AUVANA: An Automated Video Analysis Tool for Visual Complexity

A Preprint

Emad A. Alghamdi
King Abdulaziz University
Jeddah, Saudi Arabia
eaalghamdi@kau.edu.sa

Eduardo Velloso
School of Computing and Information Systems
The University of Melbourne
Melbourne, Australia
eduardo.velloso@unimelb.edu.au

Paul Gruba
School of Languages and Linguistics
The University of Melbourne
Melbourne, Australia
p.gruba@unimelb.edu.au

May 8, 2021

Abstract

Visual complexity is widely considered to be an important variable underlying visual perception. While videos have become versatile in their use of visual imagery, surprisingly, little research has been devoted to understanding the impact of visual complexity. In this paper, we present Automated Video Analysis (AUVANA) software, an open-source tool for extracting, computing, and visualizing visual complexity in digital videos. Through leveraging more sophisticated computer vision and video processing algorithms, AUVANA automatically extracts and computes 78 video visual complexity indices. Results of explanatory analyses demonstrated that rather than a unitary construct video visual complexity is more likely a multidimensional and multifaceted phenomenon. We conclude the paper with a discussion about the potential applications of the software.

Keywords Visual complexity · Video difficulty · Multimedia

1 Introduction

Research on visual complexity has been carried out in many areas, including cognitive science, marketing, psychology, human-computer interaction, aesthetics, and psychobiology (e.g., Braun, Amirshahi, Denzler, & Redies, 2013; Pieters, Wedel, & Batra, 2010; Tuch, Bargas-Avila, Opwis, & Wilhelm, 2009). That being said, the notion of visual complexity remains elusive and hard to pin down. Originally proposed for static images, visual complexity broadly refers to the amount of detail or intricacy contained within an image (Snodgrass & Vanderwart, 1980). Perceiving a visual stimulus more or less complex is also suggested to be influenced by several other factors, including, for example, the type and quantify of elements it contains, their spatial distribution or layout, variety of colors, and so forth (Palumbo, Makin, & Bertamini, 2014). Heaps and Handel (1999) contended that visual complexity corresponds to the degree of difficulty people encounter when describing a visual stimulus. Moreover, visual complexity can be estimated at a wide range of levels; from pixel arrangement to semantics (Da Silva, Courboulay, & Estraillier, 2011). Conversely, "any study of complexity faces the problem of selecting a definition and a measure of visual complexity" (Palumbo et al., 2014, p.3).

For measuring visual complexity, researchers have utilized qualitative and quantitative approaches. A qualitative approach often involves human judgment of visual complexity using a predefined subjective scale. One drawback of such subjective ratings is that they are costly to obtain, less consistent, and cannot be generalized to other visual stimuli. On the other hand, quantitative measures seek to provide a reliable and consistent numerical value that corresponds

to some behavioral data of complexity in visual stimuli. A popular method for computing visual complexity is image compression. It is generally assumed that complexity and compression are intimately related concepts. That is, an image of complex visual stimuli requires a large file size when compressed than an image of simple stimuli (Salomon, 2004; Sayood, 2017). Using this rationale, several studies have found a correlation between compressed image sizes and human perception of visual complexity (e.g., Donderi & McFadden, 2005).

While a great deal of visual complexity research is devoted to static images, little research has been conducted on visual complexity in videos. The dynamic and multimodal nature of videos makes measuring visual complexity in videos a very challenging task. In this paper, *video visual complexity* refers to all visual and spatiotemporal attributes of video that make processing visual imagery more effortful and challenging. Rather than being a single construct, we hypothesize that video visual complexity is more likely a multilayered and multidimensional construct.

It should be noted that while it is generally assumed that visual complexity has a detrimental effect on perception and cognition, there is some suggestive evidence that complexity may have a positive or nonmonotonic effect on perception. In a recent study, Ellis and Turk-Browne (2019) observed that while increasing information overload, complexity enhances long-short memory and perceptual sensitivity of video clips. The researchers attributed the positive effect to the fact that complex videos afford richer and distinctive audiovisual representations, which may help in forming long-term memories. Though it remains unclear how and what attributes of complex videos are responsible for this better scaffolding.

The current explanatory study contributes to the existing literature on video visual complexity through developing an open-sourced tool—Automated Video Analysis (AUVANA)[1]—for extracting and computing a wide range of visual complexity indices. To the best of our knowledge, there is no freely available tool for automatically measuring visual complexity in videos. Researchers interested in exploring visual and spatial-temporal complexity in videos can use the tool to automatically extract, compute, and visualize numerous visual complexity indices.

In the next sections, we first describe the main functionalities of the tool and present its interfaces. Then, we present the video visual complexity indices, describe their underlying assumptions, and explicate how they are extracted and computed. Finally, we report the finding of an explanatory analysis on the proposed video visual complexity indices.

2 The Development of AUVANA

The primary goal of developing AUVANA software was to make analyzing visual complexity in videos accessible to users who are not familiar with programming or have expertise in computer vision and image processing. Therefore, the tool's graphical user interfaces (GUIs) were designed to be intuitive and user-friendly. Another key advantage of the tool is that it stores user data locally and requires no connection to the Internet.

For developing the application, we used the open-sourced software framework Electron [2] with the help of the Vue Electron Builder [3]. We chose the Electron framework for developing our tool because it allows for the development of desktop applications that are compatible with Mac, Windows, and Linux. The tool's main user interfaces are shown in Figure 1, Figure 2, Figure 3, and Figure 4.

For processing videos and extracting and computing visual complexity indices, we used Python 3.8 and several open-sourced libraries (more information will be presented below). The first version of AUVANA computes 78 visual complexity indices at four levels of granularity: video, shot, keyframe, and frame. A shot is a continuous sequence of frames taken from a single run of the camera, depicting an action over time. A keyframe is a single frame that best represents the visual content in a shot whereas a frame is a single still image.

The software is composed of three main modules. The first module concerns detecting shot boundaries and extract frames from the uploaded video. The second module concerns with computing indices of visual complexity from the extracted frames or entire video. Finally, the third module provides different visualization options. The three modules will be discussed in great detail below.

3 Module 1: Shot Boundary Detection and Frame Extraction

Upon starting the program, users are allowed to upload a video file from a local folder or provide a URL for a YouTube video. Then, users are directed to the frame extraction interface (see Figure 2) where they can detect shots and extract frames or keyframes from the uploaded video.

[1] The Tool is currently under active development and will be released at https://github.com/Eaalghamdi/auvana

[2] https://www.electronjs.org

[3] https://nklayman.github.io/vue-cli-plugin-electron-builder/

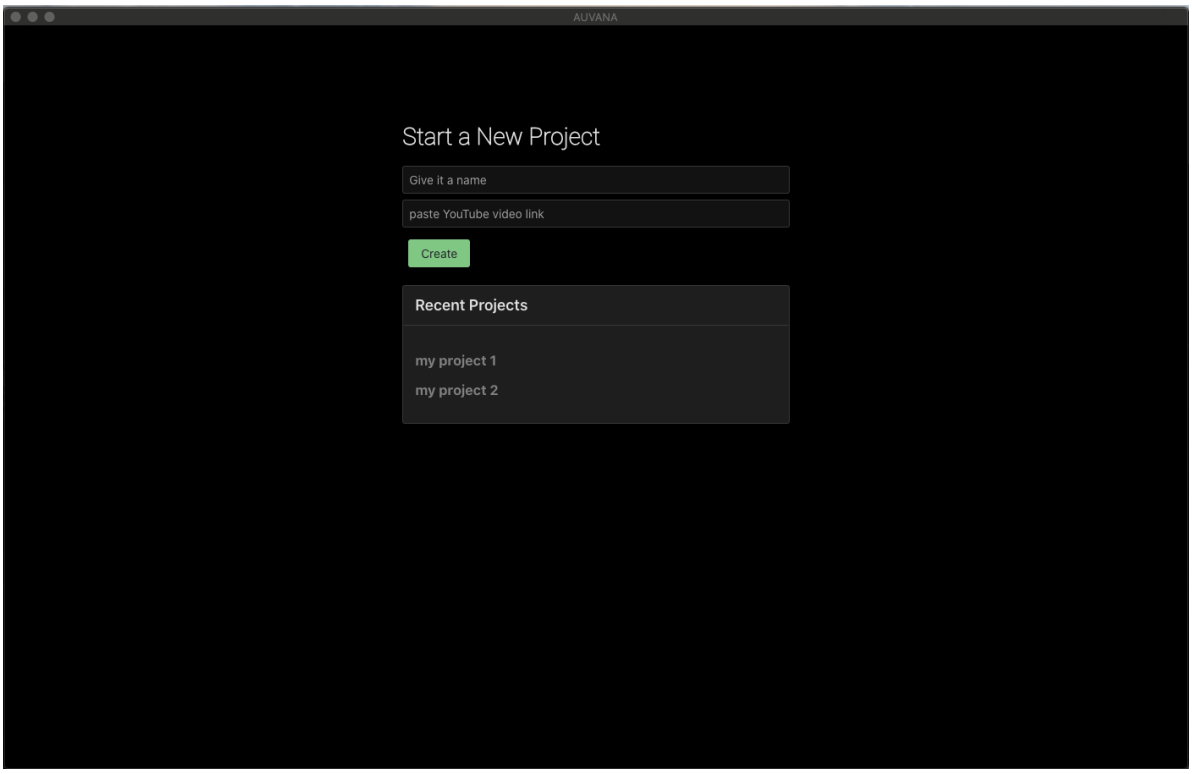

Figure 1: AUVANA video upload interface where users can upload a video from a local folder or provide an URL for YouTube video to be downloaded.

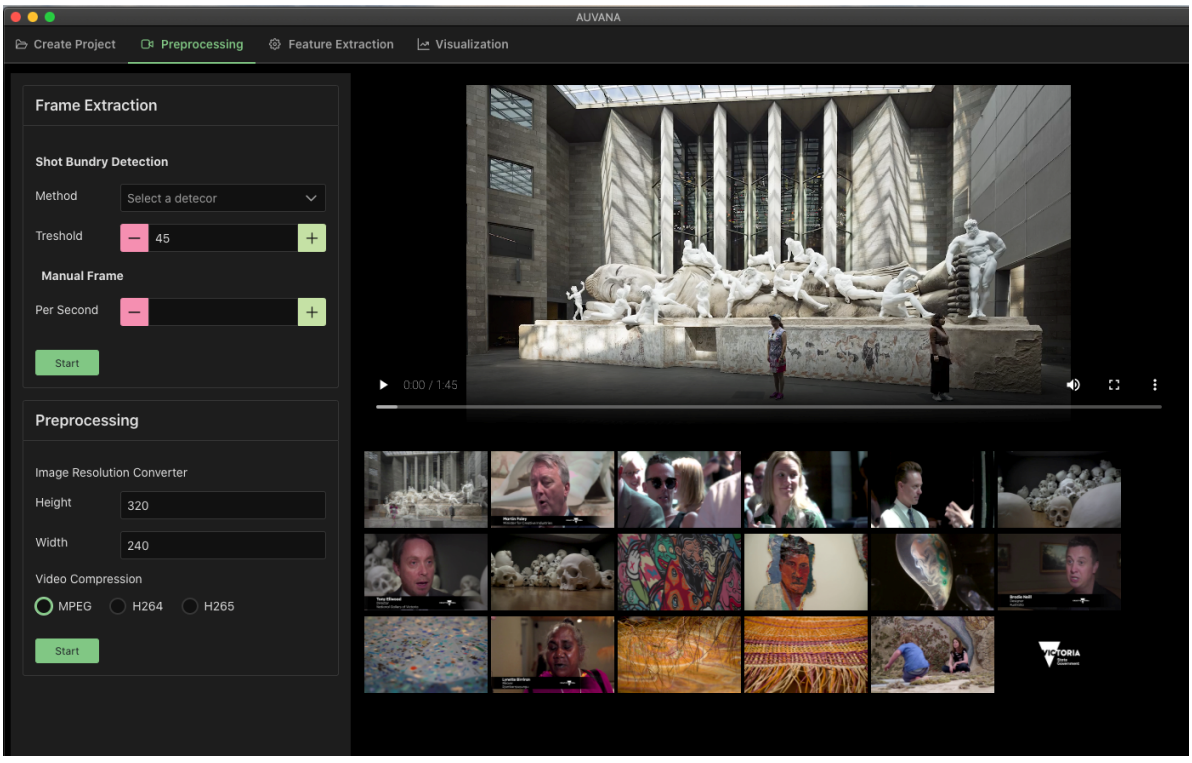

Figure 2: AUVANA preprocessing interface where users can detect shots and extract frames from the uploaded video.

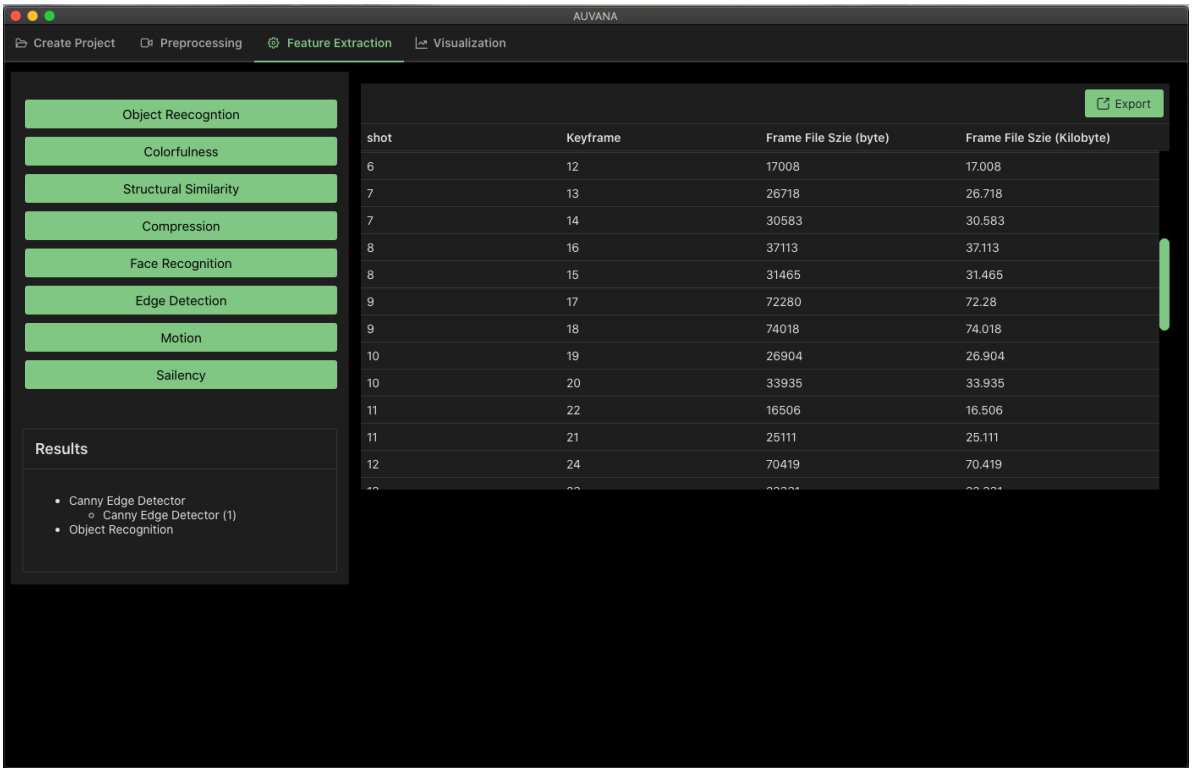

Figure 3: AUVANA feature interface where users can choose what indices to compute.

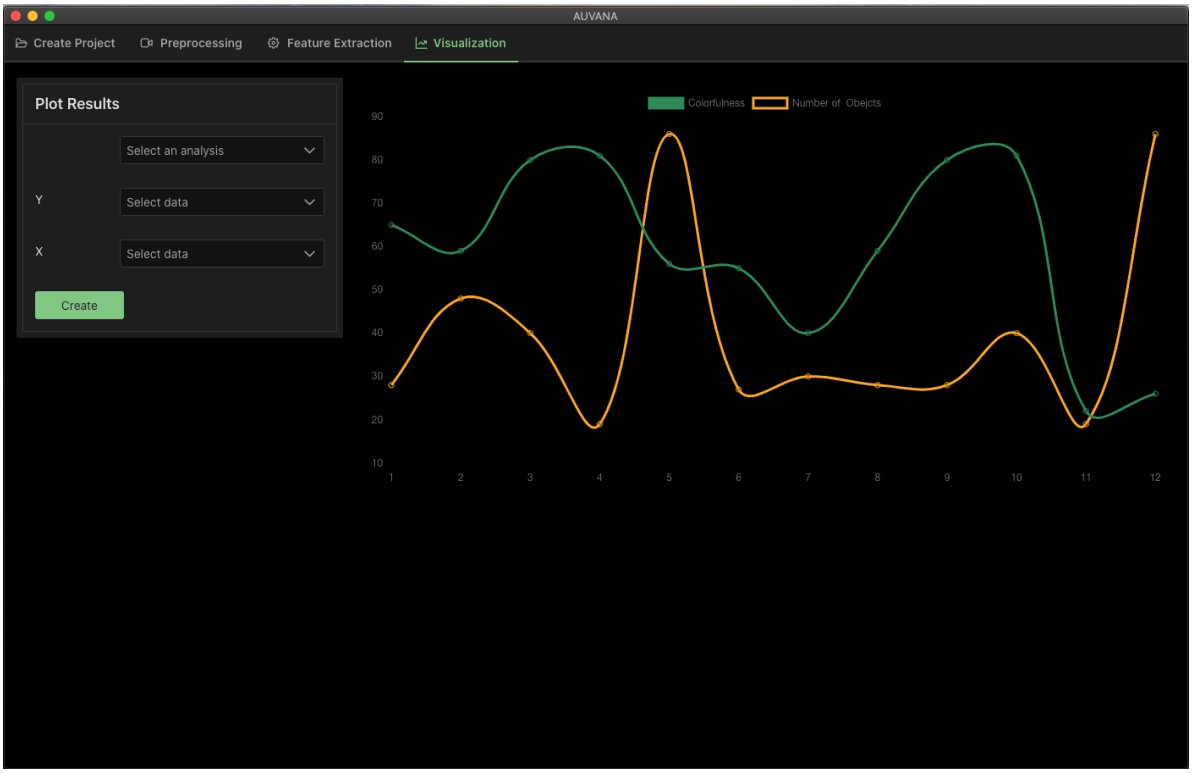

Figure 4: AUVANA visualization interface where users can diagnose and plot the computed indices.

For detecting shot boundaries in a video, AUVANA offers two shot boundary detection (SBD) algorithms, namely the threshold-based algorithm and the content-aware algorithm, both implemented in the PySceneDetect library [4]. The threshold detector compares the intensity or brightness of the pixel values in the current frame with a predefined threshold and triggers a shot boundary when this value crosses the threshold. The content-aware detector, on the other hand, relies on changes in HSV (hue, saturation, value) color space.

For each shot, AUVANA extracts a keyframe—a single frame that represents the shot. While a keyframe can be taken from anywhere in the shot, AUVANA selects the first stable frame from the onset of each shot. Users can also manually selects frames from videos.

The tool's users can change the resolutions of the extracted frames or keyframes before computing visual complexity indices. Because frame size can directly affect the calculation of several indices of visual complexity (e.g., image compression size), users can ensure that all extracted frames have the same dimensions, especially when comparing visual complexity across different videos.

4 Module 2: Extracting and Computing Indices

The current version of the AUVANA extracts and computes several visual complexity indices related to edges, saliency, human faces, objects, colorfulness, structural similarity, motion, and compression. For all indices, AUVANA computes basic statistics such as mean and standard deviation. Table 1 shows the different levels at which each visual complexity measure is extracted.

Table 1: Indices levels

Indices	Video	Shot	5SW	RS	Keyframe
Compression	✓			✓	✓
Colorfuness	✓		✓		✓
SSI	✓		✓		
Saliency	✓				✓
Faces	✓				
Movement	✓			✓	
Visual text	✓	✓			
Edges	✓			✓	✓
Objects					✓

4.1 Compression indices

In image processing literature, researchers have developed various objective metrics to quantify complexity in static pictures. A simple and commonly used quantitative metric for image complexity is the image file size after compression, typically using lossy algorithms such as JPEG and GIF. In essence, lossy algorithms compress data by removing redundant visual information in the picture and optimizing how information is stored and retrieved using more efficient mathematical operations. Therefore, given the same picture dimensions, more visually complex images have larger file sizes when compressed than less visually complex images. Despite being simple, compression-based metrics have been found to correlate with human evaluations of complexity in still images (Donderi & McFadden, 2005; Forsythe, Mulhern, & Sawey, 2008; Machado et al., 2015; Yu & Winkler, 2013).

Similar to static picture compression algorithms, lossy video compression algorithms reduce or compress the video data so it can be transmitted and shared over the Internet, though how video compression algorithms work is different from those of static images giving the temporal nature of video data. The additional dimension of time means that video compression algorithms should eliminate information redundancies in both spatial and temporal dimensions (Sayood, 2017). For example, in a one-minute shot of a person speaking in front of a static background, it would be counterproductive to encode the background image in every frame. A more efficient way is to encode the background frame once and refer back to it until the visual content in the background changes. Besides discarding spatial and temporal redundancies, video compression algorithms also rely upon how the human visual system works and capitalizes on its lack of sensitivity to colors (Cohen & Rubenstein, 2020; Lee & Kalva, 2008).

There are several types of video compression schemes or codecs, and while they employ different techniques, they all work toward the same end: reducing video file size with as little loss of quality as possible. Video codecs are also

[4] https://pyscenedetect.readthedocs.io/en/latest/

specifically designed to meet the practical considerations of the context of use, such as online streaming. Practically speaking, there is a trade-off between video quality and the time compression algorithm needs to encode videos, that is, a compression algorithm takes a long time to encode video data with higher quality. A thorough description of how video compression algorithms work is beyond the scope of this study but interested readers can refer to Akramullah (2014) and Lee and Kalva (2008).

Such quantitative measures of complexity, assuming they correlate with human perception of video difficulty, would be very useful, as they are relatively easy to obtain compared to subjective human ratings of videotext difficulty. To this end, we used two popular video compression codecs: Advanced Video Coding (AVC or H264) and HEVC (High-Efficiency Video Coding or H.265). These two video compression codecs employ different techniques to optimize for video quality and file size. For compressing videos, AUVANA uses the FFmpeg (version: 4.2.2) program (FFmpeg Developers, 2016).

4.2 Visual clutter indices

According to Rosenholtz, Li, and Nakano (2007), clutter "is the state in which excess items, or their representation or organization, lead to a degradation of performance at some task" (p. 3). Intuitively, the more cluttered an image is, the more difficult it becomes to search for and locate a particular item in it. Conversely, visually cluttered images are more likely to impede visual object processing and deplete working memory resources. Empirical studies have shown that cluttered images or displays increase memory load, delay visual search, negatively affect object recognition, reading, and linguistic processing (Coco & Keller, 2012). In the literature, there are several approaches and algorithmic methods to measure clutter in visual displays (see Moacdieh & Sarter, 2015, for comprehensive review)., most notably the *feature contestation* and *subband entropy* algorithms proposed in Rosenholtz, Li, Mansfield, and Jin (2005) and Rosenholtz et al. (2007).

The feature congestion algorithm computes how cluttered an image is based on color, luminance contrast, and orientation, whereas subband entropy quantifies the organization of objects in a scene using image encoding efficiency as a proxy. The two algorithms work best on still images. Therefore, we only assessed the visual clutter in the extracted keyframes from each video using a freely available MATLAB script provided by the original authors for both algorithms (Rosenholtz et al., 2007). After applying the two algorithms on each video keyframes, the values of each metric relating to each video are summed up to yield a single score. The mean and standard deviation of the clutter values generated by the two algorithms are also computed.

4.3 Edge indices

Object boundaries are important for human vision, for example, to identify and resolve ambiguity in recognizing objects in a (cluttered) visual scene (e.g., Palmer, 1999). According to Machado et al. "edges in an image usually indicate changes in depth, orientation, illumination, material, object boundaries" 2015, p. 45 Therefore, it is reasonable to assume that perceived visual complexity would relate to the frequency and distribution of edges in video frames. Measures based on edge detection techniques have been found to correlate with human perception of static image complexity (Machado et al., 2015), but little is known about their potential role in explaining visual complexity in videos.

Edge detection is a computer vision task that seeks to locate and identify sharp discontinuities in an image, which are typically due to abrupt changes in pixel color intensity (Bhardwaj & Mittal, 2012). These discontinuities in color intensity can help to identify boundaries of objects in a visual scene. As such, edge detection is very useful and often is the first step in many downstream signal processing and computer vision tasks, including object segmentation, recognition, image compression, and many others.

Although there are several edge detection algorithms (see, Maini & Aggarwal, 2009, for a review), the Canny edge detector (Canny, 1986) is a widely popular edge detection algorithm capable of detecting a wide range of edges in an image and it has been extensively used in visual complexity research (e.g., Gartus & Leder, 2017; Machado et al., 2015; Madan, Bayer, Gamer, Lonsdorf, & Sommer, 2018). It is a multi-stage algorithm that removes insignificant information from images and keeps only (see Figure 5).

We applied the Canny detector on both keyframes and videos. As there is no straightforward way to count edges, we used the file size to quantify edge information in videos and keyframes. To ensure comparability, two procedures were taken; firstly, the video and keyframe were scaled to the same resolution, and secondly, the compression method is kept the same. After applying the Canny edge detector on keyframes and videos, we computed basic statistics on the file size. Specifically, we summed keyframe file size in Kilobyte (KB) and calculated the mean, standard deviation, and range, whereas we used the file size in Megabyte (MB) to quantify edges in videos.

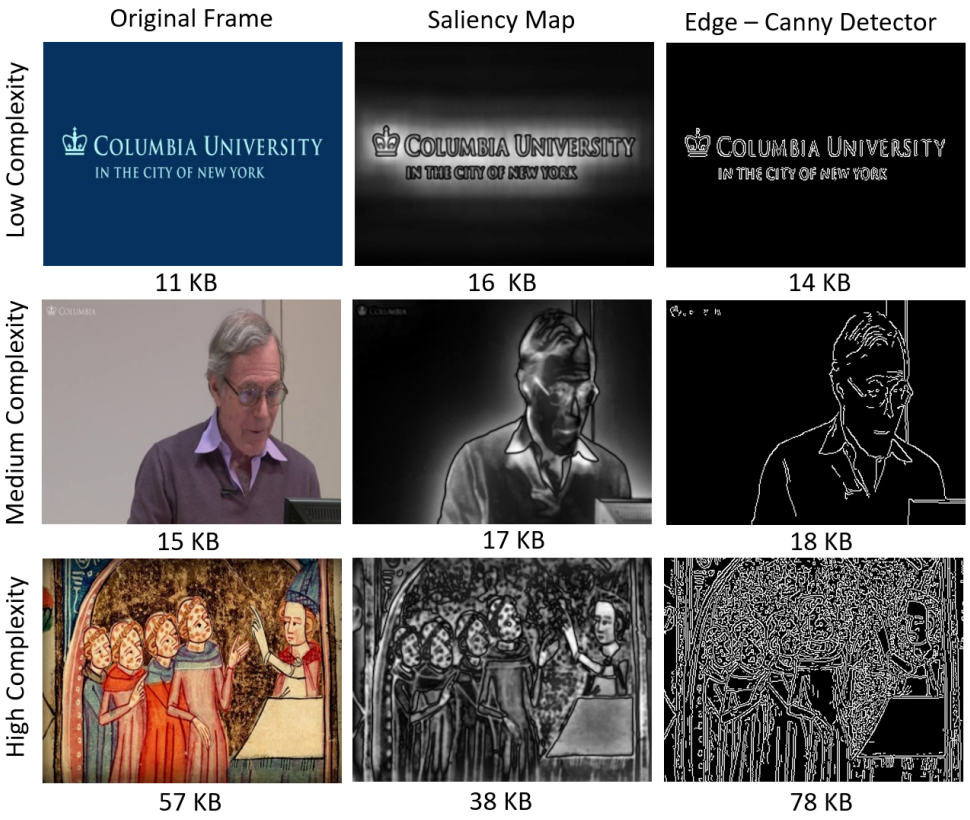

Figure 5: The Readership between File Sizes and the Saliency Maps and Edges of Frames of Varying Complexity

4.4 Saliency indices

When viewing an image without a task in mind, humans typically fixate their eyes on regions or objects that stand out amongst their surroundings. These objects or areas are often referred to as *salient* and, what makes them so has intrigued many researchers from different fields, including psychology, neuroscience, computer vision, and robotics. In the last two decades, a large effort has been put toward developing computational models that predict saliency in both static and dynamic images, some of these models were inspired by neural mechanisms (see Kavak, Erdem, & Erdem, 2017; Zhao & Koch, 2013, for a comprehensive review). There is, however, no universally agreed-upon definition of saliency, and existing computational models vary greatly in how they compute saliency. In their computation, saliency models generally make use of either bottom-up (low-level) image features, or top-down (high-level) features, or a combination of both. The majority of existing saliency models are bottom-up driven and mostly rely on a biologically plausible set of features that mimic early visual processing (e.g., Itti, Koch, & Niebur, 1998).

Itti et al. (1998) developed the first computational model of saliency in 1998 and since then researchers have developed numerous models, These models can be classified, according to their mechanisms for estimating visual attention, into several categories. We are interested here in those models that, from a still image or video, produce a visual saliency map. A saliency map is typically an aggregation of several maps that represent (predict) how salient each area of that image is to a human observer (Frintrop, Rome, & Christensen, 2010). While the main objective of saliency models is to output a single saliency map from an image or video stream, they differ as to what features to include in computing the saliency map (Koch & Ullman, 1985).

Computer vision researchers typically make a distinction between fixation prediction and object/region saliency detection (Borji, 2015; Borji & Itti, 2013). Fixation prediction models aim to predict points in an image where humans might look at in a free-viewing task. On the other hand, object saliency models aim to detect and segment salient objects, for example, by drawing a contour around the object. While the objective of the two types of models is different, the two types of models, in practice, often generate similar saliency maps (Borji, 2015).

Another distinction is whether the model is developed for predicting static or dynamic saliency. Dynamic saliency models should also account for the time dimension in the video input, for example, by tracking moving objects.

In this study, we consider both static and dynamic saliency as a potential predictor of video difficulty. Static saliency was computed based on keyframes extracted from the videos, whereas dynamic or motion saliency was computed using the entire video. Specifically, three different static saliency models were used for estimating saliency in keyframes. The first is a simple, yet influential saliency model developed by Itti et al. (1998). The other two are the Spectral Residual (SR) method (Hou & Zhang, 2007) and the Fine-Grained (FG) method (Montabone & Soto, 2010). For estimating video saliency, a motion saliency algorithm developed by B. Wang and Dudek (2014) was utilized. This motion saliency detector is based on the background subtraction technique and it considers moving objects as salient objects.

Because there is no direct way to quantify saliency, the summation of image and video size of the generated saliency maps for each algorithm was used as a proxy of saliency in keyframes and video, respectively. In the case of static saliency, the average, range, and standard deviation were also calculated. Figure 6 shows a comparison of saliency maps generated by the three algorithms and their corresponding file size in KB.

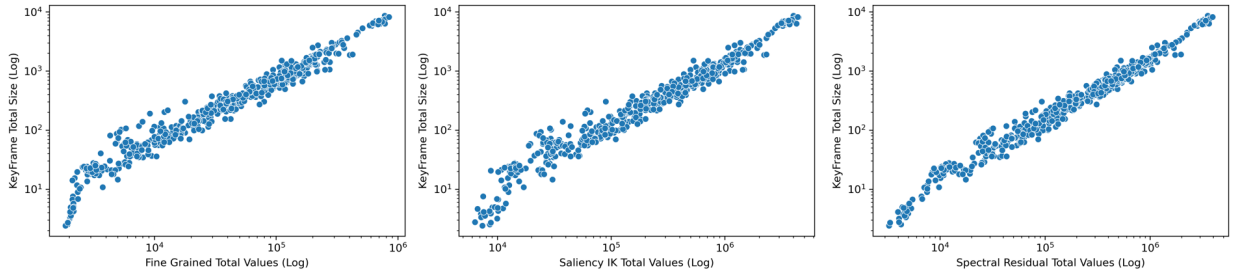

Figure 6: A comparison of saliency maps generated by the three algorithms and their corresponding file size in KB.

4.5 Colorfulness indices

There is mounting evidence suggesting that color grabs attention, invokes emotional reactions, and plays an important role in object recognition (Tanaka, Weiskopf, & Williams, 2001; Witzel & Gegenfurtner, 2018). A well-established measure of colorfulness in image quality literature is the one proposed by Hasler and Suesstrunk (2003). In their study, Hasler and Suesstrunk asked 20 non-expert participants to rate a set of 84 images using a 1-7 scale (ranging from Not Colorful to Extremely Colorful). Then, through running a series of experimental calculations on the collected data, they derived a simple metric that highly correlates with the participants' subjective rating. Specifically, the researchers found a simple opponent color space representation along with the mean and standard deviations of these values correlates to 95.3 % of the participants' ratings. Hasler and Suesstrunk's colorfulness metric has been widely adopted in several research domains and for different applications, for example, in assessing the complexity and anesthetic appeal of websites and visual displays (e.g., Reinecke et al., 2013), and measuring image and video quality (e.g., Winkler, 2012).

AUVANA computes colorfulness in frames, keyframes, as well as on the entire video (on frame every 1 second). The colorfulness algorithm was implemented using Python and the OpenCV library. To determine how colorful a frame is, the frame is split into three color channels; *Red, Green*, and *Blue*. Then, the following set of equations Equations (1) to (5) are applied to derive our metric C:

$$re = R - G \tag{1}$$

$$yb = \frac{1}{2}(R + G) - B \tag{2}$$

$$\sigma_r gyb = \sqrt{\sigma_r g^2 + \sigma_x b^2} \tag{3}$$

$$\mu_r gyb = \sqrt{\mu_r g^2 + \mu_y b^2} \tag{4}$$

$$C = \alpha_r gyb + 0.3 \times \mu_r gyb \tag{5}$$

In equation 1, rg is the difference between the Red channel and the Green channel. In equation 2, yb represents half of the sum of the Red and Green channels minus the Blue channel. Next, the standard deviation of $rgyb$ and mean of $rgyb$ are computed in equations, 3 and 4, receptively, before calculating the final colorfulness metric, C, in equation 5. To derive a global colorfulness metric, GC, for the entire video, the colorfulness values for all frames are simply summed up.

While GC estimates the degree of colorfulness in the entire video, it did not capture abrupt changes in color in two consequent frames. Abrupt changes in color values may impact visual perception. Therefore, AUVANA also computes the differences in colorfulness between two adjacent frames, ADJGC, using the equation in 6:

$$ADJGC = \sum_{i=1}^{n} (C_{t-1} - C_t)^2 \tag{6}$$

Additionally, for both GC and $ADJGC$, AUVANA calculates the average and standard deviation of C values derived from keyframes, frames, and the entire video.

4.6 Structural similarity indices

Structural Similarity (SSIM) index is an image quality assessment metric for evaluating the visual similarity between two images (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004). Technically, SSIM is used on two similar images—a reference image x and a test image y—to quantify their visual similarity. We used it here with two adjacent nonidentical frames as a way to assess structural changes between two frames. SSIM is computed in several steps, as shown in the Equations (7) to (10) below:

$$l(x, y) = (2\mu_x\mu_y + C1)/(\mu_x^2 + \mu_y^2 + C1) \tag{7}$$

$$c(x, y) = (2\sigma_x\sigma_y + C1)/(\sigma_x^2 + \sigma_y^2 + C2) \tag{8}$$

$$r(x, y) = (\sigma_x y + C3)/(\sigma_x\sigma_y + C3) \tag{9}$$

$$SSIM(x, y) = [l(x, y)]^{\alpha}.[c(x, y)]^{\beta}.[r(x, y)]^{\gamma} \tag{10}$$

where $C1$, $C2$, and $C3$ are constant terms added to avoid instabilities when $(\mu_x^2 + \mu_y^2)$, $(\sigma_x^2 + \sigma_y^2)$, or $\sigma_x\sigma_y$ is close to zero. The $l(x, y)$ index is related with luminance differences, $c(x, y)$ with contrast differences, and $r(x, y)$ with structure variations between x and y. The parameters α, β, and γ are added to the general form of SSIM equation in 10 to estimate the relative importance of each component.

We used the Scikit-image Python library's implementation of SSIM (version: N van der Walt et al., 2014). It should be noted that the SSIM algorithm takes in two consecutive frames which must have the same dimensions, and output a single value indicating to which degree the two frames are similar. A value of 0 indicates the two frames are completely different, whereas a value of 1 indicates the two frames are identical. Then, to derive a single global score for each video, all SSIM values are summed as shown in equation 11 :

$$Global\ SSIM = \sum_{i=1}^{n} SSIM_i \tag{11}$$

To compute the adjacent score ADJSSI, the equation in 12 was used:

$$ADJSSI = \sum_{i=1}^{n} (SSI_{t-1} - SSI_t)^2 \tag{12}$$

The SSI values can be computed on the extracted frames, keyframes, or the entire video.

Table 2: Visual Complexity Features, their Impact & Supporting Empirical Evidence

Feature	Hypothesized Impact	Empirical Evidence
Visual clutter	The more cluttered an image is, the harder it becomes to search and allocate an item within it	(Coco & Keller, 2012)
Visual saliency	Given that human attention is limited, only silent objects are processed while other objects are ignored	(Clarke, Coco, & Keller, 2013)
Color	Color guide visual attention and may help in object recognition, especially when objects are color diagnostic (e.g., strawberry is strongly associated with the color red).	(Bramão, Reis, Petersson, & Faísca, 2011; Tanaka et al., 2001)
Camera changes & cuts	A camera change signifies a change in information to the viewer, hence it elicits an automatic allocation of additional cognitive resources.	(Hendriks Vettehen & Kleemans, 2015; Lang & Geiger, 1996; Reeves et al., 1985)
Human Face	Human face automatically attracts more attention than any other objects in the visual scene. When there is no task being demonstrated, showing a human face can be helpful, but, if a task is being demonstrated, showing the model's face may lead to a division of attention.	(Hershler, Golan, Bentin, & Hochstein, 2010; Ouwehand, van Gog, & Paas, 2015)
Motion	Motions, especially abrupt and sudden motions, can readily attract attention and hence demanding additional attention resources.	(Mital, Smith, Hill, & Henderson, 2011)

4.7 Object recognition indices

The quantity of objects shown in video frames is more likely to affect the speed and efficacy of successfully processing and recognizing objects, hence making understanding the video more difficult. In the last few years, deep learning-based object recognition models have made great strides in terms of prediction accuracy. Object detection and extraction models take an image or video feed (frame-by-frame) as input and output a predicted label for each object along with the model's accuracy values (confidence) corresponding to each object classified by the model. AUVANA uses the ImageAI library's implantation of RetinaNet, a pre-trained deep learning model trained on the COCO dataset (Lin et al., 2014). The pre-trained model can detect 80 different everyday objects (e.g., car, chair, cup, and fork). The users can change confidence threshold of the model prediction accuracy to disregard predictions that do not meet the chosen threshold. Based on the output of the object detection, AUVANA calculates some basic indices such as the ratio of objects per shot, the sum, average, and standard deviation of all objects and unique objects per video.

4.8 Human face indices

Human faces are attention magnets. Both anecdotal and scientific evidence suggests that human faces attract more attention than any other object in the scene. When presented with other objects, our attention is automatically drawn towards human faces, and then to any other objects. More interestingly, we also pay more attention to human-like faces, such as faces of humanoid and animated pedagogical agents (Louwerse, Graesser, McNamara, & Lu, 2009). However, the question of how and under what conditions face guide attention is constantly debated in visual cognition literature. Several studies have demonstrated that the human face is rapidly and easily detected when presented among other objects (Hershler et al., 2010).

In the field of multimedia learning, it is generally assumed that showing the instructor face (and body) in the video provides useful nonverbal cues, e.g., mutual gaze, facial expression, and gestures, to learners which may aid them in processing the verbal information that is narrated by the instructor. As verbal and visual information is processed in two separate channels, as suggested by the Dual Coding Theory (Clark & Paivio, 1991), integrating both cues may result in an enhanced comprehension of the material. Further support for showing the instructor's face comes from the social presence theory which postulates that different media (e.g., face-to-face, telephone, text, and video) convey differing degrees of social cues that make interlocutors feel more or less psychologically present with one another. From this perspective, the presence of the instructor's face in video replicates the social aspects of human interaction and makes the learner feel as if they are interacting with another person (J. Wang & Antonenko, 2017).

AUVANA computes simple indices related to the appearance of human faces in videos, for example, the frequency of unique human faces that appeared in the video. It should be also noted that the recognized faces are most likely the faces of actual speakers in the video, but they can also include any human face displayed in the video. As

the appearance of the non-speaking faces may distract the viewer, we calculated the ratio of speaking faces to non-speaking faces. Further, the ratio of appearance/speaking time for the speaker to total running time. If there is more than one speaker, the durations of their appearance time are summed up and averaged to give one score. For detecting faces, we used OpenCV's pre-trained face recognition classifier based on the Haar cascades algorithm. The cascade classifier has been trained on numerous positive and negative images of face objects and it can subsequently be used to detect faces in other images or videos. One key advantage of this approach is that it is fast compared to deep learning-based classifiers.

4.9 Visual text indices

We use the term *visual text* in this study to refer to any letters and numbers displayed in the videos. A large number of visual texts can be distracting and may lead to information overload. Computationally, visual texts can be detected and recognized through the use of Optical Character Recognition (OCR) systems. Traditional OCRs are based on carefully crafted features, whereas in modern deep learning-based approaches, informative features are learned from a massive training dataset (e.g., Jaderberg, Vedaldi, & Zisserman, 2014; Zhou et al., 2017). Text detection in video streams, as opposed to text documents or still images, is a challenging task due to several factors, including motion, shadow, lighting, viewing angles, and complex backgrounds, all of which make detecting and extracting text from its background difficult (see Mancas & Gosseli, 2007, for a comprehensive review).

To detect visual texts, AUVANA uses the Efficient Accurate Scene Text detector (EAST: Zhou et al., 2017), using the OpenCV implementation of the detector. EAST is a specialized fully convolutional neural network that has been configured for detecting text from still images and videos. Based on the EAST outputs, basic statistics, e.g., the frequency, average, and standard deviation, of visual texts in the entire video as well as in each video shot were computed. Additionally, the ratio of visual texts per shot was considered to be a potential predictor of visual complexity.

4.10 Motion and object tracking indices

Moving objects attract attention in a visual scene (Wolfe & Horowitz, 2004, 2017). There is considerable evidence that a viewer's ability to track objects diminishes as the quantity of moving objects increases (Pylyshyn & Storm, 1988). Therefore, we hypothesized that the frequency of moving objects in the video stream as well as their longevity may distract visual attention, hence contributing to the perception of video complexity. There are several techniques to detect moving objects in a video stream, including optical flow-based techniques, background subtracting (KaewTraKulPong & Bowden, 2002; Zivkovic & Van Der Heijden, 2006), and frame differencing methods. By and large, motion detection techniques assume that the background of in video stream largely static and does not change over consecutive frames in videos. Therefore, to detect motion in a video stream, the background should firstly be differentiated from the foreground and then for review and comparison (Sehairi, Chouireb, & Meunier, 2017; Xu, Dong, Zhang, & Xu, 2016).

To track moving objects, AUVANA uses the frame differencing method because it is memory efficient. This technique detects motion, M, by calculating the absolute difference in pixel intensity between two adjacent frames. That is, in a binary or gray image, for each pixel with coordinates (x, y) in frame I, we computed the absolute difference with its corresponding coordinates in the next frame I_t using the following equation 13:

$$M(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \tag{13}$$

Using pixel coordinates, contours are drawn around each moving object in each frame and then tracked over the subsequent frames until the movement halts. The duration of each moving object is logged in milliseconds. This information allows us to compute some useful indicators of motion in videos, such as the frequency of moving objects and computed their mean and standard deviation in each video. Finally, to compute a global measure of motion GM, we use the following formula:

$$GM = \frac{\sum_{i=i}^{n} m_i}{d} \tag{14}$$

where m is local motion duration in seconds and d is the total duration of the video in seconds.

5 Module 3: Visualization

The third module gives the tool's users the ability to visualize the computed complexity indices using the Chart.js library [5]. Users have several options for plotting the results, such as scatter plots, line plots, and bar charts, and can also save the plots to a local folder.

6 Explanatory Analysis

In this section, we report the findings of an explanatory analysis on the extracted indices. Specifically, we investigated how the proposed visual complexity indices relate to each other and whether they represent underlying latent structures. To the end, the Second Language Video Complexity (SLVC) corpus (Alghamdi, 2021) was used. The corpus composes 640 videos in two video genres: academic lectures and government advertisements. Principle component analysis was conducted on the extracted indices. Before running PCA, indices that are basically derivative of other features were excluded and the remaining features were scaled because PCA is sensitive to the scaling of the original features. Then, the factorability of the visual complexity indices was examined. Firstly, it was observed that a large number of indices correlated at least .3 with at least one other item, suggesting reasonable factorability. Secondly, the sampling adequacy as determined by the Kaiser-Meyer-Olkin test was .83—which is above the commonly recommended value of .6—and Bartlett's test of sphericity was significant ($\chi_2(1891) = 65221.11, p < .01$). Then, different PCA solutions were explored. The most appropriate PCA solution had 14 components as indicated by a scree plot. The PCA analysis was followed by Direct Oblimin (a non-orthogonal) rotation because it was expected that some of the indices are moderately or highly correlated with each other [6]. A total of four indices were eliminated because they did not contribute to a simple factor structure and failed to meet minimum criteria of having a primary factor loading of .5 or above and no cross-loading of .3 or above. Table 4 shows the loadings of the factors and their commonalities.

This first component explained 22.52% of the variance and ixt comprised 12 visual complexity features. Specifically, indices in this component estimate the total values of visual clutter, saliency, objects, and compression. The second component explained about 12.17% of the variance in the dataset and it contains indices related to the spread of saliency, clutter, and structural similarity in videos. Five indices related to visual texts loaded heavily on the third component. The fourth component had six motion indices. Indices to the appearance of human faces in videos loaded on a single factor. The number of video shots had a higher loading on factor nine but also factor one.

The correlation matrix showed that the components did not have a higher correlation among each other (see Table 3)

Table 3: Component Correlation Matrix

Component	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2	0.15	–	–	–	–	–	–	–	–	–	–	–	–	–
3	-0.06	0.00	–	–	–	–	–	–	–	–	–	–	–	–
4	0.01	0.16	-0.02	–	–	–	–	–	–	–	–	–	–	–
5	-0.23	0.09	-0.15	-0.03	–	–	–	–	–	–	–	–	–	–
6	-0.20	-0.20	-0.08	-0.13	0.14	–	–	–	–	–	–	–	–	–
7	-0.12	-0.25	0.09	-0.16	-0.03	0.18	–	–	–	–	–	–	–	–
8	-0.16	0.11	-0.19	0.06	0.11	-0.07	-0.08	–	–	–	–	–	–	–
9	0.11	0.03	0.07	-0.11	-0.04	0.12	0.02	-0.15	–	–	–	–	–	–
10	-0.20	-0.18	-0.07	-0.08	0.25	0.11	0.10	-0.01	0.00	–	–	–	–	–
11	0.13	0.19	-0.06	0.13	-0.05	-0.07	-0.06	0.12	-0.12	-0.18	–	–	–	–
12	0.11	0.12	-0.04	0.13	0.02	-0.06	0.01	0.01	0.02	-0.14	0.04	–	–	–
13	-0.11	-0.09	0.04	0.05	0.02	0.01	0.09	0.03	-0.07	-0.06	0.04	0.03	–	–
14	-0.11	-0.09	0.00	0.01	-0.02	0.02	0.05	-0.07	-0.05	0.05	-0.07	-0.04	0.02	–

7 Discussion

In this paper, we introduced an open-sourced, cross-platform software for analyzing visual complexity in videos. The tool was primarily developed to help researchers who are interested in investigating visual complexity in videos but

[5] https://www.chartjs.org

[6] The fourteen oblimin solution was also compared with a varimax solution and no significant difference was found

lack programming skills or familiarity with computer vision and signal processing literature. The current version of the tool extracts and computes 76 video visual complexity indices that were motivated by research on visual attention and perception. The results of PCA analysis showed that the visual complexity indices are grouped into 14 different components or clusters, suggesting that the clustered indices assess different aspects of visual complexity. While it is tempting to name or label the extracted components, this step was skipped to avoid reification—assuming that the clusters of variables represent a latent variable while in reality, they do not. But most importantly, this finding suggests that video visual complexity is a multidimensional and multifaceted construct. Further research is of course needed for ascertaining if the patterns found in our dataset exist in other types of videotext.

Concerning the utilities of the tool, we believe our tool can be used in various research domains and applications. For example, researchers, in the fields of educational psychology and multimedia learning, can use the tool to assess visual complexity in educational videos and how it might impact learning from videos. Similarly, advertisement and marketing researchers can examine how visual complexity impacts customers' experiences.

In our future work, we plan further examine the visual complexity indices and run experimental studies to empirically validate the proposed indices. We will also continue adding more features and indices to the tool. While GUI applications are convenient and user-friendly, they are not readily accessible to developers who may want to contribute to the development of the tool. Therefore, we plan to develop an open-sourced Python package to allow developers and interested researchers to add more features to the package.

8 Conclusion

In this paper, we introduced an open-source tool for automatically extracting, computing, and visualizing a broadening array of visual complexity in videos. The indices were extracted using a variety of algorithms and techniques from fields of computer vision and signal processing. To encourage further research, we open-sourced AUVANA to the scientific community and we hope it will stimulate future research on understanding visual complexity in videos.

References

Akramullah, S. (2014). *Digital video concepts, methods, and metrics: quality, compression, performance, and power trade-off analysis*. Springer Nature.

Alghamdi, E. A. (2021). *Automated video difficulty assessment* (Unpublished doctoral dissertation).

Bhardwaj, S., & Mittal, A. (2012). A Survey on various edge detector techniques. *Procedia Technology*, *4*, 220–226. doi: 10.1016/j.protcy.2012.05.033

Borji, A. (2015). What is a salient object? A dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing*, *24*(2), 742–756. doi: 10.1109/tip.2014.2383320

Borji, A., & Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 185–207. doi: 10.1109/tpami.2012.89

Bramão, I., Reis, A., Petersson, K. M., & Faísca, L. (2011). The role of color information on object recognition: A review and meta-analysis. *Acta Psychologica*, *138*(1), 244–253. doi: 10.1016/j.actpsy.2011.06.010

Braun, J., Amirshahi, S. A., Denzler, J., & Redies, C. (2013). Statistical image properties of print advertisements, visual artworks and images of architecture. *Frontiers in Psychology*, *4*, 808.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Pami-8*(6), 679–698. doi: 10.1109/tpami.1986.4767851

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review*, *3*(3), 149–210. doi: 10.1007/bf01320076

Clarke, A. D., Coco, M. I., & Keller, F. (2013). The impact of attentional, linguistic, and visual features during object naming. *Frontiers in Psychology*, *4*(Dec), 1–12. doi: 10.3389/fpsyg.2013.00927

Coco, M. I., & Keller, F. (2012). Scan Patterns Predict Sentence Production in the Cross-Modal Processing of Visual Scenes. *Cognitive Science*, *36*(7), 1204–1223. doi: 10.1111/j.1551-6709.2012.01246.x

Cohen, M. A., & Rubenstein, J. (2020). How much color do we see in the blink of an eye? *Cognition*, *200*, 104268.

Da Silva, M. P., Courboulay, V., & Estraillier, P. (2011). Image complexity measure based on visual attention. In *2011 18th ieee international conference on image processing* (pp. 3281–3284).

Donderi, D. C., & McFadden, S. (2005). Compressed file length predicts search time and errors on visual displays. *Displays*, *26*(2), 71–78.

Ellis, C. T., & Turk-Browne, N. B. (2019). Complexity can facilitate visual and auditory perception. *Journal of experimental psychology: human perception and performance*, *45*(9), 1271.

FFmpeg Developers. (2016). *ffmpeg*.

Forsythe, A., Mulhern, G., & Sawey, M. (2008). Confounds in pictorial sets: the role of complexity and familiarity in basic-level picture processing. *Behavior research methods*, *40*(1), 116–129. doi: 10.3758/brm.40.1.116

Frintrop, S., Rome, E., & Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations. *ACM Transactions on Applied Perception*, *7*(1), 1–39. doi: 10.1145/1658349.1658355

Gartus, A., & Leder, H. (2017). Predicting perceived visual complexity of abstract patterns using computational measures: The influence of mirror symmetry on complexity perception. *PLoS ONE*, *12*(11), 1–30. doi: 10.1371/journal.pone.0185276

Hasler, D., & Suesstrunk, S. E. (2003). Measuring colorfulness in natural images. In B. E. Rogowitz & T. N. Pappas (Eds.), (p. 87). doi: 10.1117/12.477378

Heaps, C., & Handel, S. (1999). Similarity and features of natural textures. *Journal of Experimental Psychology: Human Perception and Performance*, *25*(2), 299–320. doi: 10.1037/0096-1523.25.2.299

Hendriks Vettehen, P., & Kleemans, M. (2015). How Camera Changes and Information Introduced Affect the Recognition of Public Service Announcements: A Test Outside the Lab. *Communication Research*, *46*(7), 908–925. doi: 10.1177/0093650215616458

Hershler, O., Golan, T., Bentin, S., & Hochstein, S. (2010). The wide window of face detection Orit Hershler Shlomo Bentin. *Journal of Vision*, *10*(2010), 1–14. doi: 10.1167/10.10.21.Introduction

Hou, X., & Zhang, L. (2007). Saliency Detection: A Spectral Residual Approach. In *2007 ieee conference on computer vision and pattern recognition* (pp. 1–8). Ieee. doi: 10.1109/cvpr.2007.383267

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259. doi: 10.1109/34.730558

Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014). Deep features for text spotting. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8692 Lncs*(Part 4), 512–528. doi: 10.1007/978-3-319-10593-2_34

KaewTraKulPong, P., & Bowden, R. (2002). An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. *Video-Based Surveillance Systems*, 135–144. doi: 10.1007/978-1-4615-0913-4_11

Kavak, Y., Erdem, E., & Erdem, A. (2017). A comparative study for feature integration strategies in dynamic saliency estimation. *Signal Processing: Image Communication*, *51*(November 2016), 13–25. doi: 10.1016/j.image.2016.11.003

Koch, C., & Ullman, S. (1985). Shifts in selective visual attention: towards the. *Human neurobiology*, *4*, 219–227.

Lang, A., & Geiger, S. (1996). *The effects of related and unrelated cuts on television* (Vol. 20) (No. 1). doi: 10.1177/009365093020001001

Lee, J.-B., & Kalva, H. (2008). *The vc-1 and h. 264 video compression standards for broadband video services* (Vol. 32). Springer Science & Business Media.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8693 Lncs*(Part 5), 740–755. doi: 10.1007/978-3-319-10602-1_48

Louwerse, M. M., Graesser, A. C., McNamara, D. S., & Lu, S. (2009). Embodied conversational agents as conversational partners. *Applied Cognitive Psychology*, *23*(9), 1244–1255. doi: 10.1002/acp.1527

Machado, P., Romero, J., Nadal, M., Santos, A., Correia, J., & Carballal, A. (2015). Computerized measures of visual complexity. *Acta Psychologica*, *160*, 43–57. doi: 10.1016/j.actpsy.2015.06.005

Madan, C. R., Bayer, J., Gamer, M., Lonsdorf, T. B., & Sommer, T. (2018). Visual Complexity and Affect: Ratings Reflect More Than Meets the Eye. *Frontiers in Psychology*, *8*(January), 1–19. doi: 10.3389/fpsyg.2017.02368

Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, *3*(1), 1–11.

Mancas, C., & Gosseli, B. (2007). Natural scene text understanding. In *Vision systems: Segmentation and pattern recognition*. IntechOpen. doi: 10.5772/4966

Mital, P. K., Smith, T. J., Hill, R. L., & Henderson, J. M. (2011). Clustering of gaze during dynamic scene viewing is predicted by motion. *Cognitive computation*, *3*(1), 5–24.

Moacdieh, N., & Sarter, N. (2015). Display clutter: A review of definitions and measurement techniques. *Human Factors*, *57*(1), 61–100. doi: 10.1177/0018720814541145

Montabone, S., & Soto, A. (2010). Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing*, *28*(3), 391–402. doi: 10.1016/j.imavis.2009.06.006

Ouwehand, K., van Gog, T., & Paas, F. (2015). Designing effective video-based modeling examples using gaze and gesture cues. *Educational Technology and Society*, *18*(4), 78–88.

Palmer, S. E. (1999). *Vision science: Photons to phenomenology*. MIT press.

Palumbo, L., Makin, A. D. J., & Bertamini, M. (2014). Examining visual complexity and its influence on perceived duration. *Journal of Vision*, *14*(14), 1–18. doi: 10.1167/14.14.3.doi

Pieters, R., Wedel, M., & Batra, R. (2010). The Stopping Power of Advertising: Measures and Effects of Visual Complexity. *Journal of Marketing*, *74*(5), 48–60. doi: 10.1509/jmkg.74.5.48

Pylyshyn, Z. W., & Storm, R. W. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial vision*, *3*(3), 179–197.

Reeves, B., Thorson, E., Rothschild, M. L., McDonald, D., Hirsch, J., & Goldstein, R. (1985). Attention to television: Intrastimulus effects of movement and scene changes on alpha variation over time. *International Journal of Neuroscience*, *27*(3-4), 241–255.

Reinecke, K., Yeh, T., Miratrix, L., Mardiko, R., Zhao, Y., Liu, J., & Gajos, K. Z. (2013). Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*, 2049. doi: 10.1145/2470654.2481281

Rosenholtz, R., Li, Y., Mansfield, J., & Jin, Z. (2005). Feature congestion: A Measure of Display Clutter. In *Proceedings of the sigchi conference on human factors in computing systems - chi '05* (p. 761). doi: 10.1145/1054972.1055078

Rosenholtz, R., Li, Y., & Nakano, L. (2007). Measuring visual clutter. *Journal of vision*, *7*(2), 17.1–22. doi: 10.1167/7.2.17

Salomon, D. (2004). *Data compression: the complete reference*. Springer Science & Business Media.

Sayood, K. (2017). *Introduction to data compression*. Morgan Kaufmann.

Sehairi, K., Chouireb, F., & Meunier, J. (2017). Comparative study of motion detection methods for video surveillance systems. *Journal of Electronic Imaging*, *26*(2), 023025. doi: 10.1117/1.jei.26.2.023025

Snodgrass, J. G., & Vanderwart, M. (1980). A standardised set of 260 pictures: normal for name agreement, familiarity and visual complexity. *Journal of Experimental Psychology: General*, *6*(2), 174–215.

Tanaka, J., Weiskopf, D., & Williams, P. (2001). The role of color in high-level vision. *Trends in Cognitive Sciences*, *5*(5), 211–215. doi: 10.1016/s1364-6613(00)01626-0

Tuch, A. N., Bargas-Avila, J. A., Opwis, K., & Wilhelm, F. H. (2009). Visual complexity of websites: Effects on users' experience, physiology, performance, and memory. *International journal of human-computer studies*, *67*(9), 703–715.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., … Yu, T. (2014). Scikit-image: image processing in Python. *PeerJ*, *2*, e453. doi: 10.7717/peerj.453

Wang, B., & Dudek, P. (2014). A fast self-tuning background subtraction algorithm. In *Ieee computer society conference on computer vision and pattern recognition workshops* (pp. 401–404). doi: 10.1109/cvprw.2014.64

Wang, J., & Antonenko, P. D. (2017). Instructor presence in instructional video: Effects on visual attention, recall, and perceived learning. *Computers in Human Behavior*, *71*, 79–89. doi: 10.1016/j.chb.2017.01.049

Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. doi: 10.1109/tip.2003.819861

Winkler, S. (2012). Analysis of public image and video databases for quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, *6*(6), 616–625.

Witzel, C., & Gegenfurtner, K. R. (2018). Color Perception: Objects, Constancy, and Categories. *Annual Review of Vision Science*, *4*(1), 475–499. doi: 10.1146/annurev-vision-091517-034231

Wolfe, J. M., & Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, *5*(6), 495–501. doi: 10.1038/nrn1411

Wolfe, J. M., & Horowitz, T. S. (2017). Five factors that guide attention in visual search. *Nature Human Behaviour*, *1*(3), 1–8. doi: 10.1038/s41562-017-0058

Xu, Y., Dong, J., Zhang, B., & Xu, D. (2016). Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Transactions on Intelligence Technology*, *1*(1), 43–60. doi: 10.1016/j.trit.2016.03.005

Yu, H., & Winkler, S. (2013). Image complexity and spatial information. *2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings*, 12–17. doi: 10.1109/QoMEX.2013.6603194

Zhao, Q., & Koch, C. (2013). Learning saliency-based visual attention: A review. *Signal Processing*, *93*(6), 1401–1407. doi: 10.1016/j.sigpro.2012.06.014

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2642–2651. doi: 10.1109/cvpr.2017.283

Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, *27*(7), 773–780. doi: 10.1016/j.patrec.2005.11.005

Table 4: Components and their Loading

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Comm.
SaliencySRKFSum	0.96	–	–	–	–	–	–	–	–	–	–	–	–	–	0.97
SaliencyFGKFSum	0.95	–	–	–	–	–	–	–	–	–	–	–	–	–	0.96
CannyKeyFrameKBSum	0.94	–	–	–	–	–	–	–	–	–	–	–	–	–	0.96
NumberofUniqueObjectsKF	0.91	–	–	–	–	–	–	–	–	–	–	–	–	–	0.85
NumberofObjectsKF	0.84	–	–	–	–	–	–	–	–	–	–	–	–	–	0.84
ClutterSESum	0.83	–	–	–	–	–	–	–	–	–	–	–	–	–	0.93
SaliencyIKsum	0.82	–	–	–	–	–	–	–	–	–	–	–	–	–	0.93
ClutterFCSum	0.80	–	–	–	–	–	–	–	–	–	–	–	–	–	0.91
VideoCompressionH264SizeMB	0.79	–	–	–	–	–	–	–	–	–	–	–	–	–	0.94
VideoCompressionH265SizeMB	0.73	–	–	–	–	–	–	–	–	–	–	–	–	–	0.75
NumberofMovingObjectsperVideo	0.66	–	–	–	–	–	–	–	–	–	–	–	–	–	0.74
CannyVideoSizeMB	0.63	–	–	–	–	–	–	–	0.50	–	–	–	–	–	0.90
ClutterSESD	–	0.82	–	–	–	–	–	–	–	–	–	–	–	–	0.81
SSIVideoW5SecSD	–	0.80	–	–	–	–	–	–	–	–	–	–	–	–	0.71
SaliencyIKSD	–	0.76	–	–	–	–	–	–	–	–	–	–	–	–	0.77
ClutterFCSD	–	0.66	–	–	–	–	–	–	–	–	–	–	–	–	0.77
SSIVideoW5Secmean	–	-0.55	–	–	–	–	–	–	–	–	–	–	–	–	0.78
FrequencyofVisualTextperBlocks	–	–	0.98	–	–	–	–	–	–	–	–	–	–	–	0.96
NumberofVisualTextperMinute	–	–	0.98	–	–	–	–	–	–	–	–	–	–	–	0.95
NumberofVisualTextperSecond	–	–	0.98	–	–	–	–	–	–	–	–	–	–	–	0.95
NumberofVisualText	–	–	0.85	–	–	–	–	–	–	–	–	–	–	–	0.88
NumberofVisualTextperShot	–	–	0.76	–	–	–	–	–	–	–	–	–	–	–	0.76
MotionDurtionDifferenceVaraincenormalized	–	–	–	0.93	–	–	–	–	–	–	–	–	–	–	0.84
MotionDurtionDifferenceVaraince	–	–	–	0.91	–	–	–	–	–	–	–	–	–	–	0.85
MotionDurtionDifferenceSDNormalized	–	–	–	0.86	–	–	–	–	–	–	–	–	–	–	0.90
MotionDurtionDifferenceMeanNormalized	–	–	–	0.81	–	–	–	–	–	–	–	–	–	–	0.87
MotionDurtionDifferenceSD	–	–	–	0.79	–	–	–	–	–	–	–	–	–	–	0.87
MotionDurtionDifferenceMean	–	–	–	0.71	–	–	–	–	–	–	–	–	–	–	0.82
KeyFrameKBmean	–	–	–	–	-0.88	–	–	–	–	–	–	–	–	–	0.93
SaliencySRKFMean	–	–	–	–	-0.85	–	–	–	–	–	–	–	–	–	0.86
CannyKeyFrameKBMean	–	–	–	–	-0.84	–	–	–	–	–	–	–	–	–	0.90
SaliencyFGKFMean	–	–	–	–	-0.70	–	–	–	–	–	–	–	–	–	0.79
ColorfulnessKFmean	–	–	–	–	–	-0.86	–	–	–	–	–	–	–	–	0.86
ColorfulnessVideoW5Secmean	–	–	–	–	–	-0.82	–	–	–	–	–	–	–	–	0.77
ColorfulnessKFSD	–	–	–	–	–	-0.82	–	–	–	–	–	–	–	–	0.87
ColorfulnessAdjacentKFmean	–	–	–	–	–	-0.80	–	–	–	–	–	–	–	–	0.80
ColorfulnessVideoW5SecSD	–	–	–	–	–	-0.72	–	–	–	–	–	–	–	–	0.88
ColorfulnessAdjacentVideoW5SecSD	–	–	–	–	–	-0.72	–	–	–	–	–	–	–	–	0.86
ColorfulnessAdjacentKFSD	–	–	–	–	–	-0.65	–	–	–	–	–	–	–	–	0.78

Table 4 continued from previous page

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Comm.
KeyFrameKBSD	–	–	–	–	–	–	-0.93	–	–	–	–	–	–	–	0.90
SaliencySRKFSD	–	–	–	–	–	–	-0.85	–	–	–	–	–	–	–	0.81
CannyKeyFrameKBSD	–	–	–	–	–	–	-0.82	–	–	–	–	–	–	–	0.86
SaliencyFGKFSD	–	–	–	–	–	–	-0.78	–	–	–	–	–	–	–	0.74
ColorfulnessAdjacentVideoW1SecSD	–	–	–	–	–	–	–	0.96	–	–	–	–	–	–	0.89
ColorfulnessVideoW1SecSD	–	–	–	–	–	–	–	0.91	–	–	–	–	–	–	0.85
SSIVideoW1SecSD	–	–	–	–	–	–	–	0.86	–	–	–	–	–	–	0.79
SSIVideoW1Secmean	–	–	–	–	–	–	–	-0.81	–	–	–	–	–	–	0.77
ColorfulnessVideoW1Secmean	–	–	–	–	–	–	–	0.64	–	–	–	–	–	–	0.62
freq_shots	–	–	–	–	–	–	–	–	0.58	–	–	–	–	–	0.89
SaliencyIKmean	–	–	–	–	–	–	–	–	–	-0.84	–	–	–	–	0.75
ClutterSEMean	–	–	–	–	–	–	–	–	–	-0.74	–	–	–	–	0.82
NumberofFacesperShot	–	–	–	–	–	–	–	–	–	–	0.81	–	–	–	0.71
NumberofFacesperMinute	–	–	–	–	–	–	–	–	–	–	0.66	–	–	–	0.79
NumberofFaces	–	–	–	–	–	–	–	–	–	–	0.65	–	–	–	0.75
TotalMotionDurationms	–	–	–	–	–	–	–	–	–	–	–	0.89	–	–	0.76
SSIKFmean	–	–	–	–	–	–	–	–	–	–	–	–	-0.63	–	0.78
ColorfulnessAdjacentVideoW1Secmean	–	–	–	–	–	–	–	–	–	–	–	–	–	0.90	0.80