# Closing-the-Loop: A Data-Driven Framework for Effective Video Summarization

Ran Xu
Purdue University
Email: xu943@purdue.edu

Haoliang Wang
Stefano Petrangeli
Viswanathan Swaminathan
Adobe Research
Email: {hawang, petrange, vishy}@adobe.com

Saurabh Bagchi
Purdue University
Email: sbagchi@purdue.edu

***Abstract*—Today, videos are the primary way in which information is shared over the Internet. Given the huge popularity of video sharing platforms, it is imperative to make videos engaging for the end-users. Content creators rely on their own experience to create engaging short videos starting from the raw content. Several approaches have been proposed in the past to assist creators in the summarization process. However, it is hard to quantify the effect of these edits on the end-user engagement. Moreover, the availability of video consumption data has opened the possibility to predict the effectiveness of a video before it is published. In this paper, we propose a novel framework to close the feedback loop between automatic video summarization and its data-driven evaluation. Our *Closing-The-Loop* framework is composed of two main steps that are repeated iteratively. Given an input video, we first generate a set of initial video summaries. Second, we predict the effectiveness of the generated variants based on a data-driven model trained on users' video consumption data. We employ a genetic algorithm to search the space of possible summaries (i.e., adding/removing shots to the video) in an efficient way, where only those variants with the highest predicted performance are allowed to survive and generate new variants in their place. Our results show that the proposed framework can improve the effectiveness of the generated summaries with minimal computation overhead compared to a baseline solution – 28.3% more video summaries are in the highest effectiveness class than those in the baseline.**

***Index Terms*—video summarization; effectiveness evaluation; genetic algorithm; deep learning;**

## I. Introduction

Video content is ubiquitous nowadays. Given the sheer amount of content shared on video platforms every day, it becomes increasingly important to create short and effective videos that can provide high level of engagement with their consumers. Quantifying the effectiveness of a video is a non-trivial task for content creators, as effectiveness depends on not only the content itself but also the target audience and publishing channels. Content creators usually rely on their experience and preference to create a short summary from long raw footage, which is not guaranteed to produce the best possible result. Machine Learning (ML)-assisted tools are used more and more to assist creators in this process, as they can greatly accelerate and improve the video summarization task. However, many of these techniques only focus on video-level characteristics (e.g., aesthetics) to generate the video summary [1], [2], without explicitly reasoning on the effectiveness of the generated output from an end-user's perspective. As an example, Gu *et al.* propose a GAN-based approach for video summarization that aims to minimize the difference in feature space between the original and summarized video [3]. While these approaches can generate visually appealing results, there is no guarantee that the final result is the most effective. Moreover, we now have access to a large amount of video content consumption data. All these rich, contextual data can be used to predict how effective a particular video will be, even before it is published [4]. For example, Lou *et al.* [5] propose an LSTM-based network to predict the *watchability* of a video based on audio-visual features. The proposed method is trained using historical data about the effectiveness of other videos. These insights can be potentially used to further optimize the video summarization process. However, being able to predict the content effectiveness alone is not enough for content creators, as it remains unclear what edits should be performed on the video to improve its effectiveness.

In this paper, we therefore propose to close the feedback loop in the video summarization process, by bridging the gap between automatic video summarization and its data-driven effectiveness prediction. Particularly, our *Closing-the-Loop* (CTL) framework iteratively searches the best video summary variant maximizing a data-driven metric, which is used to evaluate the effectiveness of the video. We formulate the problem of finding the near-optimal variant as an incremental genetic search problem. A *Creation App* (CA) is responsible to generate possible summaries, based on the input content and editing parameters. An *Evaluation App* (EA) evaluates these variants and predicts their effectiveness. A genetic algorithm intelligently improves the video summary generation, iteration after iteration, by selecting only a subset of the variants with the highest predicted performance. The selected variants are then used as new inputs for the CA. Ultimately, this iterative process produces the video summary with the highest predicted effectiveness by the EA. The main contributions of this paper are therefore two-fold:

- We design *Closing-the-Loop*, a data-driven video summarization framework that automatically summarizes an input video to maximize its predicted effectiveness, using a combination of a CA, to generate possible variants, and an EA, to evaluate these variants;

- We leverage a genetic algorithm to search the large and complex space of possible summaries in an efficient and scalable way and focus the effort on the most promising ones, with minimal computing overhead. Different from hard-to-interpret deep learning models, our approach provides an interpretable and incremental editing path leading to the final summary with highest effectiveness.

We evaluate the proposed CTL framework on the video summarization task, using the data-driven effectiveness score proposed by Lou *et al.* [5] as the feedback metric. Compared to a baseline ML solution that only consider video-level characteristics to generate a summary [3], we show how the proposed approach can generate new video summaries with the highest possible effectiveness score for 28.3% more videos in the analyzed dataset [4] compared to the baseline. Our proposed framework only adds marginal execution time overhead compared to the baseline.

## II. Related Work

Several ML-based works have been proposed in the past to automate and streamline the video summarization process, and to predict the effectiveness of a video before it is published.

In terms of video summarization, Gao *et al.* [6] use a combination of color, motion, and audio features to select the most important frames of the video. The advent of deep neural networks (DNNs) have brought consistent advancements to this task. Ranking models for video segments are popular solutions for video highlight detection. Specific models include EM-like self-paced model selection procedures [7] and deep learning techniques [8]. Gu *et al.* [3] and Mahasseni *et al.* [9] use a generative model where the summarizer network aims to generate summaries that the discriminator network cannot distinguish from the input. Even though these approaches can generate visually appealing results, they only consider video-specific objectives when creating a summary. In other words, these works can be categorized as *open-loop*, as content effectiveness is not explicitly taken into account.

In terms of video performance prediction, several works have investigated how to predict the effectiveness of a video for a particular user segment or publishing platform. This prediction is particularly important for creators, as it indicates how much impact the created content will have on the target audience. To achieve this goal, Lou *et al.* [5] use visual and metadata information associated with a video to predict its effectiveness using a mixture of LSTM network and logistic regression model. Hussain *et al.* [4] collect several datasets to analyze and evaluate the effectiveness of image and video advertisements. The datasets contain information on the topic and sentiment of the ad, what actions are performed etc. Even though using the predicted performance to drive the content creation is possible in theory, it is hard to apply this concept in practice since very often these models lack interpretability (especially for deep-learning-based solutions). Although a few approaches have been proposed to solve this issue [10], it remains challenging to fully interpret the decision taken by the effectiveness prediction model.

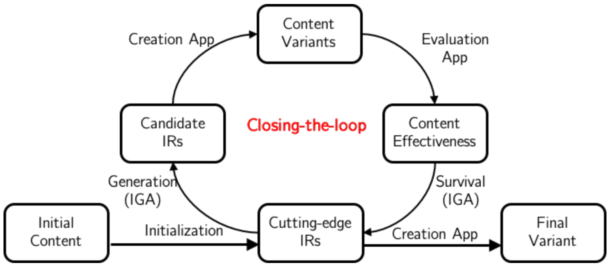


Fig. 1. The workflow of the Closing-the-Loop framework.

As seen above, the problem of closing the feedback loop between video summarization and its performance evaluation/prediction remains unaddressed. Previous works mostly focus on: 1) predicting the effectiveness of existing videos or 2) automatically generating summaries without directly considering how effective the created content would be. This paper closes this gap by proposing an iterative search based on Genetic Algorithms (GA) [11], [12]. More generally, such evolutionary approach are used for automatic video editing [13], video production [14], and video analytic systems [15].

## III. The Closing-the-Loop Framework

Our proposed CTL framework automates the process of finding the best video summary that maximizes the predicted content effectiveness and engagement for the end-users. Particularly, we use a CA to generate different video summary variants, and an EA to assess the performance of the generated variants. A GA allows to connect these two steps and efficiently search the best video summary variant. This design choice allows to plug any CA and EA in the proposed framework to optimize the video summarization process, according to the specific requirements of the content creator. Fig. 1 shows an overview of the process, which can be described as follows:

1) Given an input, CA generates a set of initial variants;
2) Generate a set of candidates from the initial variants, based on the generation policy of the GA and the CA;
3) Use the EA to predict the effectiveness of each candidate variant, in the form of a numerical score;
4) Based on the survival policy of the GA, select a subset of the variants that are going to be carried over to the next generation, also called the *cutting-edge video variants*;
5) Loop between 2-4 until the termination condition is met.

The final result of this iterative search process is a video summary that maximizes the effectiveness score as indicated by the EA. In the reminder of this section, we will present each step of the CTL framework in detail. Without loss of generality, we assume the input video is divided into shots, a set of consecutive frames belonging to the same scene. Particularly, we denote with $L$ the number of shots and with $x_i$ the $i^{th}$ shot of the video. Thus, the video summarization algorithm can be simplified as an $L$ binary selection problem.

### A. Intermediate Representation

To simplify the search process, we design a compact representation of the variants, which we call *Intermediate*

*Representation* (IR). An IR, $R$, is an $L$-long binary array, where $r_i = 1$ means the shot, $x_i$, is selected for the summary.

### *B. Creation App and Search Initialization*

In our data-driven framework, the CA is a generic open-loop algorithm that, given an input video, generates a summary with a user-specified duration. Our CTL framework can support any kind of summarization algorithm that falls in this category. As it will be detailed in Section IV, we choose a GAN-based approach [3], which minimizes the difference between the visual features of the input video and those of the generated summary. This solution is open-loop because it does not consider any video consumption data to generate the summary.

The CA is in charge of generating the first video summary, before the genetic algorithm starts searching for the best variant. The initial IR is the CA's choice of video shots to include in the summary. This initialization is an important part of our framework. A naive approach would be to generate a random initial summary, which will likely be associated with a low effectiveness score. Instead, we decide to use the CA for initialization. Intuitively, even though the CA does not directly optimize the effectiveness of the content, it can still provide a reasonable starting point that is easier to optimize. The CTL framework will then further improve this initialization.

### *C. Evaluation App*

Given a video, the EA is in charge of predicting its effectiveness score. Our framework can support any algorithm that, given a video, produces a numerical score indicating its effectiveness. For example, the approach proposed by Lou *et al.* [5] used in Section IV generates a score $s \in \{1,\ 2,\ 3,\ 4,\ 5\}$, where a higher score indicates better effectiveness, and an associated confidence score $c \in [0,\ 1]$, where a higher value means higher confidence. Both the effectiveness score $s$ and confidence $c$ provide useful information about the video effectiveness. We generate a final score from the summation of these two, to drive the search towards a video variant with the highest possible score class and higher confidence (secondary).

We assume in this paper that the EA is designed to predict the effectiveness of a video based on historical video consumption data. It is worth stressing that the quality of our framework is strictly connected with the quality of the EA itself. Despite this, our proposed framework is flexible enough to support a wide range of effectiveness prediction algorithms, such as the popularity on a particular platform, or an engagement score representing the time spent by the users watching the video.

### *D. Incremental Genetic Algorithm*

Given an input video $V$, our goal is to find a video summary $\hat{V}$ that maximizes the predicted effectiveness as follows:

$$\hat{R} = \arg\max_{R} E(C(R)) \qquad \hat{V} = C(\hat{R})$$

where $\hat{R}$ is the intermediate representation associated with $\hat{V}$, and $C$ and $E$ indicate the CA and EA, respectively. We design our search algorithm according to the following principles:

- The search should be time and computationally efficient;
- Each search iteration should be incremental in order to show the effect of one edit (i.e., adding/removing one shot from the summary) on the performance of the newly generated variant, which can be surfaced to the content creators as an insight on how different edits have impacted the final predicted effectiveness of the summary.

Cutting-edge IR = [10001 00000 00100 00101 00100 00000 10]
Candidate IR 1 = [10001 01000 00100 00101 00100 00000 10]
Candidate IR 2 = [10001 00000 00100 00001 00100 00000 10]
Candidate IR 3 = [10101 00000 00100 00101 00100 00000 10]

Fig. 2. Illustration of the incremental generation policy. Cutting-edge IR indicates the current summary includes 7 of 32 shots and from it 3 candidates are generated by adding/removing one shot.

To this end, we propose an *Incremental Genetic Algorithm (IGA)* to search for the best summary. IGA is an iterative algorithm that includes a generation policy and a survival policy. At iteration $n+1$, the generation policy defines how to generate a set of candidate summary variants and associated IRs $R_c^{[n+1]}$, based on the IRs $R^{[n]}$ of the previous iteration. The survival policy selects a subset of the variants $R^{[n+1]}$ with the highest effectiveness scores (as defined in Section III-C), which provides the starting point for the next iteration.

**Random-M Incremental Generation Policy** In the proposed generation policy, we introduce the constraint that only one video shot can be added to or removed from an existing summary variant to generated a new variant. More formally, this entails that only one element can be changed from an IR in iteration $n$ to generate the variant in iteration $n+1$:

$$D(R_c^{[n+1]}, R^{[n]}) = 1$$

where $D(\cdot)$ denotes the hamming distance between the two one-hot coded vectors. Fig. 2 shows an example of this policy.

Despite this constraint, a very large number of variants can still be generated (as an $L$-long IR can produce $L$ candidate IRs). To improve speed and reduce the computational overhead, we randomly select $M$ variants to be part of the candidate set to be evaluated by the EA. In our experiments, we set $M = 20$, while $L$ (the number of shots composing the video) is usually between 30 and 200.

**Top-k Survival Policy** Among all the candidate variants generated as described above, only the $k$ candidates with the highest effectiveness score (as calculated by the EA) will survive and be used to generate new variants in the next iteration. In our experiments, we set $k = 3$.

**Per-duration Top-k Survival Policy** Alternatively, as in the video summarization task we might be interested in generating a summary with a user-specified duration, we first group the candidate variants based on their duration (e.g., all summaries with 10 and 11 seconds duration), and then select the top-k candidates for each duration group. It is worth noting that variants in one group are likely to affect those in other groups as well, as their duration can change during the search.

**History Hash Map** To prevent an infinite cycle between two IRs, we set up a historical seen set to track the IRs that have already been evaluated. This guarantees that an IR is considered at most once during the search process. The memory consumption is negligible since the IR is a lightweight representation (an array) and the number of IRs are bounded by the maximum number of iterations and $M$.

**Termination Condition** The search terminates when one of these is met: (1) the number of iterations reaches the maximum limit, (2) the output video summary meets the requirement (e.g. score-5 and 90% confidence), (3) the cutting-edge IRs do not change for a few consecutive iterations (e.g. 3).

## IV. Evaluation

### *A. Creation/Evaluation Apps, Dataset, and Implementation*

We select the CA and the EA based on two off-the-shelf algorithms. We use the video summarization method by Gu *et al.* [3] as the CA. Given the frame-level features of the input video, the network proposed by Gu *et al.* aims to minimize the difference between the input features and those of the output summary. Particularly, the authors propose a GAN-based approach where a variational auto-encoder operates as generator. This unsupervised method does not require human annotations for training, and it has shown promising results when evaluated against summaries generated by human experts. Despite that, this open-loop method generates less optimal video summaries in terms of the effectiveness from an end-user perspective. As introduced in Section III-B, the open-loop CA initializes the starting point of our search algorithm. The summarization generated by the CA also acts as the baseline in our evaluation. The video effectiveness prediction network proposed by Lou *et al.* [5] is used as our EA. This work designs an LSTM-based mixture model to predict the effectiveness score of an input video into five classes, with confidence value on each class between 0 and 1. The network has been trained on the Video Ad Dataset [4], which contains rich annotations encompassing the topic and sentiment of the ads and human-generated effectiveness scores for a broad range of videos. These human scores are used as ground-truth effectiveness to train the prediction model. We use this off-the-shelf network and do not re-train the video effectiveness prediction model.

We use the same test dataset as in [5], which is a subset of 530 videos from the Video Ad Dataset [4] to evaluate our CTL framework. We implement CTL in Python 3 and evaluate in a container running on top of AWS with Intel Xeon E5-2686 v4 CPU @2.30GHz and nVidia Tesla V100 16GB GPU.

Here we present quantitative result. For additional qualitative result, please see the link: https://tinyurl.com/yykh72be.

### *B. Higher effectiveness with higher confidence*

We first evaluate our CTL framework on generating video summaries given a fixed duration, i.e. 5 seconds, and set the shot granularity as 1 second (i.e., 5 shots are selected for the summary). We compare the effectiveness score improvement over the open-loop CA, as introduced in Section IV-A. Table I presents the distribution of predicted effectiveness scores for the summaries generated by both the baseline and our approach. Using our CTL framework, we are able to increase the ratio of videos in the score-5 class from 71.5% in baseline to 98.8% in CTL. We also compare the confidence improvement in the predicted score class. The mean confidence equals to 49.6% in the baseline and increases to 65.8% in the CTL framework, which represents a 15.2% increase over the baseline. Overall, these results confirm that CTL is able to find the best summary variants for most videos with much higher effectiveness score and confidence compared to baseline.

TABLE I
Distribution of the videos in each effectiveness score class.

| Score class | Baseline | CTL | CTL, flexible duration | CTL, flexible duration & low-cost |
|---|---|---|---|---|
| 1 | 0% | 0% | 0% | 0% |
| 2 | 1.3% | 0.4% | 0% | 0% |
| 3 | 27.1% | 0.8% | 0.2% | 0.2% |
| 4 | 0% | 0% | 0% | 0% |
| 5 | 71.5% | **98.8%** | **99.8%** | **99.8%** |
| confidence | 49.6% | **65.8%** | **71.0%** | 49.2% |

TABLE II
Runtime comparison between CTL and baseline.

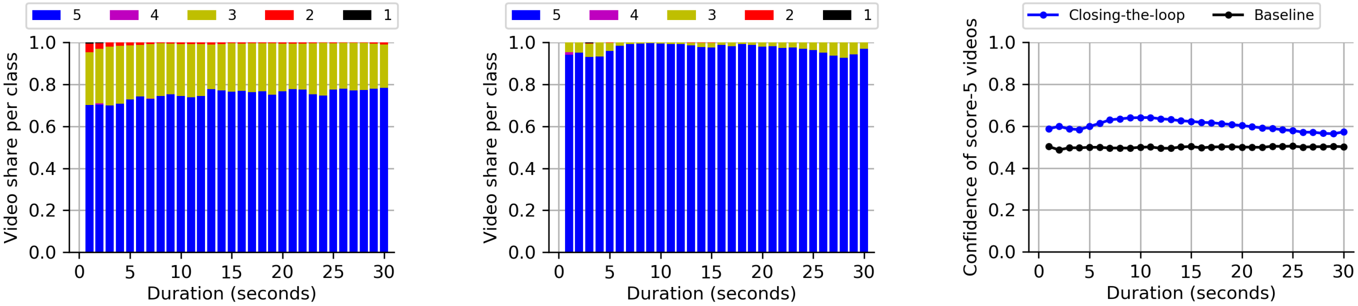
| Method (task level) | Execution time per video |
|---|---|
| Baseline, fixed duration | 159.91 sec |
| CTL, fixed duration | 288.29 sec (+80.3%) |
| CTL, flexible duration | 259.29 sec (+62.1%) |
| CTL, flexible duration and low-cost | 171.49 sec (+7.2%) |
| Baseline, every duration | 163.93 sec |
| CTL, every duration | 394.53 sec (+140.7%) |

### *C. Cost and Performance Trade-offs*

An important evaluation metric here is the computation overhead introduced by searching process. We report the runtime cost of the CTL framework in Table II for the fixed video summary duration use case. Despite the exponentially growing search space in the number of video shots to include/exclude from the summary, the proposed CTL framework adds only 80.3% overhead given a fixed duration requirement, compared to the baseline (first and second row in Table II).

We further explore several cost-performance trade-offs, based on slightly changed summarization requirements. First, we relax the constraint on the final summary duration, meaning that the final summary can be of any length. This configuration allows to generate more video variant choices and allows the IGA search to terminate in fewer iterations. The fourth column in Table I shows that an even higher amount (99.8%) of videos fall now in the score-5 effectiveness class, which is a 28.3% increase over the baseline. Table I shows the mean confidence score increases to 71.0%, which is a 5.2% increase compared to CTL with fixed duration constraint. This configuration also reduces the computational overhead. As shown in Table II-third row, the overhead of CTL over the baseline decreases to 62.1%. Consequently, we can conclude that relaxing the duration requirement improves both score and confidence and reduces execution costs, but also reduces flexibility, as the user cannot directly control the final summary duration anymore.

We next showcase how removing the constraint on generating a summary with high confidence can lead to consistent savings of runtime. In this scenario, the search will terminate as long as a score-5 summary is found. Such optimization can significantly reduce the number of iterations in the IGA. We see in Table I that the ratio of score-5 video is also 99.8%. Table II-fourth row shows that the computation cost



(a) Baseline

(b) CTL

(c) Every duration in a range

Fig. 3. Effectiveness score class distribution comparison given summaries at different duration.

over the baseline is now only 7.2%. Considering the number of summaries belonging to the highest predicted effectiveness score class is much higher in CTL compared to the baseline even for this scenario, this configuration choice provides a low-cost option to quickly find an effective video summary.

Finally, we further consider the option to output video summaries at every duration in a given range, *i.e.* 1, 2, ..., 30 seconds. These output summaries are generated at the same time by the search algorithm, in one single search pass. This would allow the user to freely pick the video summary at the preferred duration. For the baseline algorithm, generating video summaries of every duration simply means to ensemble the top-N ranked shots together, where N is the summary duration (given that, in our experiments, the shot granularity is set to 1 second). The computation cost increases slightly with respect to the baseline to 163.93 seconds, on average (Table II, fifth row). In CTL, the per-duration top-k survival policy introduced in Section III-D keeps track of the best video variants at every duration. This allows to share the best variants across multiple duration in the search process. In Fig. 3(a), we see that using the baseline video summarization approach, 22% to 30% of the output summaries cannot reach the highest effectiveness score class, for the different duration ranges. On the other hand, using CTL, almost all videos (92.7% – 99.6%) can be summarized into a score-5 summary (Fig. 3(b)). The confidence values distribution for all the score-5 summaries is shown in Fig. 3(c). Our CTL framework is able to consistently improve the confidence of score-5 summaries, independently of the duration, by 6.0% to 14.4%. Moreover, CTL is only 1.4X slower than the baseline, despite having produced a much larger number of summaries in a single search execution. This happens because during the search process, summaries can change duration and therefore end up in different duration buckets, which can consistently speed-up the search. Particularly, this configuration shows the proposed IGA design is capable of improving summary score and confidence, while allowing an efficient use of compute resources.

## V. Conclusion

We propose a data-driven framework for automatic video summarization, which exploits an IGA to efficiently generate the best possible summary maximizing the predicted content effectiveness while providing an interpretable editing path. Evaluations shows our CTL framework significantly improves the predicted effectiveness score and confidence for most videos with only modest execution overhead. Future work includes three directions. First, alternative summary variant generation methods to speed up convergence. Second, evaluate the performance of CTL with a user study to better identify its gains. Third, although CTL has been tailored for the video summarization task in this paper, it can be applied to optimize other video editing tasks and media types as well.

## References

[1] M. Rochan and Y. Wang, “Video summarization by learning from unpaired data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[2] K. Zhou, Y. Qiao, and T. Xiang, “Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] H. Gu and V. Swaminathan, “From thumbnails to summaries-a single deep neural network to rule them all,” in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1–6.

[4] Z. Hussain, M. Zhang, X. Zhang, K. Ye, C. Thomas, Z. Agha, N. Ong, and A. Kovashka, “Automatic understanding of image and video advertisements,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1705–1715.

[5] Q. Lou, S. Sarkhel, S. Mitra, and V. Swaminathan, “Content-based effectiveness prediction of video advertisements,” in *2018 IEEE International Symposium on Multimedia (ISM)*, 2018, pp. 69–72.

[6] Y. Gao, T. Zhang, and J. Xiao, “Thematic video thumbnail selection,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 4333–4336.

[7] M. Sun, A. Farhadi, and S. Seitz, “Ranking domain-specific highlights by analyzing edited videos,” in *European conference on computer vision*, 2014, pp. 787–802.

[8] H. Kim, T. Mei, H. Byun, and T. Yao, “Exploiting web images for video highlight detection with triplet deep ranking,” *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2415–2426, 2018.

[9] B. Mahasseni, M. Lam, and S. Todorovic, “Unsupervised video summarization with adversarial lstm networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[11] J. R. Sampson, “Adaptation in natural and artificial systems (john h. holland),” 1976.

[12] L. Davis, *Genetic Algorithms and Simulated Annealing*. Pitman, 1987.

[13] T. Wang, A. Mansfield, R. Hu, and J. P. Collomosse, “An evolutionary approach to automatic video editing,” in *2009 Conference for Visual Media Production*, 2009, pp. 127–134.

[14] N. A. Henriques, N. Correia, J. Manzolli, L. Correia, and T. Chambel, “Moviegene: Evolutionary video production based on genetic algorithms and cinematic properties,” in *Workshops on Applications of Evolutionary Computation*, 2006, pp. 707–711.

[15] R. Xu, J. Koo, R. Kumar, P. Bai, S. Mitra, S. Misailovic, and S. Bagchi, “Videochef: efficient approximation for streaming video processing pipelines,” in *2018 USENIX ATC*, 2018, pp. 43–56.