemantic-based video querying with bootleg video retrieval for major bootleg video manipulations such as dimensions, speed, flipping editing altogether with camcorder captured videos, indexing more than 1088 videos with 65 hours of videos, consisting of around 113502 shots that contain 338502 keyframes. Qualitative evaluation of the retrieval system using the proposed feature vector of temporal and key-objects/concepts applying a combinational matching algorithm was accomplished using more than 200 nonsemantic-based video queries with a retrieval precision for normal videos group of 97.9% and retrieval recall of 100% combined by F1 measure to be 98.3%, as for bootleg videos a retrieval precision of 99.2% and retrieval recall of 96.7% combined by F1 measure to be 97.9%. Conclusion can be drawn that this technique will help to emphasize traditional commercial and non-commercial text-based video search engines to provide a further transparent searching tool for nonsemantic-based queries such as query by example video. Furthermore, extensive testing was done on the indexing and retrieval systems comparing them to state-of-the-art techniques and tools, applying multiple similarity metrics that proved to be efficient and reliable in multiple experimental models. Also, the video index dataset was diverse containing various video genres representations which ensured evaluation based on qualitative outputs for the retrieval system.

Future work will include testing more various tools and systems that provide shot boundary detection and video annotation and objects/concepts detectors in process of building more faster and suitable tools for this research to maximize efficiency and minimize query segmentation time providing a faster performance on query analysis and feature extraction. Although, the expansion of the video index dataset will be more than a million video records for streaming videos on YouTube, DailyMotion, Vimeo, and other public streaming services over the world wide web.

Finally, compensations for the video retrieval system showed fast performance accurate similarity video retrieval using combination-based matching algorithms with different similarity metrics altogether with video index dataset classification based on objects and events. The experiments conducted showed high performance in nonsemantic-based query time on a very large-scaled dataset.

References

[1] A. S. Adly, I. Hegazy, T. Elarif, M. S. Abdelwahab, "Issues and challenges for content-based video search engines: A survey," *Proceedings of the 2020 21st IEEE International Arab Conference on Information Technology (ACIT)*, 2020, pp. 1–18.

[2] "YouTube for Press." [Online]. Available at: https://blog.youtube/press/.

[3] A. R. Baloch, U. A. Kashif, K. G. Chachar, and M. A. Solangi, "Video Copyright detection using high level objects in video clip," *Sukkur IBA J. Comput. Math. Sci.*, vol. 1, no. 2, p. 95, 2017.

[4] A. Mazaheri, B. Gong, and M. Shah, "Learning a multi-concept video retrieval model with multiple latent variables," *Proceedings of the 2016 IEEE International Symposium on Multimedia, ISM 2016*, 2017, pp. 615–620.

[5] N. Garcia, "Temporal aggregation of visual features for large-scale image-to-video retrieval," *Proceedings of the 2018 ACM International Conference on Multimedia Retrieval ICMR 2018*, 2018, pp. 489–492.

[6] N. Garcia and G. Vogiatzis, "Dress like a star: Retrieving fashion products from videos," *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2017, pp. 2293–2299.

[7] M. Mühling *et al.*, "Deep learning for content-based video retrieval in film and television production," *Multimed. Tools Appl.*, vol. 76, no. 21, pp. 22169–22194, 2017.

[8] E. G. Ortiz, A. Wright, and M. Shah, "Face recognition in movie trailers via mean sequence sparse representation-based classification," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3531–3538.

[9] G. De Oliveira Barra, M. Lux, and X. Giro-I-Nieto, "Large scale content-based video retrieval with LIvRE," *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, 2016, pp. 1–4.

[10] L. Rossetto *et al.*, "IMOTION – a content-based video retrieval engine," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 8936, pp. 255–260.

[11] S. S. Gornale, A. K. Babaleshwar, and P. L. Yannawar, "Analysis and detection of content based video retrieval," *Int. J. Image, Graph. Signal Process.*, vol. 11, no. 3, p. 43, 2019.

[12] G. S. N. Kumar, V. S. K. Reddy, and S. Srinivas Kumar, "High-performance video retrieval based on spatio-temporal features," *Microelectronics, Electromagnetics and Telecommunications*, 2018, pp. 433–441.

[13] R. Gaikwad and J. R. Neve, "A comprehensive study in novel content based video retrieval using vector quantization over a diversity of color spaces," *Proceedings of the International Conference on Global Trends in Signal Processing, Information Computing and Communication, ICGTSPICC 2016*, 2017, pp. 38–42.

[14] Y. Yang, Z. Ma, Z. Xu, S. Yan, and A. G. Hauptmann, "How related exemplars help complex event detection in web videos?," *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2104–2111.

[15] Z. Z. Lan, Y. Yang, N. Ballas, S. I. Yu, and A. Haputmann, "Resource constrained multimedia event detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8325, LNCS, no. PART 1, pp. 388–399.

[16] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2593–2600.

[17] S. Yu *et al.*, "Informedia@TRECVID 2014 MED and MER," *TRECVID Video Retrieval Evaluation Workshop, NIST*, 2014. [Online]. Available: https://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.14.org.html.

[18] P. A. Zeitschrift, S. N. Band, P. Link, and E. Dienst, "Étude comparative de la distribution florale dans une portion des Alpes et du Jura," *Bull. la Société Vaudoise des Sci. Nat.*, vol. 37, pp. 547-579, 2013. (in French)

[19] M. Almousa, R. Benlamri, and R. Khoury, "NLP-enriched automatic video segmentation," *Proceedings of the International Conference on Multimedia Computing and Systems -Proceedings*, 2018, pp. 1–6.

[20] B. Castellano, "PySceneDetect." [Online]. Available at: https://pyscenedetect.readthedocs.io/en/latest/. [Accessed: 04-May-2019].

[21] "FFmpeg." [Online]. Available at: https://ffmpeg.org/. [Accessed: 04-May-2019].

[22] A. S. Adly, I. Hegazy, T. Elarif, and M. S. Abdelwahab, "Indexed dataset from YouTube for a content-based video search engine," *Int. J. Intell. Comput. Inf. Sci.*, vol. 21, no. 1, pp. 196–215, Feb. 2021.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020.
[24] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos, "Scene detection in videos using shot clustering and sequence alignment," *IEEE Trans. Multimed.*, vol. 11, no. 1, pp. 89–100, 2009.
[25] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1217–1229, 2008.
[26] Z. Mehmood, T. Mahmood, and M. A. Javid, "Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine," *Appl. Intell.*, vol. 48, no. 1, pp. 166–181, Jan. 2018.
[27] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully convolutional localization networks for dense captioning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4565–4574.
[28] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 3, pp. 2048–2057.
[29] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings*, 2015, pp. 1–15.
[30] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," *Advances in Neural Information Processing Systems*, 2014, vol. 3, no. January, pp. 2204–2212.
[31] J. L. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings*, 2015, pp. 1–10.
[32] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9396–9405.
[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 5987–5995.
[34] R. Zhu *et al.*, "Scratchdet: Training single-shot object detectors from scratch," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2263–2272.
[35] Z. Shen, Z. Liu, J. Li, Y. G. Jiang, Y. Chen, and X. Xue, "Object Detection from Scratch with Deep Supervision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 398–412, 2020.
[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
[37] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, 2014.
[38] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv Prepr. arXiv1804.02767*, vol. 1804.02767, pp. 1–6, Apr. 2018.
[39] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 21–37.
[40] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2020.
[41] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.
[42] S. A. Sanchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework," in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 844, no. 1, pp. 1–12.
[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
[44] C. Szegedy *et al.*, "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
[45] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, pp. 6517–6525.
[46] M. Olafenwa and J. Olafenwa, "ImageAI, an open source python library built to empower developers to build applications and systems with self-contained Computer Vision capabilities," *Github*, 2018. [Online]. Available at: https://github.com/OlafenwaMoses/ImageAI.
[47] P. Mehta, S. Maheshkar, and V. Maheshkar, "An effective video bootleg detection algorithm based on noise analysis in frequency domain," *Commun. Comput. Inf. Sci.*, vol. 1147 CCIS, pp. 227–238, 2019.
[48] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition Terms of Use Creative Commons Attribution-Noncommercial-Share Alike 3.0 HMDB: A Large Video Database for Human Motion Recognition," *IEEE*, 2011.
[49] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 human actions classes from videos in the wild," *CoRR, abs/1212.0402*, vol. abs/1212.0, pp. 1–7, 2012.
[50] H. Idrees *et al.*, "The THUMOS challenge on action recognition for videos 'in the wild,'" *Comput. Vis. Image Underst.*, vol. 155, pp. 1–23, 2017.
[51] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693, LNCS, no. PART 5, pp. 740–755.
[52] M. Kan, D. Xu, S. Shan, and X. Chen, "Semisupervised hashing via kernel hyperplane learning for scalable image search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 704–713, 2014.
[53] GuoKehua, PanWei, LuMingming, ZhouXiaoke, and MaJianhua, "An effective and economical architecture for semantic-based heterogeneous multimedia big data retrieval," *J. Syst. Softw.*, vol. 102, pp. 207–216, 2015.
[54] R. Fernandez-Beltran and F. Pla, "Latent topics-based relevance feedback for video retrieval," *Pattern Recognit.*, vol. 51, pp. 72–84, 2016.
[55] X. Han, B. Singh, V. I. Morariu, and L. S. Davis, "VRFP: On-the-fly video retrieval using web images and fast fisher vector products," *IEEE Trans. Multimed.*, vol. 19, no. 7, pp. 1583–1595, 2017.
[56] V. K. Jyothi, D. S. Guru, and Y. H. Sharath Kumar, "Deep Learning for Retrieval of Natural Flower Videos," in *Procedia Computer Science*, vol. 132, pp. 1533–1542, 2018.
[57] D. Asha, Y. Madhavee Latha, and V. S. K. Reddy, "Content Based Video Retrieval System Using Multiple Features," *Int. J. Pure Appl. Math.*, vol. 118, no. 14, pp. 287–294, 2018.

***AHMAD SEDKY ADLY,** a Lecturer of Computer Science in the Department of Computer Science, Faculty of Information Technology, Misr University for Science & Technology (MUST). Research interests include Content-Based Video Search Engines, Motion Analysis, Computer Vision, Digital Image Processing, Computer Graphics, Bioinformatics.*

***Assistant Prof. Dr. ISLAM HEGAZY,** an Assistant Professor of Computer Science in Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University (ASU). Dr. Islam has more than 18 years of experience in research and teaching in many fields in the fields of Computer Science. He holds a Ph.D. degree from the University of Calgary, Canada. An M.Sc. and B.Sc. degrees, Faculty of Computer and Information Sciences, ASU. His research interests focus on network, security, cloud computing, image processing, and AI. Dr. Islam acquired several managerial skills as the Director of the Scientific Computing Center, ASU, and the coordinator of the Software Engineering Credit Hours program, Faculty of Computer and Information Sciences.*

***Prof. Dr. TAHA ELARIF**, a Professor of Computer Science in Computer Science Department, Faculty of Computer and Information Sciences, Ain Shams University. Prof. Taha has more than 45 years of experience in research and teaching in many fields in the fields of Computer Science. He holds a PhD degree from the Université de Technologie de Compiègne (UTC), France. A Master in Computer Engineering Faculty of Engineering and a Diploma in Computer Science, Institute of Statistical Study & Research, Cairo University. Prof. Taha acquired several managerial positions as the Vice Dean for Community Service and Environmental Affairs and Chairman of Department Council From for 4 cycles. Major fields of scientific research: Computer Graphics, Image Processing, and Artificial Intelligent.*

***Prof. Dr. M. S. Abdelwahab**, a Professor of Computer Science in Computer Science Department, Faculty of Information Technology, Misr University for Science & Technology (MUST). Prof. Abdelwahab has more than 50 years of experience in research and teaching in many fields in the fields of Computer Science. He holds a Dsc degree in Nuclear physics from the Karlsruhe University of Applied Sciences, Germany. Prof. Abdelwahab was former Dean and founder of Faculty of Computer and Information Science Ain Shams University for 5 years, and the former Dean of Faculty of Information Technology, MUST for more than 7 years, he is now holding the position of CEO Consultant and Vice President for Information Technology in MUST. Major fields of scientific research: Triangle-Quad Meshes, Face-Based Non-Split Connectivity, Non-Split Connectivity Compression, Quad, Efficient Connectivity Encoding, Edge breaker Compression Algorithm.*