AUTOCONFIGURE: A CONTEXT-DRIVEN AUTOMATIC CONFIGURATION FRAMEWORK FOR VIDEO ANALYTICS PIPELINE

Zhaoliang He[1], *Yuan Wang*[3], *Chen Tang*[1], *Zhi Wang*[2], *Wenwu Zhu*[1]

[1]Department of Computer Science and Techonology, Tsinghua University, China
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[3]Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, China

ABSTRACT

Deep convolutional neural networks (NN)-based video analytics demands intensive computation resources and high inference accuracy. A NN-based video analytics *pipeline* includes multiple *knobs*, such as frame rate, resolution and target detector model. A combination of the knob values is a video analytics *configuration*. Due to the highly variable video content, the *best* configuration for a video analytics pipeline also varies over time. Periodically profiling the processing pipeline to find an optimal resource-accuracy *trade-off* 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. Searching a large space of configurations periodically causes an overwhelming resource overhead that far outstrips the gains of periodically profiling.too long Knowing this, designing an *automatic* approach to decide what is the best configuration for the current video content is meaningful. In this paper, we propose a reinforcement learning (RL)-based automatic video analytics configuration framework, AutoConfigure.add RL challenges In particular, we design a reinforcement learning agent that learns the *optimal* configuration based on its own experience to decide the best configuration (the most fitting configuration) for current video content.not necessary The unique feature of AutoConfigure is *content-driven*, meaning that AutoConfigure can adapt the best configuration to intrusive dynamics of video contents. add more context-driven We implement and evaluate this approach in object detection task comparing its performance to static configuration and dynamic configuration baseline. We show that AutoConfigure achieves 20-30%(goal) higher accuracy with the same amount of resources, or achieve the same accuracy with only 30-50%(goal) of the resources. Furthermore, AutoConfigure proves to be more efficient than existing baselines by creating an overhead of less than 1%(goal) to the overall video analytics pipeline.compare to chameleon

1. INTRODUCTION

With the increasing demand for continuous video analytics in public safety and transportation, more and more cameras are being deployed to various locations. The video analytics are based on classical computer vision techniques as well as deep convolutional neural networks. In recent years, we have also witnessed the emergence of a large number of excellent models for target detection [1], such as FasterRCNN [2], RFCN [3], Multibox [4], SSD [5] and YOLO [6].

For the collected video, the classical computer vision and deep neural network technology are generally used for video analytics. A video analytics 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., SSD+{MobileNet [7], ResNet [8]}, FasterRCNN+{ResNet,InceptionResNet [9]}). A combination of the knob values is a video analytics *configuration*. The configuration space grows *exponentially* with the number of knobs and their values [10]. Since video analytics applications demand intensive computation resources and high accuracy, we pay much attention to the consumption of resources in the calculation process and the accuracy of inference. Therefore, the problem that follows is how to balance *resource consumption* and *accuracy*.

Apparently, choosing different configurations will affect resource consumption and accuracy. As shown in Figure 1, at a fixed frame rate, using a complex model with high resolution (such as FasterRCNN+InceptionResnet 1024p) can obviously and accurately detect the target object, but it also requires more computing resources. However, when using the same model, choosing a lower resolution can reduce the computing resources, and the subsequent cost is the decline of accuracy. Another example is that choosing a simple model with low resolution (such as FasterRCNN+ResNet50 640p) can significantly reduce resource consumption, although it reduces the accuracy to some extent. Obviously, when the target car (object) is large, a smaller model and a lower resolution can meet the accuracy with a large reduction of resource consumption, while the target car (object) is small, a more complex model and higher resolution are needed to achieve a satisfactory accuracy. And in the case of a highway video analytics, 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) instead of having to use a constant high frame rate throughout whole video. This can significantly reduce resource consumption, but does not affect the accuracy of the video analytics. We use Figure 2 to support that, the data in this figure come from a real road. It can be found that with the decrease of frame rate, the resource consumption can be significantly reduced, but the decrease of accuracy is partly acceptable.

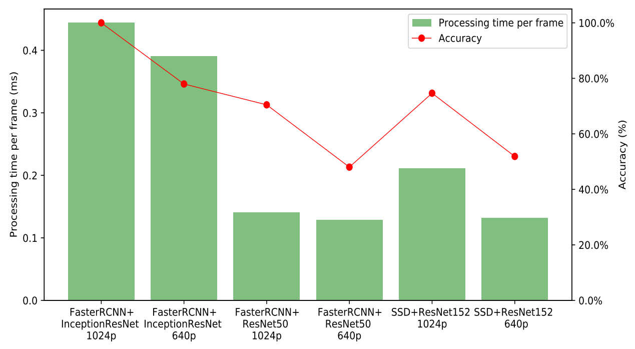

Fig. 1. The effect of different models and resolutions on accuracy and processing time at a fixed frame rate.

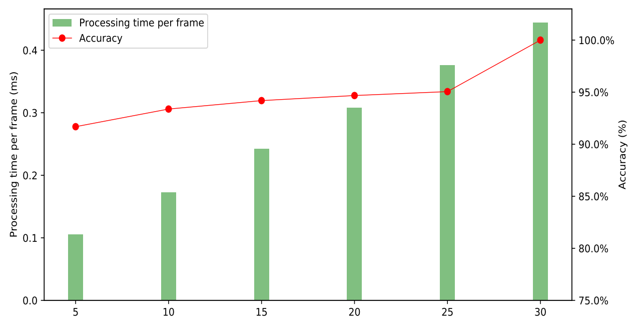

Fig. 2. Under the best model, the accuracy and processing time vary with the different frame rate.

The *best* configuration for a video analytics pipeline also varies over time, often at a timescale of minutes or even seconds [10]. 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 processing pipeline to find an optimal resource-accuracy *tradeoff* by exhaustive all configurations, it would be prohibitively expensive since the configuration space is extremely large, 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. An automatic approach is needed, one that is able to learn from video contents, to decide what is the best configuration for the current video content. 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-world 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 analytics 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 analytics configuration for video analytics 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 analytics 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 content-driven automatic configuration framework based on reinforcement learning, called AutoConfigure, 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 video analytics applications. We propose an asynchronous advantage actor-critic-based [11] agent to pick the best configuration for video analytics. Also, we define the elements of the RL-based approach, such as actions, states and rewards, to adapt the configuration to current video content.

- 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 analytics after iteratively interacting with the environment by feeding the carefully designed reward that considers both accuracy and resources.

- AutoConfigure can achieve xx-xx% higher accuracy with the same amount of resources, or achieve the same accuracy with only xx-xx% of the resources. Furthermore, AutoConfigure proves to be more efficient than existing baselines by creating an overhead of less than xx% to the overall video analytics pipeline.

2. RELATED WORKS

2.1. Static configuration optimization

Several previous papers have considered optimizing video analytics pipelines by either adjusting the configuration knobs or training specialized NN models. VideoStorm [12] profiles thousands of video analytics queries on live video streams over large clusters, achieving resource-quality tradeoff with multi-dimensional configurations. VideoEdge [13] 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 [14] provides a heuristic scheduling algorithm to adaptively select model variants of different accuracy for deep stream processing under resource constraints. Focus [15] 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. In their works, video is divided into pictures at a constant frame rate, and the work of video analyzing is regarded as a fixed task either. In other words, they do not handle changes in video stream content. But the optimal configurations do change over time because of the complex and changeable video stream contents.

2.2. Dynamic configuration optimization

Some papers study how to dynamically optimize the configuration for video analytics when the video stream content changes. [16] adaptively retrains the NN model to detect the set of popular objects as it changes over time in the video classification task. [17] 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 [18] 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. QuickAdapt [19] uses descriptive statistics of security events data and fuzzy rules to enable a Big Data Cyber Security Analytics (BDCA) system to quickly adapt to the changes in security events data. JCAB [20] 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. [21] leverages tabular Q-learning to adapt configuration, but their agent's states only consider last latency. Our framework pays attention to the complex and changeable video stream contents, and picks the best configuration according to current video content.

The closest work to ours is Chameleon [10], 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.[10] Notice that Chameleon still has non-trivial room for improvement compared to the optimal (idealized) performance, i.e., periodic updating with zero profiling cost. 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 is also significantly high, which is equal to one profiling cost multiply the number of profiling (5400). Our solution, called AutoConfigure, leverages a reinforcement learning-agent to automatically choose the best configuration periodically, significantly reducing the cost of profiling since the agent's choosing time is extremely lower.

Model and Image size	Inference time	Accuracy
SSD MobileNetV2 320p	49.5 ms	xxx
SSD MobileNetV2 640p	58.5 ms	0.494
SSD ResNet152V1 640p	100 ms	0.579
SSD ResNet152V1 1024p	182.3 ms	0.614
FasterRCNN ResNet50V1 640p	106.4 ms	0.637
FasterRCNN ResNet50V1 1024p	120.5 ms	0.786
FasterRCNN InceptionResNetV2 640p	361.8 ms	0.734
FasterRCNN InceptionResNetV2 1024p	418.4 ms	1

Table 1. Inference time and F1 for different models and resolutions car truck

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 2. Inference time and F1 for different models and resolutions

3. EXPERIMENT

3.1. Experiment Setup

3.2. some results

4. CONCLUSION

5. REFERENCES

[1] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in *Advances in neural information processing systems*, 2016, pp. 379–387.

[4] Christian Szegedy, Scott Reed, Dumitru Erhan, Dragomir Anguelov, and Sergey Ioffe, “Scalable, high-quality object detection,” *arXiv preprint arXiv:1412.1441*, 2014.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” *arXiv preprint arXiv:1602.07261*, 2016.

[10] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica, “Chameleon: scalable adaptation of video analytics,” in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 253–266.

[11] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, 2016, pp. 1928–1937.

[12] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman, “Live video analytics at scale with approximation and delay-tolerance,” in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 377–392.

[13] Chien-Chun Hung, Ganesh Ananthanarayanan, Peter Bodik, Leana Golubchik, Minlan Yu, Paramvir Bahl, and Matthai Philipose, “Videoedge: Processing camera streams using hierarchical clusters,” in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 115–131.

[14] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy, “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*, 2016, pp. 123–136.

[15] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu, “Focus:

Querying large video datasets with low latency and low cost," in *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, 2018, pp. 269–286.

[16] Haichen Shen, Seungyeop Han, Matthai Philipose, and Arvind Krishnamurthy, "Fast video classification via adaptive cascading of deep models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3646–3654.

[17] Peng Yang, Feng Lyu, Wen Wu, Ning Zhang, Li Yu, and Xuemin Sherman Shen, "Edge coordinated query configuration for low-latency and accurate video analytics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4855–4864, 2019.

[18] Francisco Romero, Qian Li, Neeraja J Yadwadkar, and Christos Kozyrakis, "Infaas: A model-less inference serving system," *arXiv preprint arXiv:1905.13348*, 2019.

[19] Faheem Ullah and M Ali Babar, "Quickadapt: Scalable adaptation for big data cyber security analytics," in *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*. IEEE, 2019, pp. 81–86.

[20] Can Wang, Sheng Zhang, Yu Chen, Zhuzhong Qian, Jie Wu, and Mingjun Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *Proc. IEEE INFOCOM*, 2020, pp. 1–10.

[21] Mauricio Fadel Argerich, Bin Cheng, and Jonathan Fürst, "Reinforcement learning based orchestration for elastic services," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 352–357.