Reuse Experts in Continuous Learning to Cope with Data Drift

1st Yuan Ou
School of Computer Science and Technology
Xi'an Jiaotong University
Xi'an, China
yukiaria@stu.xjtu.edu.cn

***Abstract*—Continuous learning has become a famous method by retraining light-weighted experts with sampled video frames from each video stream to cope with the data drift in real time video analysis area. However, retraining models consumes a considerable amount of computational resources, affecting video inference and leading to a decline in inference accuracy and throughput. Given that video streams often display temporal or spatial consistency (for instance, vehicles on the same route will pass through the same video scenes; similar video scenes will be brought about by the same lighting and weather in different places), reusing previous expert models on different cameras holds potential.To validate the potential of reusing models, I carried out a comparative experiment between reusing and retraining models on the Cityscapes dataset. It was observed that as the number of cameras increased progressively, model reuse brought about an improvement of around 10 mAP compared to model retraining.**

***Index Terms*—video analysis, edge compute, continuous learning, model reuse**

I. Introduction

Deep neural network (DNN)-based image analysis technology has broad application prospects in corporate security, retail, traffic management, transportation, and other fields. In these real applications, the system needs to be deployed in edge devices to carry out analysis work directly, such as using local edge servers, to provide real-time results [1]. However, edge computing resources cannot support the continuous growth of video analysis workload, DNN models, and video streams [2], [3]. For applications running in resource-rich environments, such as shared clouds, despite recent progress in DNN resource utilization efficiency, the cost of running video analysis is still high. For example, the high-end NVIDIA V100 GPU can only support the YOLOv5-L [4] at a speed of 30 FPS while processing two-channel video streams, and the cost on shared clouds is as high as $1100/month/stream [5].

The resources provided by edge computing are limited. The inconsistency between the growth rate of model computation requirements and processor computation cycles further exacerbates this boundary. Therefore, edge deployment is dependent on model compression. The compression of DNNs was initially trained for representative data in each video stream, but when used in the field, DNNs are affected by data drift, that is, there is a significant difference between on-site video data and the data used in training. Over time, cameras installed on streets and intelligent cars will see a variety of scenes with different lighting, different crowd densities, and different object combinations. Even small changes in the scene can affect the accuracy of inference, but it is difficult to reflect all changes during training. Therefore, data drift greatly reduces the accuracy of edge DNNs. In fact, compressed DNNs have fewer weight values and shallow structures, which are not suitable for providing highly accurate predictions when data changes greatly. Such expert DNNs require continuous retraining to maintain high inference accuracy under the limitation of the number of object appearances and scenes that can be learned in their compressed structure. Recent research in the field of computer vision and systems [6]–[8] shows that this method is effective in edge video analysis, providing high resource efficiency and result accuracy.

Model training is much more expensive than model inference, and these systems [7], [8] need to spend a total of 70%-90% of computational resources to retrain their expert models, which makes adaptive services become a key bottleneck for the resource efficiency and accuracy consistency of the video analysis system. In addition, knowledge distillation requires large-teacher models to annotate sample frames, which also occupy computational resources. In order to solve data drift in a resource-limited environment, it is necessary to reduce the system's need for retraining.

Research indicates that video streams often exhibit cyclical patterns in time and space [9], [10], that is, video scenes with similar environmental factors such as lighting (morning or evening), weather (clear or rainy), and location (drones revisit the same street) recur on the same camera. Importantly, video scenes from one camera may also occur on other cameras, especially those in geographically adjacent locations, such as an autonomous car passing through places where other cars on the same route have passed. These temporal and spatial correlations mean that some expert models that were trained on video scenes in the past can perform reasonably on current video scenes. Therefore, these historical expert models can be utilized to minimize the need for retraining.

II. Related Works

A. Continuous Learning for video analysis

In resource-constrained environments like Mobile Edge Computing (MEC) networks [11], the currently most advanced

general-purpose DNNs are often too expensive to be continuously used for video analysis. A common approach is to deploy specialized and compressed DNNs (or "expert" models) that are trained using knowledge from general and more costly DNNs (or "teacher" models). The idea is to utilize knowledge distillation [12] to transfer knowledge from a large teacher model to a smaller expert model tailored for specific video clips or streams. On matching video clips, expert models can save orders of magnitude of computational resources while achieving model precision similar to the large teacher models.

Given that expert models are only capable of identifying a confined assortment of object looks and video setups, their performance isn't optimal on dynamic live videos, which undergo unavoidable changes in objects and scenes over time such as various positioning, light conditions, and object categories [8]. An encouraging technique for utilizing expert models in dynamic real-time videos would involve recurrently retraining the expert model utilizing the most recent video frames. The most recent research [6]–[8], [13] has made it clear that the perpetual retraining and application of more compact, specialized models can yield both high precision and resource sustainability when dealing with constantly changing video content. It's also worth noting that this method of constant retraining has proven to be more effective than using a larger, instructor-based model employed on only a select amount of frames and then extrapolating the labels (such as via optical flow tracking techniques) [7].

Fig. 1 serves as an illustration of the high-level components of a video analysis system capable of continuous retraining and deployment of expert models. The major components include: (1) camera service: It periodically sends new sample video frames to the adaptation service. (2) adaptation service: It fine-tunes the camera's expert model (a copy) using recent sample frames, emulating the larger teacher model in the current scene, then sends the updated expert model to the inference service. And (3) inference service: It uses the received lightweight expert model to perform real-time inference on video frames from the camera service.

Retraining expert models demands significant computational resources and time, resulting in the adaptation service being imperative for resource efficiency and consistency in accuracy, thus serving as a key bottleneck. These systems [7], [8] require a total of 70%-90% of their computational resources to retrain their expert models, due to the much higher cost of model training compared to model inference. Additionally, knowledge distillation requires the operation of a large-scale teacher model to generate data labels on sample frames. Therefore, as a response to this fundamental challenge, an efficient method is needed to minimize the necessity for extensive expert retraining.

B. Optimization of Video Analysis Systems

Optimization of video analysis systems is a dynamic field that aims to enhance effectiveness and efficiency in the processing, interpretation, and use of data derived from video sources. Ibrahimet et al. [14] focuses on how to improve

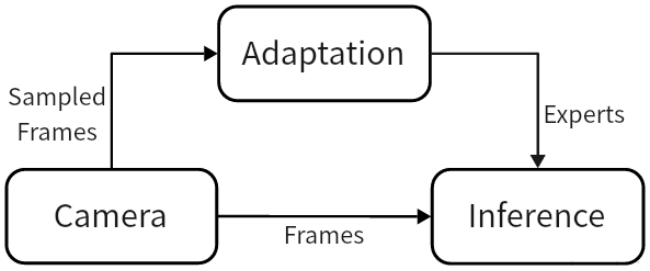

Fig. 1. Architecture of a continuous learning video analysis system.

the end-to-end performance of video analysis systems by optimizing video analytics systems' performance, including neural network-based methods. Hua et al. [15] revolves around the application of video analysis for the optimization of sports techniques, in this case, running. Arun et al. [16] addresses the reduction of transmission time and the increase in video resolution in IP-based video analytical systems. The focus of Hua et al. [17] is on automating home video editing using a set of video and music analysis algorithms to produce near-professional results.

In order to maintain high inference accuracy in environments demanding low resource utilization and speedy responses, video analysis systems have explored various methods, including model distillation [12], model architecture pruning, adaptive configuration, and frame selection.Model distillation entails creating a lightweight model that is small, fast, and adequately accurate for specific scenes. The challenge with this approach is that changes in the scene can lead to drops in the model's accuracy; therefore, the system has to create new expert models to adapt to new scenes. Existing solutions rely on the following two methods: Model retraining techniques: retraining on the latest video frames [6]–[8] or the most relevant images in the training set [13]; Model selection techniques: selecting models from a collection of historical models [10] or cascading models with increased capacity [13].

C. Model Selection under Data Drift

A naive solution to the model selection problem is to exhaustively search all the experts. However, the number of experts may be large when dealing with multiple video streams, and testing all the experts on the sampled frames of each update window to select models becomes prohibitively costly. To scale model selection to large ensembles of experts, RECL [18] uses the Mixture of Experts architecture, which is a gating network to directly infer which models in the ensemble are more appropriate for a given video content. The gating network is a lightweight DNN that, given a picture, infers a score for each model in the expert ensemble. The higher the score, the higher the likely accuracy of the model on the picture. Conceptually, the gating network is like an image classifier, except the labels are not object classes but models in the model zoo.

The Mixture of Experts (MoE) is a form of ensemble learning method that models complex input-output relationship by strategically combining the decisions of multiple "experts" or simpler models, with each being highly proficient at a certain subset of the data space. The model dynamically determines which expert to rely upon by computing a weighted combination, where the weights are typically found using a "gating" function that quantifies the expert's reliability in regard to a specific input. An indicative study on MoE is [19], where the concept was applied to solve classification problems. The model's broad range applicability and efficacy can be appreciated in diverse studies such as [20] wherein the researchers embedded MoE into a deep learning framework to scale network capacity.

After Shazeer et al. [20] found that MoE can significantly reduce the computational cost of DNN, MoE has attracted widespread attention. Recent works have achieved precision equivalent to advanced models requiring extensive computational resources with few computational resources [21].

To manage MoEs gradually comprising new models, the video analysis system must retrain the gating network or model selector. To avoid retraining the model selector (which would consume a significant amount of computational resources and time), recent work [10] uses autoencoders to project input video frames to latent space and map new models to a region in the latent space. Models are selected by looking for the model region to which the new video frames belong in the latent space, so no retraining of the selector is needed when adding new models. However, the training of the autoencoder is to learn the distribution of only the input data, not to simply learn which frames can share a good expert model. The former task is too general, making it difficult to learn an efficient encoder for deployment in practice.

Another approach to model selection [10], [22] is to map video content into an embedding space (via an autoencoder), partition the embedding space, and map each partition to a specific expert model. This approach does poorly in practice [18]. The intuitive reason is that training an autoencoder is to learn the distribution only of input data (e.g., which video frames look similar) rather than simply learning which frames can share a good expert model. The former task is too generic, thus, learning an efficient autoencoder and deploying it in practice is very challenging.

D. Resource Allocation for DNNs

Resource Allocation for DNNs centers around efficiently managing computational resources to optimize the performance and efficacy of DNNs. Managing these resources involves allocating and distributing elements such as power, networking capacity, processing ability, and memory storage across different parts of the network. This has implications not only in improving the speed of computations, but also in ensuring the reliability and robustness of DNNs.

Taking reference from multiple scholarly works, we see different angles approached for resource allocation in DNNs:One study by Sun et al., 2017 showcases the application of DNNs in wireless resource management, aimed at optimizing the non-linear mapping between input and output of resource allocation. Zhou et al., 2018 propose a resource allocation strategy based on DNNs specifically for cognitive radio networks, showcasing a specific implementation context of resource allocation. Yang et al., 2019 explored the role of DNNs for resource management in Non-Orthogonal Multiple Access (NOMA) networks, illustrating the deep relationships between resource allocation and network architecture. Gao et al., 2019 investigated power allocation considerations for DNNs, highlighting the role of resource allocation in calibrating performance.

Resource sharing for DNN-related tasks has been widely studied in the systems literature. This includes sharing of GPU and network resources among multiple concurrent DNN training tasks, inference tasks for video analysis, as well as resource sharing between inference and training tasks [8]. The common challenge these settings face is predicting how much each task's precision can be improved given the same amount of computational/network resources. Prediction methods can be categorized as offline, periodic, or through reusing computational data.

III. IMPLEMENTATION

A. Datasets

I evaluate these two methods on object detection using Cityscapes dataset [1]. The Cityscapes dataset is a comprehensive library focusing on the semantic understanding of urban street scenes. It contains high-quality, pixel-level annotations of 5,000 images from 50 different European cities, with more than 30 categories being marked and segmented per scene. The dataset supports advanced machine learning tasks including, but not limited to, semantic segmentation, instance segmentation, and object detection. It serves as a pivotal asset in the development of autonomous driving technology, as it facilitates the understanding and identification of diverse urban elements like pedestrians, vehicles, and road markings.

B. Models

In object detection, I used YoloX-Nano and YoloX-x [1] as the expert and teacher models respectively. Both models were pre-trained on the COCO [1] dataset. More details about the models can be found in Table I.

C. Metrics

In order to evaluate the accuracy of various methods, I compare the results output by the expert models with the ground truth provided in the dataset. I use *mean Average Precision (mAP)* as the benchmark for object detection, only considering the eight categories that are most pertinent to road condition detection within the Cityscapes dataset: car, pedestrian, truck, bus, rider, caravan, motorcycle, bicycle.

TABLE I
MODEL SPECIFICATIONS

Model	Speed V100 (ms)[a]	Params (M)	FLOPs (G)
YoloX-Nano	3.2	0.91	1.08
YoloX-x	17.3	99.1	281.9

[a]time required to process a single frame using NVIDIA V100.

D. Setup

In my experiment, since the Cityscapes dataset only provides sampled images, I divided every 10 images into a group, defined as a retraining window. Each retraining window has a retraining time of 30 seconds. And also, model selection and retraining happen in a adaption center (e.g., in the cloud or an edge compute cluster, etc.), and edge devices only perform inference tasks. Instead of simulating a real system, I saved each expert used and conducted the tests all at once at the end. Since the chosen expert (YoloX-Nano) can perform real-time inference on edge devices with inferior computational capabilities (like NVIDIA Jetson Nano) with an inference throughput of up to 30FPS, this method can be used to simulate actual results.

E. Baselines

I compared the performance of the following four methods:

- No Adaptation: train a single expert model based on all training data, and deploy this expert on the test data.
- Continuous Retraining: periodically retrain an expert model for each camera using the most recent video segments. This can serve as a method for recent model retraining systems, such as AMS [7] and Ekya [8]. The retraining algorithm is shown in Algorithm 1. The early stop time μ is a hyperparameter and here I set it to 1s.
- Ideal Model Reuse: Deploy the best expert model from a collection of experts each trained using data from different cities in the training set (ignoring the overhead of model selection). This can be regarded as a strictly better version of ODIN [10].
- Ideal Reuse with Retraining: Combining 2 and 3 (retraining the expert model selected in 3). This shows how much improvement an ideal model reuse scheme can bring in a continuous retraining framework.

IV. RESULTS

A. Benefits in Resource Efficiency

Fig. 2 shows the mean Average Precision (mAP) score on the test data while varying the number of cameras. The observations are two-fold.

To begin with, in the quest to minimize retraining, model reuse emerges as a propitious approach. The merits of this technique become notably apparent when computational resources are insufficient for the retraining of expert models on more expansive camera networks (comprising 4, 6, and 8 cameras). Happily, even when computing resources are sufficient for retraining (as in the case with 2 cameras), the Ideal Model Reuse approach can still match the mAP of Continuous Retraining. This finding is heartening as it suggests that the use of historical models circumvents the need for resources (not demonstrated here) to train new expert models. Moreover, the most effective historical model can achieve a level of accuracy that is on a par with expert models trained on the latest video data.

Algorithm 1 Retraining Scheduler Algorithm

Input: retraining tasks set $\mathcal{R}$, early stop time μ seconds, retrain window T seconds

for r in $\mathcal{R}$ **do**
  $gain[r] \leftarrow +\infty$
end for
while $T > \mu$ **do**
  $r \leftarrow \text{argmax} gain$
  $acc \leftarrow r.eval()$
  train request r for μ seconds
  $T \leftarrow T - \mu$
  $gain[r] \leftarrow r.eval() - acc$
end while

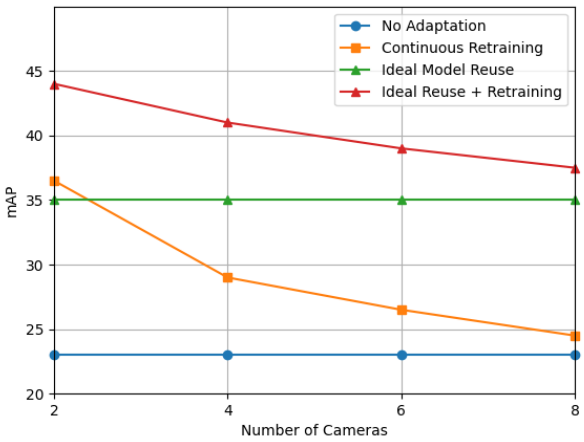

Fig. 2. The object detection accuracy of different designs under varying numbers of cameras. (mAP)

In addition, there exists an auspicious synergy between model reuse and continuous retraining, as evidenced by the Ideal Reuse with Retraining technique consistently achieving the highest mAP. The rationale behind this success is that the reused model offers a robust launchpad for retraining, thus decreasing the computational resources required by the retraining process (due to quicker convergence). Simultaneously, it heightens the precision of the evolved expert models' inference capabilities.

B. Benefits in accuracy consistency

An additional notable advantage of model reuse is the diminished necessity to await the completion of an expert model's retraining. This becomes critically important when a camera undergoes a dramatic scene shift and immediately requires a new model. For instance, if a car enters a tunnel, we

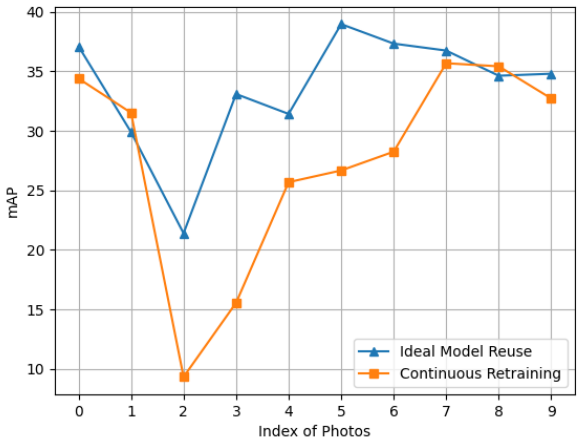

Fig. 3. The object detection accuracy of different designs under varying numbers of cameras. (mAP)

have the ability to rapidly select and alter the expert model, sidestepping the delay associated with training a new expert (as detailed in Fig. 3). For example, at the 1st percentile, Ideal Model Reuse sustains a 24% mAP, whereas Continuous Retraining dips to an unsatisfactory 7% mAP. Fig. 3 provides a tangible illustration of this scenario. As the vehicle enters the tunnel (id = 2), Ideal Model Reuse transitions to a suitable expert much more swiftly (id = 5) compared to Continuous Retraining (id = 7), resulting in a considerably lesser decline in model accuracy.

V. Discussion

To maximize the benefits of model reuse, several technical obstacles must be overcome. The ideal scenario for Model Reuse is to always select the optimal expert model without any computation cost or delay when browsing the entire model zoo, which is however, not feasible. Recent solutions for model reuse in the database sector, such as ODIN [10], have not yet addressed these issues, as they lack the design to efficiently share computation resources among model selection and retraining functions for numerous edge devices. In order to realize the potential of model reuse in a practical sense, a mechanism must be in place to swiftly and precisely find the best expert model. It's also essential to control the cost and latency of model selection to prevent them from escalating indefinitely with the proliferation of videos or cameras.

To conclude, leveraging historical expert models serves as a beneficial supplement to model retraining, and when utilized together, it promotes better resource efficiency and a more consistent, accurate model adaptation. Nevertheless, to transform model reuse into a practical strategy, numerous technical difficulties linger, which we will address in the subsequent section.

Acknowledgment

I am grateful to Ms. Nagasaki Soyo for bringing a touch of joy to my dreary college life.

References

[1] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodk, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. Computer, 50(10), 2017.

[2] Ion Stoica. The future of computing is distributed. https://www.datanami.com/2020/02/26/the-future-of-computing-is-distributed/, 2020.

[3] Shadi A. Noghabi, Landon P. Cox, Sharad Agarwal, and Ganesh Ananthanarayanan. The emerging landscape of edge computing. GetMobile Mob. Comput. Commun., 23(4), 2019.

[4] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. CoRR, abs/2107.08430, 2021.

[5] Azure linux virtual machine pricing. https: //azure.microsoft.com/en-us/pricing/ details/virtual-machines/linux/.

[6] Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[7] Mehrdad Khani, Pouya Hamadanian, Arash NasrEsfahany, and Mohammad Alizadeh. Real-time video inference on edge devices via adaptive model streaming. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[8] Romil Bhardwaj, Zhengxu Xia, Ganesh Ananthanarayanan, Yuanchao Shu, Nikolaos Karianakis, Kevin Hsieh, Paramvir Bahl, and Ion Stoica. Ekya: Continuous learning of video analytics models on edge compute servers. In USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2022.

[9] Samvit Jain, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, and Joseph Gonzalez. Scaling video analytics systems to large camera deployments. In Proceedings of the International Workshop on Mobile Computing Systems and Applications (HotMobile), 2019.

[10] Abhijit Suprem, Joy Arulraj, Calton Pu, and Joao Ferreira. Odin: Automated drift detection and recovery in video analytics. Proc. VLDB Endow., 13(12):24532465, jul 2020.

[11] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled Ben Letaief. A survey on mobile edge computing: The communication perspective. IEEE Commun. Surv. Tutorials, 19(4), 2017.

[12] Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2(7), 2015.

[13] 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 (CVPR), pages 36463654, 2017.

[14] Nada Ibrahim, Preeti Maurya, Omid Jafari, Parth Nagarkar. A Survey of Performance Optimization in Neural Network-Based Video Analytics Systems. arXiv preprint arXiv:2105.14195, 2021.

[15] A G Dobre1 and C Gheorghe. The Optimization of the Running Technique using Video Analysis Method. 2021 J. Phys.: Conf. Ser. 1746 012086

[16] M R Arun, Subramanian Selvakumar, M R Sheeba and F Shabina Fred Rishma. Latency Time and Resolution Optimization in Video Monitoring System. International Conference on Energy Efficient Technologies for Sustainability (ICEETS) 2018, April 6, 2018.

[17] Xian-Sheng Hua, Lie Lu and Hong-Jiang Zhang. Optimization-based automated home video editing system. In IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 5, pp. 572-583, May 2004.

[18] Mehrdad Khani, Ganesh Ananthanarayanan, Kevin Hsieh, Junchen Jiang, Ravi Netravali, Yuanchao Shu, Mohammad Alizadeh and Victor Bahl. RECL: Responsive Resource-Efficient Continuous Learning for Video Analytics. 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23), 2023.

[19] Liang, Ta Chen, and N. Balakrishnan. A Characterization of Exponential Distributions Through Conditional Independence. Journal of the Royal Statistical Society. Series B (Methodological), vol. 54, no. 1, 1992, pp. 26971. JSTOR, http://www.jstor.org/stable/2345962. Accessed 31 Mar. 2024.

[20] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538, 2017.

[21] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, Andr Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. Advances in Neural Information Processing Systems (NeurIPS), 34, 2021.

[22] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 33663375, 2017.