2022 9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS) | 979-8-3503-2045-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IOTSMS58070.2022.10062226

APT: Adaptive Perceptual quality based camera Tuning using reinforcement learning

Sibendu Paul†, Kunal Rao*, Giuseppe Coviello*, Murugan Sankaradas*, Oliver Po*,
Y. Charlie Hu‡ and Srimat Chakradhar*
*{sipaul, kunal, giuseppe.coviello, murugs, oliver, chak}@nec-labs.com
†‡{paul90, ychu}@purdue.edu
**NEC Laboratories America, Inc.* - Princeton, NJ, USA
†‡*Purdue University* - West Lafayette, PA, USA

***Abstract*—Cameras are increasingly being deployed in cities, enterprises and roads world-wide to enable many applications in public safety, intelligent transportation, retail, healthcare and manufacturing. Often, after initial deployment of the cameras, the environmental conditions and the scenes around these cameras change, and our experiments show that these changes can adversely impact the accuracy of insights from video analytics. This is because the camera parameter settings, though optimal at deployment time, are not the best settings for good-quality video capture as the environmental conditions and scenes around a camera change during operation. Capturing poor-quality video adversely affects the accuracy of analytics. To mitigate the loss in accuracy of insights, we propose a novel, reinforcement-learning based system** APT **that dynamically, and remotely (over 5G networks), tunes the camera parameters, to ensure a high-quality video capture, which mitigates any loss in accuracy of video analytics. As a result, such tuning restores the accuracy of insights when environmental conditions or scene content change.** APT **uses reinforcement learning, with no-reference perceptual quality estimation as the reward function. We conducted extensive real-world experiments, where we simultaneously deployed two cameras side-by-side overlooking an enterprise parking lot (one camera only has manufacturer-suggested default setting, while the other camera is dynamically tuned by** APT **during operation). Our experiments demonstrated that due to dynamic tuning by** APT**, the analytics insights are consistently better at all times of the day: the accuracy of object detection video analytics application was improved on average by $\sim$ 42%. Since our reward function is independent of any analytics task,** APT **can be readily used for different video analytics tasks.**

I. Introduction

The number of IoT sensors, especially video cameras deployed around the world have proliferated tremendously. It is estimated that their number will continue to grow further, thanks to advances in computer vision, machine learning, etc. and infrastructure support through 5G, edge computing, cloud computing, etc. These video cameras are being used for a variety of applications including video surveillance, intelligent transportation, healthcare, retail, entertainment, safety and security, and home and building automation. The global video surveillance camera market, which was valued at US $28.02 billion in 2021, is estimated to reach US $45.54 billion in 2027. Furthermore, the volume, which was 214.3 million units in 2021, is estimated to reach 524.75 million units in 2027 [27]. As the video camera market grows, the video analytics market also grows with it. The global video analytics market is estimated to grow from $5 billion in 2020 to $21 billion by 2027, at a CAGR of 22.70% [7].

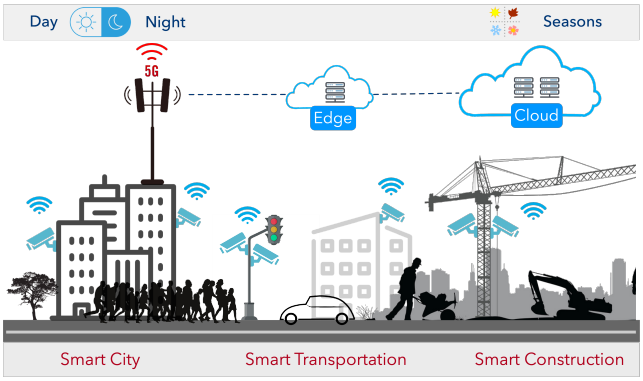


Fig. 1: City-scale video analytics.

A city-scale deployment of IoT cameras and video analytics being performed on those cameras is shown in Figure 1. Here, the video feed from cameras is streamed over 5G and the analytics is being performed in the edge/cloud infrastructure. Based on the results of the analytics, insights are generated and appropriate actions are taken. As the environmental conditions around the cameras (e.g. day or night, seasonal variations) and the scene in front of the camera (e.g. number of people/cars) change, the quality of video feed produced by the cameras also changes. This is due to the manner in which cameras capture, process, encode and transmit video frames before they are delivered to Analytics Units (AUs) in a Video Analytics Pipeline (VAP).

In this paper, we show that the accuracy of video analytics application is impacted by variation in environmental conditions or in the scene in front of the camera, and it may even degrade. One of the reasons for this degradation is the poor quality of frames being delivered to the AUs. Camera vendors often expose a large number of camera parameter settings to end users so that they can tune them according to their deployment location. These camera parameters play a significant role in the quality of frames being produced by the camera and delivered to the AUs. We show that if we adjust these camera

† Work mostly done as an intern at NEC Laboratories America, Inc.

parameter values, we can improve the quality of frames and thereby mitigate the loss in analytics accuracy due to changes in environment or video content. This adjustment in camera parameter values however is non-trivial. We observe that the desirable adjustment values vary according to the condition and is very specific to the deployment location. There is not any single adjustment setting that works across all conditions and across all deployment locations. Therefore, we need a dynamic adjustment technique that automatically adapts to the changing conditions at the specific deployment location.

To mitigate the loss in analytics accuracy, we propose to adaptively tune camera parameters in real time using reinforcement learning. In particular, we propose to develop a system that dynamically tunes four camera parameters, i.e., brightness, color, contrast and sharpness, which directly affect the quality of image produced by the video camera. Such dynamic tuning of camera parameters happens remotely over 5G and leads to better quality of the video feed, which directly helps in improving the analytics accuracy.

In designing reinforcement learning, for the agent to learn and adapt to the changes in conditions, we use perceptual no-reference quality estimator as the reward function. We show via experiments that such a reward function works well in adjusting camera settings so that the loss in analytics accuracy is mitigated. This technique is independent of the video analytics being performed, and therefore is easy to design and deploy in real-world settings.

There are different methods to calculate the perceptual quality, but testing and comparing them to check which one works the best for real-world setting is not straight forward. First, if we test these methods one by one, then it is impossible to repeat exact same environment and video content changes, if the video analytics system (VAS) is deployed in the wild. Such a setup will not lead to "apples to apples" comparison. Second, if we test these methods at the same time, then it is not practical to simultaneously deploy as many cameras as the number of methods to calculate perceptual quality for each method on each camera at the same time. These challenges lead us to consider a mock experimental setup, which allows us to repeat the environment and content changes in a controlled setting, and objectively test and compare different perceptual quality estimators one-by-one.

In summary, our key contributions are as follows:

- We empirically show that changes in environmental conditions and video content can have adverse effect on video analytics accuracy, and this loss in accuracy can be mitigated by dynamically tuning camera settings.
- We propose novel Reinforcement Learning (RL) based system called APT, which automatically and adaptively tunes camera parameters remotely over 5G, in order to produce good quality video feed, which directly helps in improving analytics insights
- We use CNN-based state-of-the-art perceptual quality estimator (*i.e., RankIQA*) as the reward function in RL, thus making APT design independent of the analytics being performed and feasible in absence of ground-truth.
- Our adaptive camera-parameter-tuning results in consistent analytics accuracy improvement through different time segments of the day and achieves an average improvement of $\sim 42\%$ when compared to the accuracy observed under fixed, manufacturer-provided default setting.

The rest of the paper is organized as follows. We discuss related works in Section II. Section III presents the negative impact of environmental condition and content changes on analytics accuracy and how it can be possibly mitigated by adaptively tuning built-in camera parameters. The design challenges of such adaptive camera-parameter-tuning system and final APT design are shown in Section IV and Section V, respectively. Extensive evaluation of APT on 3D mock-up scene as well as real-world deployment is discussed in Section VI. Finally, in Section VII we show how to possibly extend APT design in the future and conclude in Section VIII.

II. Related Works

Several recent proposals have investigated the tuning of parameters of vision algorithms to improve computing resource usage of video analytics pipelines based on input video content. Chameleon [11], Videostorm [36], and AWStream [35] tune the after-capture video stream parameters like frame sampling rate, frame resolution, type of detector to ensure efficient resource usage while processing video analytics queries at scale. However, they do not address directly tuning of camera parameters to enhance video analytics accuracy.

A recent work [10] also reports the impact of environmental condition changes on video analytics accuracy but it adapts to such changes by using different AUs depending on specific environmental condition, while keeping the camera settings the same. Since environmental changes can take place due to change of the sun's movement throughout the day, different weather conditions (*e.g.,* rain, fog and snow), as well as for different deployment sites (*e.g.,* parking lot, shopping mall, airport), it is infeasible to develop a separate AU specific to each environment. Impact of environmental changes on AU accuracy is also shown in [25], but they address it by re-training the AU using transfer learning. In contrast, APT takes a different approach where the AU is kept the same, but camera settings are dynamically tuned in reaction to changes in environmental conditions.

Several recent works like AMS [14] and Ekya [3] aim to improve video analytics accuracy by periodically re-training AI/ML models so that they work well for the specific deployment conditions. This technique however, requires additional computational resources and it does not quickly adapt to the changes in the environment or video content. APT, on the other hand does not rely on continuous re-training, rather it improves video analytics accuracy by dynamically tuning configurable camera parameters, thereby quickly reacting to the changes in environmental conditions or video content.

There is a considerable body of work to configure image signal processing pipeline (ISP) in cameras to improve camera capture quality. For example, VisionISP [33] modifies the ISP pipeline to reduce the size of final image output by reducing

the bit-depth and resolution. Others have proposed custom optimizations of the ISP for specific computer-vision tasks [6], [9], [16], [21], [28], [33], [37]. However, careful re-design or optimization of ISP module for specific vision tasks is time consuming. In our proposed approach APT, we do not modify the ISP pipeline, rather we focus on dynamic tuning of configurable camera parameters to consistently produce high-quality video output, which enhances the quality of insights from analytics tasks.

III. Motivation

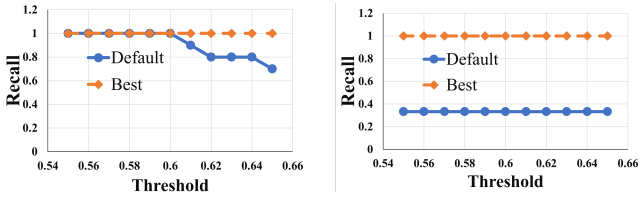
In this section, we show that environmental conditions and video content variation can adversely impact analytics accuracy and this loss in accuracy can be mitigated by adjusting camera parameter values. To illustrate the impact of environment and content variation on AU accuracy, we consider four popular parameters that are exposed by almost all cameras: *brightness, contrast, color-saturation* (also known as colorfulness), and *sharpness*. We focused on these four parameters because they are widely available in both PTZ and non-PTZ cameras and these parameters are more challenging to tune due to their large range of parameter values (for example between 1 and 100).

Methodology: Analyzing the impact of camera settings on video analytics poses a significant challenge: it requires applying different camera parameter settings to the same input scene and measuring the difference in the resulting accuracy of insights from an AU. The straight-forward approach is to use multiple cameras with different camera parameter settings to record the same input scene. However, such an approach is impractical as there are thousands of different combinations of even just the four camera parameters we consider. To overcome the challenge, we proceed with two workarounds. First, we will show the impact of camera settings on a stationary scene with a real camera. Second, we apply post-capture image transformation on pre-recorded video snippets from public datasets to analyze the equivalent impact of different camera settings on those video snippets, *i.e.,* groups of frames.

A. Impact of environment variation on AU accuracy

To study the impact of environmental changes on AU performance, we simulate DAY and NIGHT conditions in our lab and evaluate the performance of the most accurate face-recognition AU (Neoface-v3 [23][1]. We use two sources of light and keep one of them always ON, while the other light is manually turned ON or OFF to emulate DAY and NIGHT conditions, respectively.

We place face cutouts of 12 unique individuals in front of the camera and first run the face recognition pipeline with the input scene captured under "Default" camera setting (*i.e.,* the default values provided by the manufacturer) and also for different face matching thresholds. Since this face-recognition AU has high precision despite environment changes, we focus on measuring Recall, *i.e.,* true-positive rate. Figure 2a shows the Recall for the DAY condition for various thresholds and Figure 2b shows the Recall for the NIGHT condition for various thresholds. We see that under the "Default" settings, the Recall for the DAY condition goes down at higher thresholds, indicating that some faces were not recognized, whereas for the NIGHT condition, the Recall remains constant at a low value for all thresholds, indicating that some faces were not being recognized regardless of the face matching thresholds. Thus, the performance of face-recognition AU (*i.e.,* recall vs matching threshold) under "Default" camera setting varies for different environment while capturing the same static scene. Next, we compare AU results under the "Default" camera settings, and "Best" settings for the four camera parameters. To find the "Best" settings, we change the four camera parameters using the VAPIX API [4] provided by the camera vendor to find the setting that gives the highest Recall value. Specifically, we vary each parameter from 0 to 100 in steps of 10 and capture the frame for each camera setting. This gives us ≈14.6K (11^4) frames for each condition. Changing one camera setting through the VAPIX API takes about 200ms, and in total it took about *7 hours* to capture and process the frames for each condition.



(a) DAY (b) NIGHT

Fig. 2: Parameter tuning impact for Face-recognition AU.

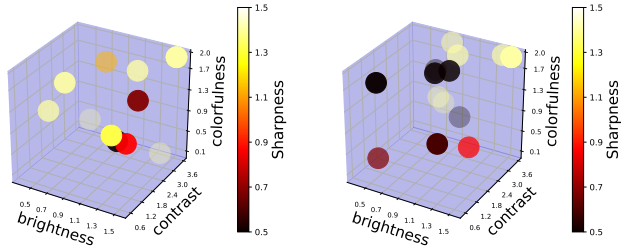
In contrast, when we changed the camera parameters for both conditions to the "Best" settings, the AU achieves the highest Recall (100%), confirming that all the faces are correctly recognized, also shown in Figure 2. These results show that it is indeed possible to improve AU accuracy by adjusting the four camera parameters.

B. Impact of video content variation on AU accuracy

We study the impact of video content variation on AU accuracy by using pre-recorded videos with different video content. The pre-recorded videos from public datasets are already captured under certain camera parameter settings, and hence we do not have the opportunity to change the real camera parameters and observe their impact. As an approximation, we apply different values of brightness, contrast, color-saturation and sharpness to these pre-recorded videos using several image transformation algorithms in the Python Imaging Library (PIL) [1], and then observe the impact of such transformation on accuracy of AU insights.

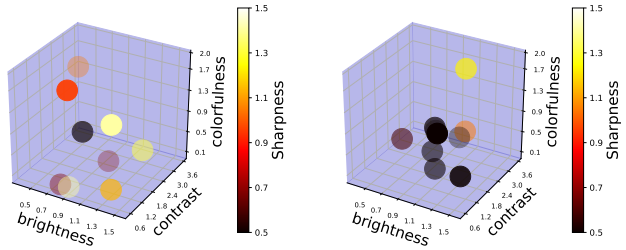
We consider 19 video snippets from the HMDB dataset [15] and 11 video snippets from the Olympics dataset [20] that capture different content under different environmental conditions while using the default camera parameter. Using cvat-tool [22], we manually annotated the face and person bounding boxes to form our ground truth. Each video-snippet contains

[1]This face-recognition AU is ranked first in the world in the most recent face-recognition technology benchmarking by NIST.



(a) Face detection (b) Person detection

Fig. 3: Distribution of best *transformation tuple* for two AUs on *HMDB* video snippets.



(a) Face detection (b) Person detection

Fig. 4: Distribution of best *transformation tuple* for two AUs on *Olympics* video snippets.

no more than a few hundred frames, and the environmental conditions vary across the video snippets due to change in video capture locations. We determine a single best tuple of those four transformations for each video, *i.e.,* one that results in the highest analytical quality for that video. Figure 3 and Figure 4 show the distribution of the best transformation tuples for the videos in the two datasets, respectively. We see that with a few exceptions, the best transformation tuples for different videos (*i.e.,* that capture different content under various environmental condition) in a dataset do not cluster, suggesting that any fixed real camera parameter settings will not be ideal for different environmental conditions or input content as well as it also varies for different analytics tasks. Table I shows the maximum and average analytical quality improvement achieved after transforming each video-snippet as per their best transformation tuple. We observe up to 58% improvement in accuracy of insights when appropriate transformations or equivalent camera parameters are applied.

In summary, environmental changes and input content variations can result in low-quality image acquisition, which in turn result in poor analytics accuracy. Tuning the camera parameter settings during capture can provide improvement in accuracy of AUs, but such camera parameter tuning is hard for a human to do manually because the best parameter combination will vary with location of the camera, the type of analytics units, and the environmental conditions. This calls for developing methods that can automatically adapt the camera parameters to improve the accuracy of AUs.

TABLE I: Accuracy improvement of best configuration.

Video-Dataset	AU	mAP improvement	
		Max	Mean
Olympics	Person Detection	40.38	8.38
	Face Detection	19.23	1.68
HMDB	Person Detection	57.59	12.63
	Face Detection	18.75	4.22

IV. Challenges

In this paper, we propose to develop a camera tuning framework that dynamically adapts the four parameter settings of the video-capturing camera in a video analytics system (VAS) to optimizes the accuracy of its AUs. Designing such a framework faces two challenges. Below, we discuss these two challenges and our approaches to addressing each one of them.

Challenge 1: Identifying the best camera settings for a particular scene. Identifying the best camera settings for a given scene that gives the best AU accuracy is challenging even during offline, as it requires comparing the impact of all possible camera settings. Doing so in an online manner is even more challenging.

Approach. To address this challenge, we propose to use an online learning method. Particularly, we use Reinforcement Learning (RL) [30], in which the agent learns the best camera settings on-the-go. Using RL, we do not have to know apriori the various scenes that the camera would observe. Instead, the RL agent learns and identifies automatically the best camera settings that give the highest AU accuracy for any particular scene. Out of several recent RL algorithms, we choose the SARSA [32] RL algorithm for identifying the best camera settings.

While RL is a fairly standard technique, applying it to tuning camera parameters in a real-time video analytics system in turn raises one unique challenge as follows.

Challenge 2: No Ground truth in real time. Implementing the online RL approach requires knowing the quality (*i.e.,* either reward or penalty) of every action taken during exploration and exploitation. Measuring the quality of camera parameters' change in absence of ground-truth is challenging.

Approach. We proposed to leverage state-of-the-art perceptual Image Quality Assessment (IQA) methods as a proxy of the quality measure (*i.e.,* the reward function). Specifically, we experimentally evaluate a list of state-of-the-art IQA methods and use the best-performing one as the reward function in the RL engine.

V. Design

Figure 5 shows the system-level architecture for APT, which automatically and adaptively tunes the camera parameters to optimize the analytics accuracy. APT incorporates two key components: a perceptual no-reference quality estimator and a Reinforcement Learning (RL) engine.

A. Perceptual No-reference Quality Estimator

Since it is not possible to obtain ground-truth in real-time to measure the accuracy of video analytics applications, we rely on a technique which won't require ground truth, but still help in improving the analytics accuracy. To this end, we leverage

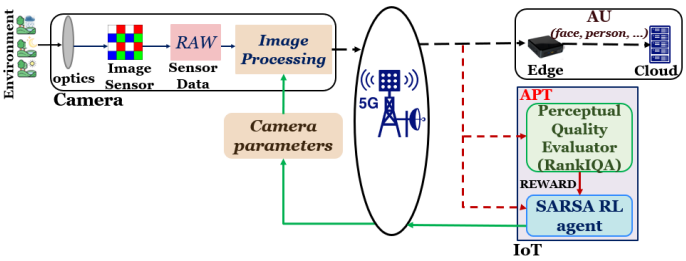


Fig. 5: *APT* system design.

SOTA perceptual no-reference quality estimator, which gives an estimate of the quality of the frame produced by the video camera. Better the quality of the frame, better will the analytics be able to generate accurate insights. Therefore, in the design on APT we employ such IQA method to help guide the system in choosing appropriate camera settings. We discuss our choice of IQA method in Section VI-A and how this IQA method is used in RL engine within APT in Section V-B.

B. Reinforcement Learning (RL) Engine

RL engine is the heart of APT, as it is the one that automatically chooses the best camera settings for a particular scene. In designing the RL engine, we considered popular RL algorithms such as Q-learning [31] and SARSA [32] which are general techniques and highly effective in learning the best action to take in order to maximize the reward. To choose between the two options, we experimentally compared them in the context of choosing the best camera settings and found that training with SARSA achieves slightly faster convergence than with Q-learning. Therefore, we decide to use the SARSA RL algorithm in APT.

Like other RL algorithms, in SARSA, an agent continuously interacts with the environment (*state*) it is operating in, by taking different *actions*. As the agent takes an action, it moves into a new state or environment. For each action, there is an associated *reward* or penalty, depending on whether the new state is more desirable or not. Over time, as the agent continues taking actions and receiving rewards and penalties, it learns to maximize the rewards by taking the right actions, which ultimately lead the agent towards desirable states.

As with many other RL algorithms, SARSA does not require any labeled data or pre-trained model, but it does require a clear definition of the *state*, *action* and *reward* for the RL agent. This combination of *state*, *action* and *reward* is unique for each application and therefore needs to be carefully chosen, which ensures that the agent learns exactly what is desired. In our setup, we define them as follows:

State: A state is a tuple of two vectors, $s_t =< P_t, M_t >$, where P_t consists of the current brightness, contrast, sharpness, and color-saturation parameter values on the camera, and M_t consists of the measured values of brightness, contrast, color-saturation, and sharpness of the captured frame at time t, measured as in [2], [5], [8], [26].

Action: The set of actions that the agent can take are (1) to increase or decrease one of the brightness, contrast, sharpness or color-saturation parameter value, or (2) not to change any parameter values.

Reward: We use the best-performing (experimentally chosen as described in Section VI-A) perceptual quality estimator as the immediate reward function (r) for the SARSA algorithm. Along with considering immediate reward, the agent also factors in future reward that may accrue as a result of the current actions. Based on this, a value, termed as Q-value (also denoted as $Q(s_t, a_t)$) is calculated for taking an action a_t when in state s_t using Equation 1.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \quad (1)$$

Here, α is learning rate (a constant between 0 and 1) used to control how much importance is to be given to new information obtained by the agent. A value of 1 will give high importance to the new information while a value of 0 will stop the learning phase for the agent.

Similar to α, γ (also known as the discount factor) is another constant used to control the importance given by the agent to any long term rewards. A value of 1 will give very high importance to long term rewards while a value of 0 will make the agent ignore any long term rewards and focus only on the immediate rewards. If the conditions do not change frequently, a higher value, *e.g.,* 0.9, can be assigned to prioritize long term rewards; if the environmental conditions change very frequently, a lower value, *e.g.,* 0.1, can be assigned to γ to prioritize immediate rewards.

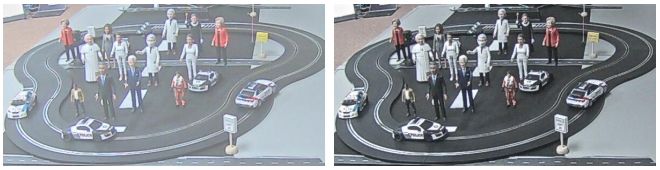
Exploration vs. Exploitation. We define a constant called ϵ (between 0 and 1) to control the balance between exploration vs. exploitation when the agent takes actions. In particular, at each step, the agent generates a random number between 0 and 1; if the random number is greater than the set value of ϵ, then a random action (exploration) is chosen, else it performs exploitation.

VI. Evaluation

We first evaluate several design choices for the perceptual IQA method to be used as the reward function in the RL engine in terms of their impact on analytics performance in a controlled mock-up scene (Section VI-A), and pick the best-performing choice for use in APT. We then extensively evaluate the effectiveness of APT on the mock-up scene under different initial parameter settings (Section VI-B) and in a real-world deployment (Section VI-C).

A. Effect of using different IQA Methods

Throughout the last decade, several no-reference (blind) IQA methods [13], [17]–[19], [29], [34] have been proposed to improve the video/image capture quality based on human perception. In this section, we evaluate the impact of three different blind IQA methods that are designed to estimate the quality of real-world distorted images for use as APT's quality evaluator. Since downstream analytics focus on low-level local features (*i.e.,* extracted via convolution layers) for deriving insights from the input video stream, we choose three popular perceptual IQA methods that employ convolution network.



(a) capture under fixed setting (b) APT camera capture

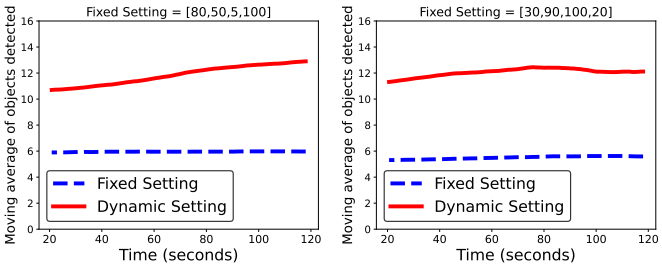
Fig. 6: Sample camera captures.

CNN-IQA [13] is the first to use the spatial domain without relying on hand-craft features used by previous IQA methods. It combines feature learning and quality regression in one optimization process which leads to a more effective quality estimation model. Hyper-IQA [29] decouples the IQA procedure into three stages: content understanding, perception rule learning and finally quality prediction. Hyper-IQA estimates image quality in a self-adaptive manner by adaptively running different hyper-networks. Finally, Rank-IQA [17] addresses the problem of limited size of the IQA dataset during training. It uses a siamese network to rank images and then uses the ranked images to train deeper and wider convolution networks for absolute quality prediction.

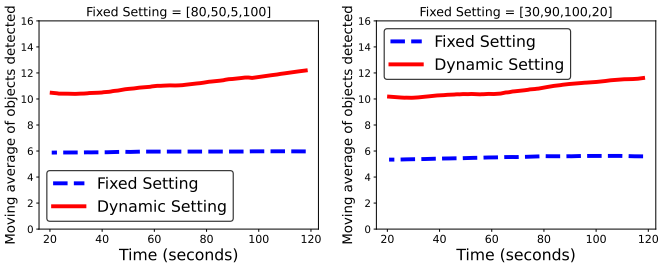
To assess the impact of using different IQA estimators as reward functions on the analytics performance under the same environmental conditions, we use a mock-up scene with a fixed number of objects (*i.e.,* cars and persons). In this mock-up scene, 3D slot cars are continuously moving along the track and 3D human models are kept stationary. In doing so, this experimentally controlled mock-up scene provides controllability and replicability in experimental setup and enables us to try out different reward functions under the same environment and content.

We first train APT using each of these three quality estimator output as the reward function for one hour on the mock-up scene. During training, after every 2 minute interval, we change the camera parameters to emulate different environmental conditions. For evaluation, we placed two identical *AXIS 3505 MK-II network cameras* side-by-side in front of the mock-up scene as shown in Figure 6. During evaluation, we used 5 different camera settings and observed how APT reacts to those initial camera parameter settings.

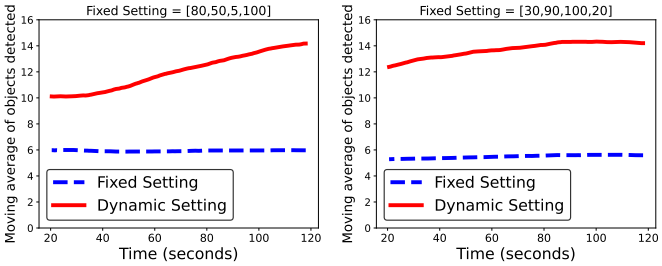
Figure 7 shows how the three different IQA methods effectively guide SARSA RL agent in APT, resulting in higher true-positive object detections when compared to object detector's performance on the stream with fixed camera setting. Table II presents the average improvement in true-positive object detections observed throughout multiple 2-minute time-intervals, and the average number of objects detected in the steady state for the three different reward functions. We observe that *Rank-IQA* guides SARSA-RL agent better under environmental variations which in turn leads to more object detections from the same scene. Thus, we use Rank-IQA as our perceptual quality estimator for APT.



(a) HyperIQA as reward



(b) CNNIQA as reward



(c) RankIQA as reward

Fig. 7: APT reaction to different initial camera settings under different IQA metrics as reward function (Moving average of per-frame object detection is computed over a window of last 100 frames, shown in Y axis.)

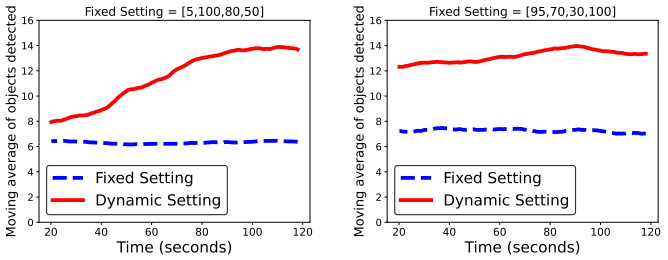
TABLE II: Comparing IQA methods as reward functions

IQA	Improv. over fixed settings (Avg) %	Objs detected steady state (Avg)
Hyper-IQA	132.6	13.5
CNN-IQA	141.2	14.2
Rank-IQA	150.5	15.1

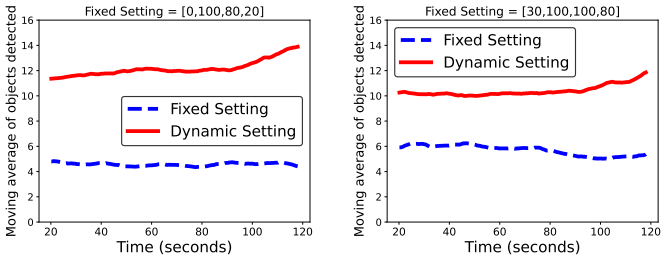
B. *Effectiveness of* APT *in a Mock-up Scene*

Here, we evaluate how quickly APT can react to any initial setting and converge to a setting that can provide better analytical outcome. We use the same controlled mock-up scene described in Section VI-A. Both cameras start with same initial setting (we use four different camera settings denoted as S1, S2, S3 and S4, respectively) and stream at 10 FPS over a 2-minute period, during which the four parameters of Camera 1 are kept to the same initial values, while the parameters of Camera 2 are tuned dynamically by APT every 2 seconds. On every frame streamed from the camera, we use Yolov5 [12] object detector to detect objects and record the type of objects with their bounding boxes [2]. Figure 8 shows the moving average

[2]Manual inspection confirms there is no false-positive detection in the 2-minute period.



(a) Fixed setting 1 (S1) (b) Fixed setting 2 (S2)



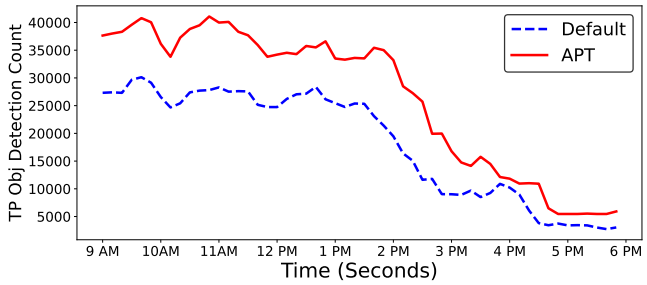
(c) Fixed setting 3 (S3) (d) Fixed setting 4 (S4)

Fig. 8: APT reaction to different camera settings using Rank-IQA reward (moving average of per-frame object detection is computed over a window of past 100 frames, shown in the Y-axis.)

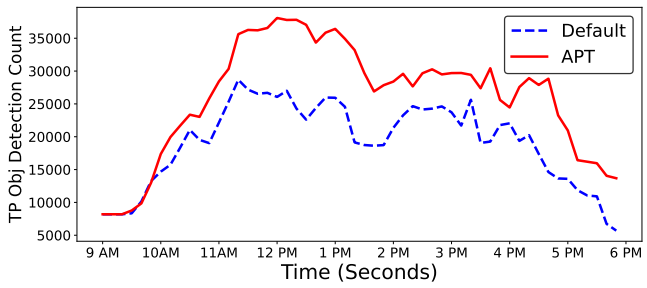
of per-frame object detections (with a window size of 100 frames) in the Y axis (to clearly show the trend) of the two cameras under four different initial settings. We observe there is an initial gap between the performance of YOLOv5 between the two camera streams which indicates that within the first 10 seconds, APT changes the camera parameters based on human-perceptual quality estimator (*i.e.,*RANKIQA) output and achieves better object detection. Furthermore, we observe that APT gradually finds best-possible setting within one minute that enables Yolov5 to detect more number of objects from the captured scene (total 4-9 more object detections per frame compared to detections on camera stream with fixed setting).

C. *Effectiveness of* APT *in Real-world Deployment*

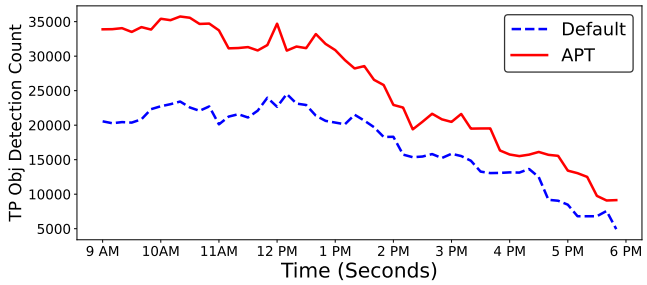
To evaluate the effectiveness of camera parameter tuning by APT in a real-world deployment, we use two co-located *AXIS Q3515* network cameras that continuously monitor an enterprise parking lot. Here, one camera is set to manufacturer-provided default settings, while the other camera parameters are adaptively tuned by APT. Captured frames from each camera are uploaded to a remote edge-server (equipped with Intel-Xeon processor and NVIDIA Geforce GTX-2080 GPU) running Yolo-v5 [12] object detector. The captured frames from both cameras are sent for AU processing on the edge-server over a *5G* network with an average frame uploading latency of *39.7* ms. While the first camera stream is sent just to the object detector AU, the second camera stream is also sent to APT which runs on a low-end IoT device (Intel NUC box with a 2.6 GHz Intel i7-6770HQ CPU) in parallel to object detector AU. We first perform in-situ training using the second camera stream to populate the Q-table for SARSA RL agent for 12 hours then we observe the performance of APT for



(a) Day 1



(b) Day 2



(c) Day 3

Fig. 9: APT performance (with Rank-IQA reward function) throughout the day in Parking lot (true-positive object detections are accumulated for each 10-minute interval).

next consecutive days in the exploitation phase. In this setup, APT adjusts the camera parameters every 30 seconds.

We ran both video analytics pipeline (VAP) for 9 continuous peak hours in each day, i.e., during daylight. We also recorded the videos captured by the cameras and the detections on those camera stream to manually inspect and validate the detections from both VAPs. Figure 9 shows the total true-positive object detections (*i.e.,* car and person) in each 10-minute interval from 9AM-6PM for three consecutive days. APT constantly provides higher true-positive detection count than the default camera stream during all segments of the day, as shown in Figure 9. We also observe an improvement of *44.71%*, *35.49%* and *45.92%* (average $\sim$ 42 %) more true-positive object detections from the camera stream tuned by APT when compared to the default camera stream for first three days of evaluation, respectively. Thus, APT is effective in adaptively tuning camera parameters such that the video quality is improved, thereby resulting in improvement in video analytics accuracy.

VII. Future Works

Our demonstration in this paper that adaptive camera parameter tuning can improve video analytics accuracy opens up several new research avenues. Here, we designed APT, which adaptively tunes four image-appearance parameters based on the perceptual quality metric. In future work, we plan to study how other non-automated image and video-specific camera parameters such as max shutter speed, maximum gain, compression, bitrate, FPS, *etc.* can be adpatively tuned to enhance video analytics accuracy. In addition to video surveillance camera sensor, we also plan to extend APT design to tune the parameters of other complex sensors such as depth and thermal cameras.

For APT design, we borrowed the quality estimator from SOTA perceptual no-reference IQA (*i.e.,* Rank-IQA). Since the captured content are consumed by downstream AUs, in future work we plan to explore the scope of quality estimation based on the downstream AU's perception, similar to the work presented in [24].

VIII. Conclusion

Video analytics applications heavily rely on good quality of video input to produce accurate analytics results. In this paper, we show that variation in environmental conditions and video content can lead to degradation in input video quality, leading to degradation of overall analytics insights. To mitigate this loss in accuracy, we propose APT, which uses reinforcement learning techniques to adaptively tune camera parameters so as to improve video quality, thereby improving accuracy of analytics. Through real-world experiments, we show that APT consistently performs better than the fixed manufacturer-provided default camera settings, and on average improves the accuracy of object detection video analytics application by $\sim 42\%$.

Acknowledgment. This project is supported in part by NEC Labs America and by NSF grant 2211459-CNS.

References

[1] Pillow library. https://pillow.readthedocs.io/en/stable/.

[2] S. Bezryadin, P. Bourov, and D. Ilinih. Brightness calculation in digital image processing. In *International symposium on technologies for digital photo fulfillment*, volume 2007, pages 10–15. Society for Imaging Science and Technology, 2007.

[3] R. Bhardwaj, Z. Xia, G. Ananthanarayanan, J. Jiang, Y. Shu, N. Karianakis, K. Hsieh, P. Bahl, and I. Stoica. Ekya: Continuous learning of video analytics models on edge compute servers. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 119–135, Renton, WA, Apr. 2022. USENIX Association.

[4] A. Communications. Vapix library.

[5] K. De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013.

[6] S. Diamond, V. Sitzmann, F. Julca-Aguilar, S. Boyd, G. Wetzstein, and F. Heide. Dirty pixels: Towards end-to-end image processing and perception. *ACM Transactions on Graphics (TOG)*, 40(3):1–15, 2021.

[7] V. Gaikwad and R. Rake. Video Analytics Market Statistics: 2027, 2021.

[8] D. Hasler and S. E. Suesstrunk. Measuring colorfulness in natural images. In *Human vision and electronic imaging VIII*, volume 5007, pages 87–95. International Society for Optics and Photonics, 2003.

[9] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM TOG*, 33(6):1–13, 2014.

[10] S. Y. Jang, Y. Lee, B. Shin, and D. Lee. Application-aware IoT camera virtualization for video analytics edge computing. In *IEEE/ACM SEC*, pages 132–144. IEEE, 2018.

[11] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica. Chameleon: scalable adaptation of video analytics. In *Proc. of ACM SIGCOMM*, pages 253–266, 2018.

[12] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, Feb. 2022.

[13] L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE CVPR*, pages 1733–1740, 2014.

[14] M. Khani, P. Hamadanian, A. Nasr-Esfahany, and M. Alizadeh. Real-time video inference on edge devices via adaptive model streaming. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4552–4562, 2021.

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of ICCV*, 2011.

[16] L. Liu, X. Jia, J. Liu, and Q. Tian. Joint demosaicing and denoising with self guidance. In *Proceedings of the IEEE/CVF CVPR*, pages 2240–2249, 2020.

[17] X. Liu, J. Van De Weijer, and A. D. Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. In *Proceedings of the IEEE ICCV*, pages 1040–1049, 2017.

[18] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

[19] A. K. Moorthy and A. C. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal processing letters*, 17(5):513–516, 2010.

[20] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, pages 392–405. Springer, 2010.

[21] J. Nishimura, T. Gerasimow, R. Sushma, A. Sutic, C.-T. Wu, and G. Michael. Automatic isp image quality tuning using nonlinear optimization. In *Proc. of IEEE ICIP*, pages 2471–2475. IEEE, 2018.

[22] openvinotoolkit. Computer vision annotation tool (cvat). https://github.com/openvinotoolkit/cvat.

[23] M. N. Patrick Grother and K. Hanaoka. Face Recognition Vendor Test (FRVT). https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8271.pdf, 2019.

[24] S. Paul, U. Drolia, Y. C. Hu, and S. T. Chakradhar. Aqua: Analytical quality assessment for optimizing video analytics systems. In *IEEE/ACM SEC*, pages 135–147. IEEE, 2021.

[25] S. Paul, K. Rao, G. Coviello, M. Sankaradas, O. Po, Y. C. Hu, and S. Chakradhar. Why is the video analytics accuracy fluctuating, and what can we do about it? *arXiv preprint arXiv:2208.12644v2*, 2022.

[26] E. Peli. Contrast in complex images. *JOSA A*, 7(10):2032–2040, 1990.

[27] D. Research. Global Surveillance Camera Market: Analysis By System Type, By Technology By Region Size and Trends with Impact of COVID-19 and Forecast up to 2027, 2022.

[28] E. Schwartz, R. Giryes, and A. M. Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2018.

[29] S. Su, Q. Yan, Y. Zhu, C. Zhang, X. Ge, J. Sun, and Y. Zhang. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *Proceedings of the IEEE/CVF CVPR*, June 2020.

[30] R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[31] C. J. C. H. Watkins and P. Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.

[32] M. Wiering and J. Schmidhuber. Fast Online q(λ). *Machine Learning*, 33(1):105–115, Oct 1998.

[33] C.-T. Wu, L. F. Isikdogan, S. Rao, B. Nayak, T. Gerasimow, A. Sutic, L. Ain-kedem, and G. Michael. Visionisp: Repurposing the image signal processor for computer vision applications. In *IEEE ICIP*, pages 4624–4628. IEEE, 2019.

[34] W. Xue, X. Mou, L. Zhang, A. C. Bovik, and X. Feng. Blind image quality assessment using joint statistics of gradient magnitude and laplacian

features. *IEEE Transactions on Image Processing*, 23(11):4850–4862, 2014.

[35] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee. Awstream: Adaptive wide-area streaming analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 236–252, 2018.

[36] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman. Live video analytics at scale with approximation and delay-tolerance. In *Proc. of 14th USENIX NSDI)*, pages 377–392, Boston, MA, Mar. 2017. USENIX Association.

[37] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.