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

Ran Xu
Saurabh Bagchi
Purdue University
Email: {xu943, sbagchi}@purdue.edu

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

***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 consistently 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 score class than those in the baseline.**

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 a high-level of engagement with the end-users. Quantifying the effectiveness of a video is a non-trivial task for content creators, as effectiveness not only depends on the content itself but also on the target audience and publishing channels. Content creators usually rely on their experience and preference to create a short summary starting from a long video, 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 video and the summarized version [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* is responsible to generate possible summaries, based on the input content and editing parameters. An *Evaluation App* 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 *Creation App*. Ultimately, this iterative process produces the video summary with the highest predicted effectiveness by the *Evaluation App*. The main contributions of this paper are therefore two-fold:

- We design *Closing-the-Loop*, a data-driven video summarization framework that allows to automatically summarize an input video in order to maximize its predicted effectiveness, using a combination of a *Creation App*, to generate possible variants, and an *Evaluation App*, to evaluate these variants;

- We leverage a genetic algorithm to efficiently search across all possible video summary variants and focus the process on the most promising ones. This allows to search the large and complex space of possible summaries in an efficient and scalable way, with minimal computing overhead. Different from deep learning models, which are hard to interpret, our approach is more interpretable as it shows the incremental editing path leading to the final video 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.

The rest of the paper is organized as follows. Section II introduces related works in the area of ML-based video summarization. Section III presents our closing-the-loop framework, with details on the genetic algorithm we used to efficiently search among the possible video summaries. In Section IV, we evaluate our framework for the video summarization task, while Section V concludes the paper and presents several directions for future research in this domain.

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. Jiao *et al.* [7], [8] propose a three-dimensional attention model to fully explore the spatial and temporal features in the video. Ranking models for video segments are popular solutions for video highlight detection. Specific models include EM-like self-paced model selection procedures [9] and deep learning techniques [10], [11]. Researchers are also exploring new model architectures. Zhang *et al.* [12] have been the first to propose Long Short-Term Memory (LSTM) networks to model the temporal dependency among frames and build representative summaries. Gu *et al.* [3] and Mahasseni *et al.* [13] 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. Particularly, this is the model used as *Evaluation App* in our CTL framework. 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. We use the Video Ad dataset collected by the authors for our evaluation. Li *et al.* [14], Figueiredo *et al.* [15], Ding *et al.* [16], Vallet *et al.* [17], and Jing *et al.* [18] predict the popularity of videos or micro-videos on online social networks, which can also be considered an indirect prediction of effectiveness. Similarly, Gürsun *et al.* [19] describe and forecast the daily video access patterns of YouTube videos. 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 [20], [21], it remains challenging to fully interpret the decision taken by the effectiveness prediction model.

As seen above, the problem of closing the feedback loop between video summarization and its performance evaluation/prediction is currently not addressed. Previous works mostly focus on: (1) predicting the effectiveness of existing videos or (2) automatically generating summaries without directly taking into account how effective the created content would be. This paper closes this gap by allowing an iterative, step-by-step search to find the optimal video summary. To perform this search, we use a Genetic Algorithm (GA) [22], [23]. GAs have been studied extensively in the past in the context of search and optimization, and they have been used for video summarization as well [24], [25], particularly because of their flexibility and time efficiency. GAs have also been applied to generate sports video summaries, like cricket video events [26] and soccer videos [27]. More generally, such evolutionary approach can be used for automatic video editing [28] and video production [29]. GAs are particularly effective when the search space is large, non-convex or discontinuous, as in our video summarization case. They are also modular in nature, which is an important aspect of the data-driven framework proposed in this paper. Finally, the incremental nature of the optimization carried out by a GA can be used to expose the editing decisions to the creator, and to allow the creator to decide whether to accept the edits or not.

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 *Creation App* to generate different video summary variants, and an *Evaluation App* to assess the performance of the generated variants. A GA allows

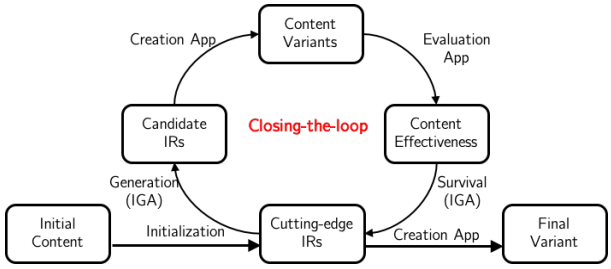


Fig. 1. The workflow of the Closing-the-Loop framework.

Fig. 2. Visualization of the Intermediate Representation for a video summarization application. A video shot is labelled in green if it is included in the summary, and in white otherwise. In this example, the summary includes five shots, namely 0th, 3rd, 10th, 15th, and 23rd shots. The first video frame of each shot is also shown.

to connect these two steps and efficiently search the best video summary variant. This design choice allows to plug any *Creation* and *Evaluation Apps* in the proposed framework to optimize the video summarization process, according to the specific requirements of the content creator. Figure 1 shows an overview of the process, which can be described as follows:

1) Given an input video, generate a set of initial variants using the *Creation App*;
2) Generate a set of candidate variants from the initial variants, based on the generation policy of the GA and the *Creation App*;
3) Use the *Evaluation App* 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 steps 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 *Evaluation App*. In the reminder of this section, we will present each step of the CTL framework in detail. Without loss of generality, we assume that the input video is pre-processed and 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 different summarization variants, which we call an *Intermediate Representation* (IR). An IR, R, is an L-long binary array, where $r_i = 1$ means that the shot, x_i, is selected for the video summary. Figure 2 provides a visualization of an IR, for a video composed of 30 shots, of which 5 are included in the summary.

B. Creation App and Search Initialization

In our data-driven framework, the *Creation App* 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 the GAN-based approach [3], which aims to minimize the difference between the visual features of the input video and those of the generated summary. In this context, this solution is open-loop because it does not consider any user video consumption data to generate the final summary.

The *Creation App* is in charge of generating the first video summary, before the genetic algorithm starts searching for the best variant. The initial IR is the *Creation App*'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 *Creation App* for initialization. Intuitively, even though the *Creation App* 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 *Evaluation App* 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 class s and confidence value c provide useful information about the video effectiveness. We generate a final score, given by the summation of these two values, 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 *Evaluation App* 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 Evaluation App 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, the goal of our framework is to find a video summary $\hat{V}$ that maximizes the predicted effectiveness as follows:

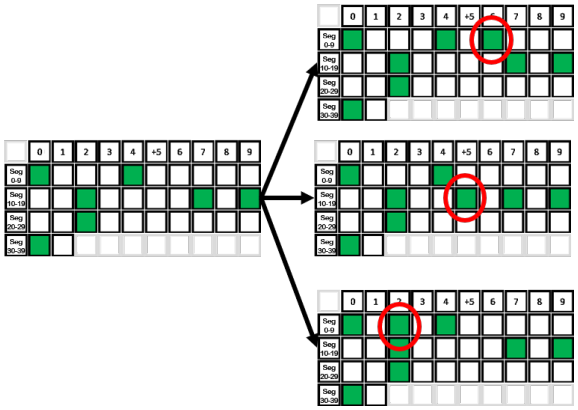


Fig. 3. Illustration of the incremental generation policy in the video summarization application. The existing IR on the left indicates that 7 out of 32 video shots have been included in the summary so far. On the right, three candidates generated from this IR by including an additional shot (highlighted with a red circle). Note that the deletion of a shot is also possible.

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

$$\hat{V} = C(\hat{R})$$

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

- The search should be time and computationally efficient;
- Each search iteration step 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. This information can be surfaced to the content creator to provide an insight on how the different edits have impacted the final predicted effectiveness of the summary.

Based on these design principles, 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. Figure 3 provides an example of this incremental generation 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 *Evaluation App*. 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 *Evaluation App*) 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 whose duration is between 10 and 11 seconds), and then select the top-k candidates for each duration group. It is worth noting that variants in a group are likely to affect variants in other groups as well, since the duration of the variants can change during the incremental 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 light-weight representation (an array) and the number of IRs are bounded by the maximum number of iterations and M.

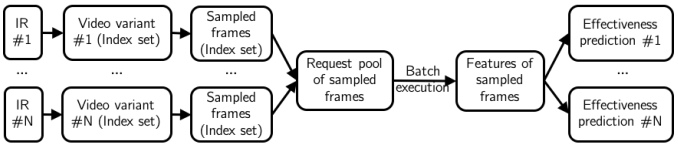
Termination Condition The search terminates when one of the following conditions is met: (1) the number of iterations reaches the maximum limit, (2) the output video summary meets the requirement (e.g. score-5 and 90% confidence), (3) the cutting-edge IRs do not change for a few consecutive iterations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

IV. Evaluation

A. Creation/Evaluation Apps, Dataset, and Implementation

To evaluate our closing-the-loop framework, we select the *Creation App* and the *Evaluation App* based on two off-the-shelf algorithms. We use the video summarization method by Gu *et al.* [3] as the *Creation App* in the CTL framework. 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 method is completely unsupervised and 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 *Creation App* initializes the starting point of our search algorithm. The summarization generated by the *Creation App* also acts as the

(a) A vanilla framework



(b) An optimized framework

Fig. 4. Illustration of the application-specific optimizations in the Closing-The-Loop framework.

baseline in our evaluation. The video effectiveness prediction network proposed by Lou *et al.* [5] is used as *Evaluation App* in the CTL framework. This work designs an LSTM-based mixture model to predict the effectiveness score of an input video into five classes $\{1, 2, 3, 4, 5\}$, 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 as proposed by the authors and do not re-train the video effectiveness prediction model.

As for the dataset used to evaluate our proposed CTL framework, we use the same test dataset as in [5], which is a subset of 530 videos from the Video Ad Dataset [4]. We implement the CTL framework 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.

B. Application-specific Speed Optimizations

We briefly describe in this Section the optimizations we carried out to improve the efficiency of our framework given our specific *Creation App* and *Evaluation App*. Figure 4(a) shows one iteration step for a vanilla CTL framework, depicting the process to go from a candidate IR to the effectiveness prediction of the associated video variant. The *Creation App* converts the IR R into a video variant V. Next, the *Evaluation App* samples a fixed number of frames from the video variant V, extracts the features of the sampled frames and computes the effectiveness score of the video variant. In this case, the execution time is dominated by the creation of the video summary variants and the prediction of their effectiveness scores. As we need to evaluate multiple IRs in our IGA, we propose the following optimizations to reduce the execution time of each iteration (see Figure 4(b)):

1) Instead of generating a video variant V out of a candidate IR R, use the index of the selected frames from the input video to present the concatenated video variant $V = \{f_1,\ f_2,\ ,\ ...,\ f_p\}$;
2) Sample the index set as if sampling the video variant $V_s = \{f_{s1},\ f_{s2},\ ,\ ...,\ f_{sk}\}$;
3) Establish a request pool P of sampled frames, motivated by the fact that the many sampled frames of different IRs are in common, $P = \cup V_s$;
4) Batch the frame extraction and feature extraction on the sampled frames;
5) Store the features of the sampled frames for future iterations.

C. 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 *Creation App*, 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. Further explorations will be discussed in Section IV-D. We also compare the confidence improvement in the predicted score class. Indeed, we are also interested in generating summaries whose predicted score is not only the highest possible, but also with the highest confidence score. This is especially important to justify the edits performed on the video with the content creator. Figure 5 shows the distribution of confidence scores for all videos whose predicted effectiveness score is 5, for both the baseline and our framework. 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.

We provide a qualitative visualization of the output video as a case study. Figure 8(a) shows the 5 representative frames of the summary generated by the baseline algorithm – the effectiveness score is equal to 3 with 54.1% confidence. Figure 8(b) shows the output generated by the CTL framework – the score class improves to 5 with 80.6% confidence. Again, further explorations will be discussed in Section IV-D.

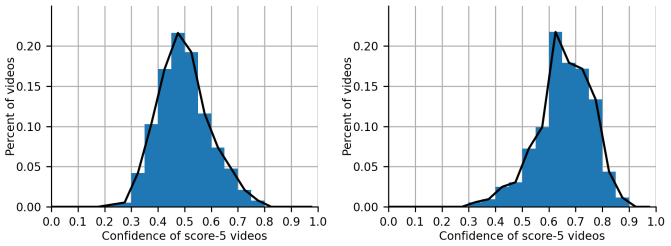
Overall, these results confirm that our proposed approach is able to find the best summary variants for most of the videos with much higher effectiveness score class and confidence compared to the baseline.

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%**

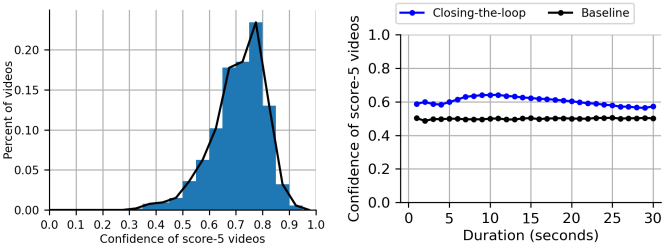
D. Cost and Performance Trade-offs

An important evaluation metric for the proposed framework is the computation overhead introduced by searching for the best content variant. We report the runtime cost of the CTL framework for the fixed video summary duration use case presented in the previous Section in Table II. Despite the



(a) Baseline (b) CTL

Fig. 5. Confidence values distribution for the videos in score-5 effectiveness score class (5-seconds summary).

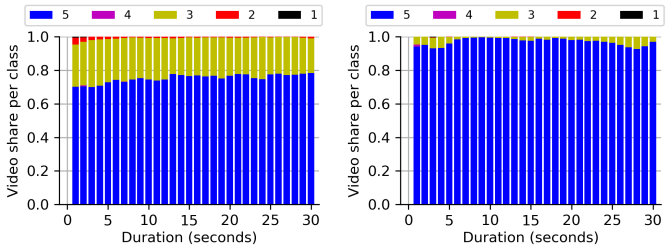


(a) CTL - Flexible duration (b) Every duration in a range

Fig. 6. Distribution of the videos in terms of confidence value in score-5 effectiveness score class.

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 the Table).

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. Figure 6(a) shows the confidence score distribution for the videos in the score-5 class. The mean confidence score increases to 71.0%, which is a 5.2% increase compared to CTL with fixed duration constraint in Figure 5(b). Qualitative results for this summarization scenario are presented in Figure 8 (c). In this case, the generated summary has a duration equal to 4 seconds. We are able to reach the same highest score class as in the fixed duration summary (Figure 8



(a) Baseline (b) CTL

Fig. 7. Effectiveness score class distribution comparison given summaries at different duration.

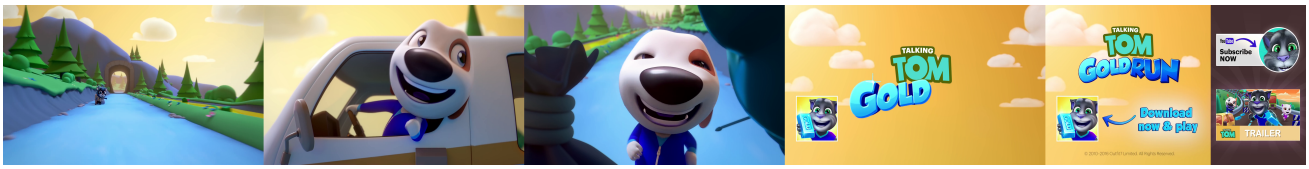
TABLE II
RUNTIME COMPARISON BETWEEN CTL AND BASELINE.

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%)

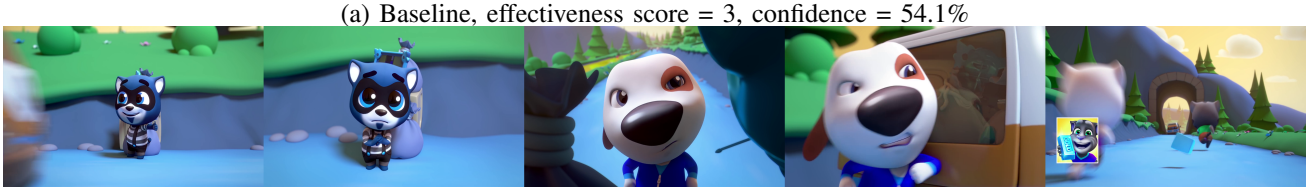
(b)) and even higher confidence – 84.4%. Another benefit of this configuration is the reduced computational overhead. As shown in Table II-third row, the overhead of CTL over the baseline decreases to 62.1%. Consequently, we can conclude that relaxing the duration requirement improves both score and confidence while reducing execution costs. This comes with a reduced 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 in terms of execution time. In this scenario, the search will terminate as long as a score-5 video 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 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.

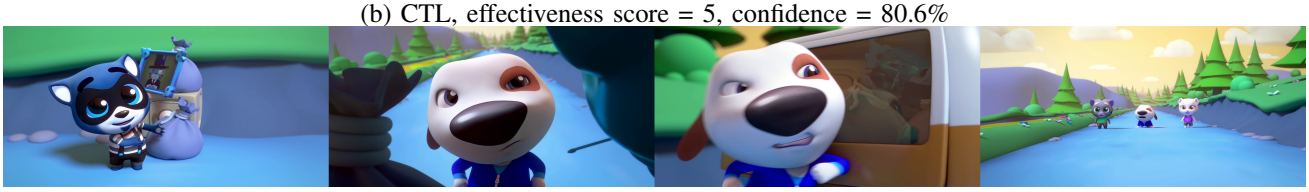
Finally, we further consider the option to output video summaries at every duration in a given range, *i.e.* 1, 2, ..., 30 seconds. These output summaries are generated at the same time by the search algorithm, in one single search pass. This would allow the user to freely pick the video summary at the 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 Figure 7(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 (Figure 7(b)). The confidence values distribution for all the score-5 summaries is shown in Figure 6(b). 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 to



(a) Baseline, effectiveness score = 3, confidence = 54.1%



(b) CTL, effectiveness score = 5, confidence = 80.6%



(c) CTL with flexible duration, effectiveness score = 5, confidence = 84.4%

(d) CTL with flexiable duration and score class target only, effectiveness score = 5, confidence = 57.4%

Fig. 8. Case study on one test video: (a) baseline achieves lower effectiveness score (b) CTL improves both effectiveness score and confidence, and output a video at a given duration (*i.e.* 5 seconds) (c) CTL improves confidence marginally with flexible duration. (d) CTL reaches highest score with minimal cost.

produce a much large number of output summaries in a single search execution. This happens because during the search process, summaries can change duration and therefore end up in different range duration buckets, which can consistently speed-up the search. Particularly, this summarization configuration shows that the proposed IGA design is capable of improving summaries score and confidence, while allowing an efficient use of computational resources.

Here we show a second case study, in Figure 9, a 2016 Scion iM TV commercial. Figure 9(a) describes the full story: Jaleel White drives with "Family Matters" wax museum Steve Urkel in the passenger seat. The dual zone automatic climate control keeps both Jaleel warm and his wax-self from melting. Jaleel finds himself starting to say Urkel's famous line, "Did I do that?" only to catch himself mid-phrase, when he sees Urkel is staring back at him with his iconic smile. As seen from Figure 9(b), the baseline summarization fails to capture the story. Instead, it chooses three similar frames in the end. In Figure 9(c), our approach selects not only the right shots but also the appropriate number of shots, which captures the original story well.

V. Conclusion

We propose in this paper a data-driven framework for automatic video summarization, which exploits an incremental genetic algorithm to efficiently generate the best possible summary maximizing the predicted content effectiveness. The incremental nature of this search would also allow content creators to understand what incremental edits have the most impact on content effectiveness and, finally, what it takes to produce an effective video summary. Our evaluation on a popular video effectiveness dataset [4] shows that our *Closing-The-Loop* framework can significantly improve both the predicted effectiveness score and confidence for most videos, compared to an open-loop baseline that only considers video-level objectives to generate a summary. We achieve this result with only modest execution overhead compared to the baseline, thanks to the efficient search carried out by the genetic algorithm.

Future work will focus on three possible directions. First, alternative summary variant generation methods to speed up convergence. Second, evaluate the performance of the framework with a user study to better identify the gains of the proposed approach. Third, although the CTL framework presented in this paper has been tailored for the video summarization task, it can be applied to optimize the performance of 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.

(a) Frames sampled from original video every 5 seconds

(b) Frames sampled from baseline summary every second, effectiveness score = 3, confidence = 45.7%

(c) Frames sampled from CTL summary every second, effectiveness score = 5, confidence = 71.9%

Fig. 9. Case study on different test video: (a) original video, (b) 5-seconds summary generated by the baseline algorithm, (c) 7-seconds summary generated by the CTL framework, which improves both the predisce effectiveness score and confidence.

[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] Y. Jiao, Z. Li, S. Huang, X. Yang, B. Liu, and T. Zhang, "Three-dimensional attention-based deep ranking model for video highlight detection," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2693–2705, 2018.

[8] Y. Jiao, T. Zhang, S. Huang, B. Liu, and C. Xu, "Video highlight detection via region-based deep ranking model," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 07, p. 1940001, 2019.

[9] 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.

[10] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 982–990.

[11] 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.

[12] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*, 2016.

[13] 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, pp. 202–211.

[14] H. Li, X. Ma, F. Wang, J. Liu, and K. Xu, "On popularity prediction of videos shared in online social networks," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 169–178.

[15] F. Figueiredo, "On the prediction of popularity of trends and hits for user generated videos," in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 741–746.

[16] W. Ding, Y. Shang, L. Guo, X. Hu, R. Yan, and T. He, "Video popularity prediction by sentiment propagation via implicit network," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 1621–1630.

[17] D. Vallet, S. Berkovsky, S. Ardon, A. Mahanti, and M. A. Kafaar, "Characterizing and predicting viral-and-popular video content," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1591–1600.

[18] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, and M. Wang, "Low-rank multi-view embedding learning for micro-video popularity prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1519–1532, 2017.

[19] G. Gürsun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns," in *2011 proceedings IEEE infocom*, 2011, pp. 16–20.

[20] 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.

[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[22] J. R. Sampson, "Adaptation in natural and artificial systems (john h. holland)," 1976.

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

[24] P. Chiu, A. Girgensohn, W. Polak, E. Rieffel, and L. Wilcox, "A genetic algorithm for video segmentation and summarization," in *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, vol. 3, 2000, pp. 1329–1332.

[25] X. Yang and Z. Wei, "Video segmentation and summarization based on genetic algorithm," in *2011 4th International Congress on Image and Signal Processing*, vol. 1, 2011, pp. 460–464.

[26] H. Narasimhan, S. Satheesh, and D. Sriram, "Automatic summarization of cricket video events using genetic algorithm," in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, 2010, pp. 2051–2054.

[27] X. Yang and Z. Wei, "Genetic keyframe extraction for soccer video," *Procedia Engineering*, vol. 23, pp. 713–717, 2011.

[28] 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.

[29] 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.