PerConfigure: Periodically Configure the Video Pipeline Based on a Reinforcement Learning approach

Written by Zhaoliang He[1], Chen Tang[1], Zhi Wang[2]
[1]Department of Computer Science and Techonology, Tsinghua University
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University

Abstract

Introduction

With the increasing demand for continuous video analysis in public safety and transportation, more and more cameras are being deployed to various locations. Video analysis can be completed according to the video analysis application built by different models misunderstand of different models, tangchen amend, which can free the staff from complex and boring tasks or search through massive amounts of video data to find what you're looking for quickly. In recent years, we have also witnessed the emergence of a large number of excellent models for target detection.

For the collected video, the classical computer vision and deep neural network technology are generally used for video analysis. A video analysis application consists of a *pipeline* of several video processing modules, typically including a decoder, a selective sampling frame application, and a target detector. Such a pipeline always has multiple *knobs*, such as frame rate, resolution, and model (e.g., MobileNet, ResNet, or InceptionResNet). A combination of the knob values is a video analysis *configuration*. The configuration space grows *exponentially* with the number of knobs and their values. Since video analysis is a very complex process, we pay much attention to the consumption of resources in the calculation process, and the accuracy of inference is also our focus. Therefore, the problem that follows is how to balance *resource consumption* and *accuracy*.

Choosing different configurations will affect the resource consumption and accuracy caused by video analysis. For example, using a complex model and high resolution can obviously accurately detect the target object, but it also requires more computing resources. However, choosing a simple model and low resolution can significantly reduce resource consumption, although it reduces the accuracy to some extent. And in the case of a highway video analysis, due to the rate of car travel cannot be predicted in advance, so when the car drives slowly (or static) because of the traffic jam, we can choose a lower frame rate (such as 1 FPS) without having to use a fixed on the whole video higher frame rate, this can significantly reduce resource consumption, but does not affect the accuracy of the video analysis.

The *best* configuration for a video analytics pipeline also varies over time, often at a timescale of minutes or even seconds (Jiang et al. 2018). Hence, our goal is to find a range of "most appropriate" configurations that takes up the minimum amount of computing resources and is accurate to the desired threshold. On the one hand, if one only profiles the processing pipeline to choose the best configuration *once*, the application would either waste resources (by picking an expensive configuration) or sacrifice accuracy (by picking a cheap configuration). On the other hand, if one periodically profiles the pipeline configurations to find an optimal resource-accuracy *tradeoff* by exhaustive all configurations, it would be prohibitively expensive since the number of possible configurations is exponential in the number of knobs and their values, and thousands of configurations can be combined with just a few knobs.

For a video analytics application, choosing the "most appropriate" configuration is a complicated decision-making problem, which is challenging to be solved by rules. The reinforcement learning method is an excellent way to solve this unsupervised complex-environmental problem. In our solution, we tackle the following design challenges.

- *The best configuration for a video analytics pipeline changes over time with the environment.* The real-word environment is non-stationary; for instance, tracking vehicles when traffic moves quickly requires a much higher frame rate than when traffic moves slowly, but when each condition occurs may vary by hour, minute, or second. As a dynamic video analysis configuration solution, we target to provide a solution that dynamically picks a configuration according to intrusive dynamics of video contents, i.e., it can *generate* video analysis configuration for video analysis in a different time.
- *How to significantly reduce the resource cost of periodic configuration profiling.* The cost of periodically profiling often exceeds any resource savings gained by adapting the best configurations we end up selecting. We leverage a Reinforcement Learning-based agent to automatically

pick the best configuration periodically, dramatically reducing the profiling cost.

- *Lack of well-labeled training data.* In our problem, one is not provided the well-labeled data on which configuration should be used in which time of the video, as in conventional supervised deep learning tasks. In practice, such a video analysis configuration is usually utilized in an online manner, and the solution has to learn from the video contents automatically.

To address the above challenges, we present a periodically configure approach based on reinforcement learning, called PerConfigure, which can dynamically select the "most appropriate" configuration according to intrusive dynamics of video contents, thus solving this difficult optimal configuration decision problem in a very low-cost way. The main contributions of this paper are summarized as follows. zhaoliang add more contributions

- We design an interactive training environment that can be applied to different online computer vision-based services. We propose an asynchronous advantage actor-critic-based (Mnih et al. 2016) agent to evaluate and predict the performance of a video analysis configuration on video analysis.
- We build a reinforcement learning-based framework to train the agent in the above environment. The agent can learn to choose a "most appropriate" configuration for each timestamp of video analysis after iteratively interacting with the environment by feeding the carefully designed reward that considers both accuracy and resources.
- PerConfigure can achieve xx-xx% higher accuracy with the same amount of resources, or achieve the same accuracy with only xx-xx% of the resources.

Related Works

Static configuration optimization

Several previous papers have considered optimizing video processing pipelines by either adjusting the configuration knobs or training specialized NN models. VideoStorm (Zhang et al. 2017) profiles thousands of video analytics queries on live video streams over large clusters, achieving resource-quality tradeoff with multi-dimensional configurations. VideoEdge (Hung et al. 2018) introduces *dominant demand* to identify the best tradeoff between multiple resources and accuracy, and narrows the search space by identifying a "Pareto ban" of promising configurations. MCDNN (Han et al. 2016) provides a heuristic scheduling algorithm to adaptively select model variants of different accuracy for deep stream processing under resource constraints. Focus (Hsieh et al. 2018) deconstructs video analytics into two phases, i.e., video ingest and video query. By tuning the share of computing resources of both phases, Focus achieves effective and flexible tradeoff of latency and accuracy of video analytics. These algorithms all profile and optimize video analytics only once at the beginning of the video. They do not handle changes in video stream content. But the optimal configurations do change over time because of the complex and changeable environment.

Dynamic configuration optimization

Some papers study how to dynamically optimize the configuration for video analytics when the video stream content changes. (Shen et al. 2017) adaptively retrains the NN model to detect the set of popular objects as it changes over time in the video classification task. (Yang et al. 2019) proposes an online video quality and computing resource configuration algorithm to gradually learn the optimal configuration strategy, effectively improving the analytic accuracy while providing the low-latency response. INFaaS (Romero et al. 2019) automatically selects a model, hardware architecture, and any compiler optimizations, and makes scaling and resource allocation decisions when application load varies and the available resources vary over time. JCAB (Wang et al. 2020) jointly optimizes configuration adaption and bandwidth allocation to address several critical challenges in edge-based video analytics systems, including edge capacity limitation, unknown network variation, intrusive dynamics of video contents. The online algorithm effectively balances analytics accuracy and energy consumption while keeping low system latency.

The closest work to ours is Chameleon (Jiang et al. 2018), which dynamically picks the best configurations for video analytics pipelines, reducing resource consumption with little degradation in accuracy. They leverage temporal and spatial correlation to amortizes the cost of profiling over time and across multiple cameras, and exploit the knob independence to reduce the search space from exponential to linear. Even the search space is linear, and the profiling cost is still expensive. Such a 24-hours video, Chameleon profiles the configuration space once in every profiling window (16s), it would profile 5400 times. One profiling cost grows linear in the number of configuration knobs and the number of values per knob. The total profiling cost which is equal to one profiling cost multiply the number of profiling (5400) is also significantly high. a little strange, can tangchen help amend? Our solution called PerConfigure leverages a reinforcement learning-agent to choose the best configuration periodically, significantly reducing the cost of profiling since the choosing time is extremely lower.

Detailed Design

Evaluation

Experiment Setup

some results

Model and Image size	Inference time	Accuracy
SSD MobileNetV2 320p	49.5 ms	xxx
SSD MobileNetV2 640p	58.5 ms	0.753
SSD ResNet152V1 640p	100 ms	0.886
SSD ResNet152V1 1024p	182.3 ms	0.942
FasterRCNN ResNet50V1 640p	106.4 ms	0.889
FasterRCNN ResNet50V1 1024p	120.5 ms	0.98
FasterRCNN InceptionResNetV2 640p	361.8 ms	0.965
FasterRCNN InceptionResNetV2 1024p	418.4 ms	1

Table 1: Inference time and F1 for different models and resolutions

Conclusion

Acknowledgments

References

Han, S.; Shen, H.; Philipose, M.; Agarwal, S.; Wolman, A.; and Krishnamurthy, A. 2016. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 123–136.

Hsieh, K.; Ananthanarayanan, G.; Bodik, P.; Venkataraman, S.; Bahl, P.; Philipose, M.; Gibbons, P. B.; and Mutlu, O. 2018. Focus: Querying large video datasets with low latency and low cost. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 269–286.

Hung, C.-C.; Ananthanarayanan, G.; Bodik, P.; Golubchik, L.; Yu, M.; Bahl, P.; and Philipose, M. 2018. Videoedge: Processing camera streams using hierarchical clusters. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 115–131. IEEE.

Jiang, J.; Ananthanarayanan, G.; Bodik, P.; Sen, S.; and Stoica, I. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 253–266.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Romero, F.; Li, Q.; Yadwadkar, N. J.; and Kozyrakis, C. 2019. Infaas: A model-less inference serving system. *arXiv preprint arXiv:1905.13348*.

Shen, H.; Han, S.; Philipose, M.; and Krishnamurthy, A. 2017. Fast video classification via adaptive cascading of deep models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3646–3654.

Wang, C.; Zhang, S.; Chen, Y.; Qian, Z.; Wu, J.; and Xiao, M. 2020. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics. In *Proc. IEEE INFOCOM*, 1–10.

Yang, P.; Lyu, F.; Wu, W.; Zhang, N.; Yu, L.; and Shen, X. S. 2019. Edge coordinated query configuration for low-latency and accurate video analytics. *IEEE Transactions on Industrial Informatics* 16(7):4855–4864.

Zhang, H.; Ananthanarayanan, G.; Bodik, P.; Philipose, M.; Bahl, P.; and Freedman, M. J. 2017. Live video analytics at scale with approximation and delay-tolerance. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 377–392.