Volatile MAB-based Configuration Selection for Offloading Video Analytics Tasks to Edges

Yu Liang[†*], Sheng Zhang[‡], Jie Wu[§]
[†]School of Computer and Electronic Information and the School of Artificial Intelligence, Nanjing Normal University
[‡]State Key Laboratory for Novel Software Technology, Nanjing University
[§]Center for Networked Computing, Temple University, US
[*]Corresponding author: liangyu@njnu.edu.cn

***Abstract*—The demand for video analytics is increasing rapidly. Due to the limited computational and network resources on edge servers, adjusting video configurations such as resolution and frame rate has become an effective strategy to reduce computational and transmission costs. However, this can also compromise detection accuracy, necessitating a balance between resource consumption and analytics accuracy. Also, the dynamic availability of edge servers and variability in their energy consumption further complicates making offloading decisions and configuration selection. In this paper, we first model the problem as a mixed planning program. Then we propose a volatile MAB-based configuration selection algorithm, VACS, which aims to maximize video analytics accuracy while reducing the overall energy consumption. Rigorous proof measures the gap between online decisions and the optimum. Extensive experiments validate the effectiveness of VACS.**

***Index Terms*—Video analytics, edge intelligence, task offloading, volatile multi-arm bandit**

I. Introduction

In recent years, the deployment of cameras for various purposes such as traffic control, crime prevention, and artificial intelligence has led to the generation of vast amounts of video data, often requiring rapid and accurate analytics. Typically, video analytics applications demand substantial computing resources and result in high energy consumption. However, the local computing resources at cameras may be insufficient to meet high-performance analytics requirements, necessitating the offloading of video data to edge servers with richer computing resources. Nevertheless, the computing and network resources at edge nodes are often limited. Therefore, adjusting video transmission configuration has become an effective approach. By adjusting configurations such as frame rate and resolution, it is possible to reduce computation and transmission costs at edges. However, this approach also faces the challenge of balancing between resource consumption and analytics accuracy. The main challenges are as follows:

Firstly, edge servers with heterogeneous hardware may result in only specific video analytics models being supported [3, 23]. Since analytics models typically have their own input formats [1, 4, 27], the supported video configurations for different models are usually limited [5–7]. Therefore, when making video task offloading decisions, it is necessary to match the video configuration for transmission and the configuration supported by edge servers. Furthermore, when the candidate server set is unstable, offloading decisions and configurations should be also dynamically adjusted over time, making it more challenging.

Secondly, more expensive transmission configurations (e.g., higher resolution or frame rate) lead to high accuracy analytics results as well as high computing and transmission energy costs. Therefore, a trade-off between accuracy and energy consumption must be made. Since transmission configuration is part of the input to the video analytics model, this trade-off decision must be made before offloading. However, in practical applications, due to the highly dynamic nature of video content over time, a CNN model with a fixed configuration may yield different analytics accuracies. Therefore, dynamically adjusting transmission configurations over time while minimizing overall energy consumption and maximizing analytics accuracy becomes a critical issue.

Thirdly, the energy consumption for transmission and computation varies dynamically over time [9, 20], making it difficult to estimate energy consumption in advance. Additionally, edge servers may experience energy depletion or movement, leading to uncertainty in the candidate server set, further increasing the difficulty of estimating server performance and balancing the trade-offs between accuracy and energy cost.

Prior research [4, 14, 15, 22, 24, 25, 27–30] optimized video analytics pipelines from different perspectives, e.g., server-driven, offline + online, parallel decoding, and super resolution. However, most of them are based on deterministic and known information of edge servers and do not take into account the dynamic candidate server set.

This work considers the differences and variations among different edge servers in terms of communication costs, computing performance, and energy expenditure. Additionally, attention is paid to the dynamic changes in the candidate server set due to edge servers' mobility or energy depletion.

In this paper, we propose a volatile MAB-based configuration selection algorithm, VACS. It utilizes the volatile multi-arm bandit framework to capture the variations in the availability and performance of edge servers, predicting the utility rewards achievable by offloading to these servers and adjusting configurations to make task offloading decisions adaptively. Rigorous proof measures the gap between online decision-making and the optimum. Extensive experiments validate the effectiveness of VACS.

II. System Model

We consider a set of time epochs $\mathcal{T} = \{1, .., t, .., T\}$, which are further divided into multiple time slots $\mathcal{J}_t$. Within each time slot, devices need to make task offloading decisions for the collected video data, i.e., selecting an edge server and a configuration. Fig. 1 illustrates the problem scenario in which multiple edge servers are located near the device, forming a candidate server set. However, edge servers may become unavailable due to mobility or energy depletion, resulting in possible changes in the available servers within each time slot. We assume that the candidate server set remains stable within each epoch j, and $\mathcal{S}_{j,t}$ is used to represent the candidate server set in slot t within j-th epoch.

Let $\mathcal{M} = \{m_t | t = 1, 2, ..., T\}$ denote the set of all video analytics tasks, where m_t represents the task at slot t. Previous studies [10–13] show that video analytics task is relatively large, thus it can be further divided into multiple subtasks (e.g., each subtask contains several video frames). We divide m_t into K_t subtasks, denoted by $\mathcal{K}_t = \{m_{k,t} | k = 1, 2, ..., K_t\}$, where $m_{k,t}$ is the k-th subtask in task m_t. Different CNN models may be deployed on edge servers. CNN models typically have their own input formats, limiting the supported video configurations for different models, such as processing only a fixed range of input resolution. Therefore, $\mathcal{R}_s$ is used to identify the input resolution set supported by server s, and the video transmission resolution $r_{k,s,t}$ of subtask $m_{k,t}$ is chosen from $\mathcal{R}_s$. We introduce a decision variable $x_{k,s,t}$ to indicate whether subtask $m_{k,t}$ is offloaded to edge server s. If $x_{k,s,t} = 1$, it indicates that edge server s is selected; otherwise $x_{k,s,t} = 0$.

Detection Accuracy Model. Different configuration attributes affect the accuracy of video analytics in distinct ways, making it challenging to characterize the relationship between video transmission configurations and accuracy. Existing research [4, 14, 15] has demonstrated through extensive experiments that the impact of frame rate and resolution on accuracy is independent, and the relationship between accuracy and frame rate/resolution can be represented by concave functions. Based on these observations, the analytics accuracy $a_{k,t}$ of sub-task $m_{k,t}$ can be expressed as:

$$a_{k,t} = \epsilon_t(\sum_{s=1}^{|\mathcal{S}_{j,t}|} x_{k,s,t} r_{k,s,t}) \phi_t(f_{k,t}), \tag{1}$$

where the concave functions $\epsilon_t(r)$ and $\phi_t(f)$ represent the accuracy with respect to resolution r and frame rate f in time slot t, respectively.

Energy Consumption Model. Given the limited battery life of edge devices, energy consumption is a crucial consideration when designing configuration selection algorithms. The energy consumption of edge devices primarily includes the transmission energy used for transmitting video data and the processing energy consumed by running CNN models on servers.

The transmission energy is usually proportional to the amount of data transmitted [4]. According to research [16], a video frame with resolution r contains αr^2 bits of data, where α is a constant. We define $\gamma_{s,t}$ as the energy consumed by server s to download a bit of video data. Consequently, the energy consumption $e^{trans}_{k,s,t}$ for edge server s to download video data of subtask $m_{k,t}$ can be expressed as:

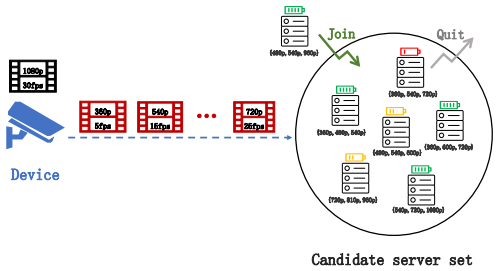


Fig. 1: Video offloading and configuration selection in multi-edge environment

$$e^{trans}_{k,s,t} = \gamma_{s,t} \alpha (x_{k,s,t} r_{k,s,t})^2 f_{k,t}. \tag{2}$$

Let $\mu_{s,t}$ denote the energy consumed by server s to process one frame. Then, the processing energy consumption $e^{pro}_{k,s,t}$ for server s to analyze subtask $m_{k,t}$ can be expressed as:

$$e^{pro}_{k,s,t} = \mu_{s,t} x_{k,s,t} f_{k,t}. \tag{3}$$

Then, the total energy consumption for processing subtask $m_{k,t}$ is $e_{k,t} = \sum_{s=1}^{|\mathcal{S}_{j,t}|} (e^{trans}_{k,s,t} + e^{pro}_{k,s,t})$. Note that, due to the dynamic variation in energy consumption on servers over time, $\gamma_{s,t}$ and $\mu_{s,t}$ are stochastic.

Problem Formulation. Our goal is to maximize the accuracy of video analytics while minimizing overall energy consumption. We introduce a utility function to evaluate the quality of video transmission configuration which is defined as the difference between the video analytics accuracy and the energy consumption generated under a certain transmission configuration:

$$P_{k,t} = a_{k,t} - \omega e_{k,t}, \tag{4}$$

in which the weight parameter ω controls the trade-off between accuracy and energy consumption. When ω is large, the algorithm will prioritize reducing energy consumption by sacrificing accuracy. The overall problem is as follows:

$$\begin{aligned} \mathcal{P}: \max \quad & \sum_{t=1}^{T} \sum_{k=1}^{K_t} \sum_{j=1}^{|\mathcal{J}|} (a_{k,t} - \omega e_{k,t}) \\ s.t. \quad & C_1: x_{k,s,t} \in \{0,1\}, \forall t, k \in \mathcal{K}_t, s \in \cup_j \mathcal{S}_{j,t}, \\ & C_2: \sum_{k=1}^{K_t} \sum_{s=1}^{|\mathcal{S}_{j,t}|} x_{k,s,t} = 1, \forall t, k \in \mathcal{K}_t, s \in \cup_j \mathcal{S}_{j,t}, \\ & C_3: f_{k,t} \in [\underline{f}, \bar{f}], \forall t, k \in \mathcal{K}_t. \end{aligned} \tag{5}$$

The constraints C_1 and C_2 jointly limit that the device can only select one server for task offloading of video analytics tasks simultaneously. Constraint C_3 restricts the selection range of video frame rates, requiring the frame rate to be neither lower than $\underline{f}$ nor higher than $\bar{f}$.

III. Algorithm Design

Analysis. The difficulty in solving $\mathcal{P}$ lies in the inability to obtain future system-related information in advance. For instance, before offloading tasks to server s, $\gamma_{s,t}$ and $\mu_{s,t}$

are stochastic. This leads to a trade-off between "exploration" and "exploitation" during the learning process. Specifically, it involves deciding whether to offload video tasks to servers that have not been previously selected to gather more information or to make the best decision based on existing information. In the long run, "exploration" may yield higher returns for future selections, but the results obtained at the moment may not be optimal. Conversely, continuous "exploitation" is more conducive to making the best decision at the current moment but may lead to the solution being trapped in a local optimum.

The uncertainty of the candidate server set further complicates the problem. Edge servers may join or quit the candidate cluster due to energy depletion or mobility. Traditional online learning algorithms may not cope with this unstable scenario, necessitating a restart of the learning process. However, apart from the servers that join or quit, the information of other servers in the cluster remains unchanged. If the learned information is not reused and the learning process starts from scratch, it will result in a large amount of redundant learning, thereby significantly reducing overall efficiency.

Overview. This work proposes VACS, a Volatile MAB-based Configuration Selection algorithm based on Volatile Multi-Arm Bandit [17]. VACS continuously "explores" the candidate edge server set by selecting edge servers that have not been chosen for offloading video tasks, updating the relevant server performance information of the current selection. When the candidate edge server set becomes relatively stable, VACS "exploits" the acquired information to estimate the expected utility rewards of each server, making the optimal offloading decision and video transmission configuration selection at the current time.

VACS Details. VACS utilizes Upper Confidence Bound (UCB) to solve the tradeoff between exploration and exploitation. The core idea is to select the action with the maximum upper confidence bound of expected rewards, which consists of two components: the mean cumulative reward and uncertainty measure. The mean cumulative reward reflects the "exploitation" value of an action by calculating the average return obtained after selection, while the uncertainty measure quantifies the "exploration" value of the action.

We denote the time when an edge server first joins the candidate set as $u_{k,t}$. If a server re-joins the candidate edge server cluster after exiting, it will be considered as a new available server. It is assumed in this work that the candidate server set remains unchanged within each slot t.

For each subtask $m_{k,t}$, the objective is to find the edge server that maximizes the utility function $P_{k,t}$ and determine the transmission configuration. The utility reward function $V_{k,t}^{s,r^*,f^*}$ is defined as the value of the utility function $P_{k,t}$ when offloading the subtask $m_{k,t}$ to edge server s, with the resolution $r_{k,s,t}^*$ and frame rate$f_{k,t}^*$. That is,

$$V_{k,t}^{s,r^*,f^*} = \epsilon_t(r_{k,s,t}^*)\phi_t(f_{k,t}^*) - \omega[\gamma_{s,t}\alpha(r_{k,s,t}^*)^2 f_{k,t}^* + \mu_{s,t} f_{k,t}^*]. \quad (6)$$

VACS is shown in Alg. 1. Lines 4-10 represent the continuous "exploration" phase, which is used to initialize edge servers that join for the first time. Since each edge server only supports several specific input resolutions, VACS attempts to offload analytics subtasks to these servers at all available resolutions to gather relevant performance information, while the information on the remaining available servers is retained and reused. Here, $\tilde{\gamma}_{s,t}$ and $\tilde{\mu}_{s,t}$ are the estimated values of $\gamma_{s,t}$ and $\mu_{s,t}$, respectively, and $\bar{\gamma}_{s,t}$ and $\bar{\mu}_{s,t}$ represent the sample means. Lines 12-17 represent the "exploitation" learning phase. When the candidate edge server set is stable, the current subtask selects an edge server for offloading based on the following rule:

$$s \leftarrow \arg\max\{\bar{V}_{k,t}^{s,r^*,f^*}(\bar{\gamma}_{s,t},\bar{\mu}_{s,t}) + \lambda\sqrt{\frac{2\ln(m_{k,t}-u_{k,t})}{\pi_{s,t}}}, \quad (7)$$

where $\bar{V}_{k,t}^{s,r^*,f^*}$ represents the mean cumulative reward obtained by choosing to offload the subtask to server s. $\pi_{s,t}$ denotes the number of times the server s has been chosen for offloading within slot t. The uncertainty measure $\sqrt{\frac{2\ln(m_{k,t}-u_{k,t})}{\pi_{s,t}}}$ is used to gauge the "exploration" value of server s. It decreases as the number of times the server is chosen increases; that is, if the server is chosen less frequently, it is considered to have higher "exploration" value, and VACS is more likely to select it in decision-making. λ is a weight that balances "exploration" and "exploitation", adjusting the tendency of VACS to "explore" or "exploit".

After determining the target server for offloading, the frame rate for video transmission is then established. Once the edge server and the video transmission resolution $r_{k,s,t}$ are set, a set of frame rates $f_{k,t}$ can be determined to maximize

Algorithm 1: VACS

1 **for** $t = 1$ *to* T **do**
2 estimate accuracy parameters ϵ_t and ϕ_t;
3 **for** $j = 1$ *to* $|\mathcal{J}_t|$ *and each subtask* $m_{k,t}$ **do**
4 **if** $\exists$ *new edge server* s **then**
5 $u_{k,t} \leftarrow m_{k,t}$;
6 select server s, observe $\tilde{\gamma}_{s,t}$ and $\tilde{\mu}_{s,t}$;
7 **for** $r_{k,s,t}$ *in* $\mathcal{R}_s$ **do**
8 get frame rate $f_{k,t}$ according to $r_{k,s,t}$;
9 get best configuration $r_{k,s,t}^*$ and $f_{k,t}^*$;
 $\bar{V}_{k,t}^{s,r^*,f^*}(\bar{\gamma}_{s,t},\bar{\mu}_{s,t}) \leftarrow \tilde{V}_{k,t}^{s,r^*,f^*}(\tilde{\gamma}_{s,t},\tilde{\mu}_{s,t})$;
10 $\pi_{s,t} \leftarrow 1$;
11 **else**
12 select server s using Eq. (7), observe $\tilde{\gamma}_{s,t}$ and $\tilde{\mu}_{s,t}$;
13 **for** $r_{k,s,t}$ *in* $\mathcal{R}_s$ **do**
14 get frame rate $f_{k,t}$ according to $r_{k,s,t}$;
15 get best configuration $r_{k,s,t}^*$ and $f_{k,t}^*$;
16 $\bar{V}_{k,t}^{s,r^*,f^*}(\bar{\gamma}_{s,t},\bar{\mu}_{s,t}) \leftarrow \frac{\bar{V}_{k,t}^{s,r^*,f^*}(\bar{\gamma}_{s,t},\bar{\mu}_{s,t})\pi_{s,t} + \tilde{V}_{k,t}^{s,r^*,f^*}(\tilde{\gamma}_{s,t},\tilde{\mu}_{s,t})}{\pi_{s,t}+1}$;
17 $\pi_{s,t} \leftarrow \pi_{s,t} + 1$;

$P_{k,t}$. Specifically, for a chosen server s, the optimal set of frame rates $\mathcal{F}_{k,t}^{s}$ can be identified by computing the first-order derivative of $P_{k,t}$ with respect to $f_{k,t}$ in $\mathcal{F}_{k,t}^{s}$ equals 0, i.e.,

$$\mathcal{F}_{k,t}^{s} = \{f_{k,t} | x_{k,s,t} = 1 \wedge \partial P_{k,t}/\partial f_{k,t} = 0\}, \tag{8}$$

then we get the optimal frame rate $f_{k,t}^{*}$ by solving

$$f_{k,t}^{*} = \underset{f_{k,t} \in \mathcal{F}_{k,t}^{s} \cup \{\underline{f}, \bar{f}\}}{\arg\max} P_{k,t}. \tag{9}$$

Theoretical Analysis. Define the regret of task m_t as:

$$Reg(t) = \sum_{k=1}^{K_t} \mathbb{E}[V_{k,t}^{s^*,r^*,f^*}] - \mathbb{E}[V_{k,t}^{s,r^*,f^*}], \tag{10}$$

where $V_{k,t}^{s^*,r^*,f^*}$ is the theoretical optimal utility function value obtained by selecting the optimal server s^* with future information, and $V_{k,t}^{s,r^*,f^*}$ is the utility obtained by VACS. We have the following theorem; for proof please refer to [31].

Theorem 1. *Without the prior information of $\gamma_{s,t}$ and $\mu_{s,t}$, the upper bound of the regret for each analytics task m_t is:*

$$\mathbb{E}[Reg(t)] \leq |\mathcal{J}_t| \sum_{s \neq s^*} \lambda(8\Delta_s^{-1} \ln K_t + \frac{8}{3}\Delta_s), \tag{11}$$

where $\Delta_s \triangleq \mathbb{E}[V_{k,t}^{s^*,r^*,f^*}/\lambda] - \mathbb{E}[V_{k,t}^{s,r^*,f^*}/\lambda]$.

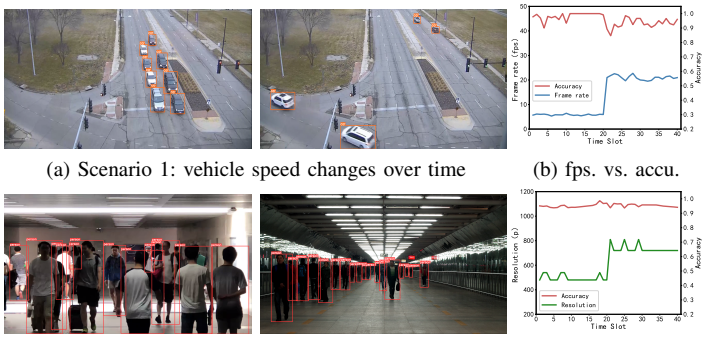
IV. Experiments and Conclusion

Settings. Our extensive trace-driven experiments use the videos derived from the AI City Datasets 2019 [18], with the row resolution of 1080p and frame rate of 30fps. YOLOv5 [19] is deployed on edge servers for object detection, supporting resolutions including 360p, 480p, 540p, 600p, 720p, 810p, 960p, and 1080p, with each server supporting three of these resolutions. Following related work [20, 21], we set the computational energy consumption $\mu_{s,t} \sim N(5, 0.5)$J/frame and the transmission energy consumption $\gamma_{s,t} \sim N(5, 0.5) \times 10^{-6}$(J) by default. We use F1-score [26] to measure the accuracy (details can be found in [31]).

Baselines include (1) Accuracy-Optimal (AO), which maximizes accuracy and ignores energy consumption, and (2) Energy-Consumption-Optimal (ECO), which minimizes energy consumption and ignores accuracy.

Effectiveness of VACS. Fig. 2 shows two typical examples of using VACS. In Figs. 2(a) and 2(b), during the first 20 slots, vehicles move slowly, and the differences between adjacent frames are small. Sampling the video at a lower frame rate can still achieve an average accuracy of 98%. In the latter 20 slots, vehicles move faster, necessitating a higher frame rate to maintain high accuracy. Similarly, as shown in Figs. 2(c) and 2(d), in the first 20 slots, pedestrians are close to the camera and gradually move away in the last 20 slots. To maintain high accuracy as pedestrian size decreases, VACS opts to transmit the video at a higher resolution.

Comparison Results. Figs. 3 shows the comparison results, in which we assume that the highest configuration (e.g., 1080p and 30fps) achieves an accuracy of 1. VACS achieves an average accuracy of 87%, only a 13% reduction compared to AO, but it saves 59% in energy consumption. While ECO

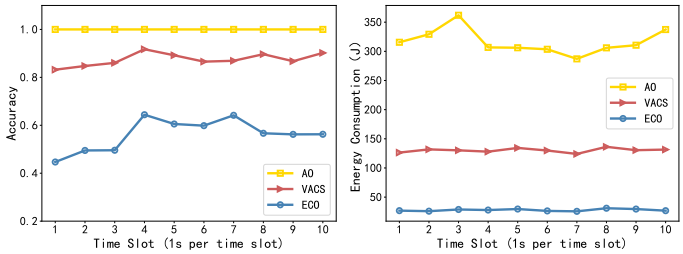


(a) Scenario 1: vehicle speed changes over time (b) fps. vs. accu.

(c) Scenario 2: pedestrian size changes over time (d) res. vs. accu.

Fig. 2: VACS dynamically adjusts configuration

achieves the lowest overall energy consumption, its average accuracy is only 56%, failing to meet the accuracy requirements of most video analytics applications. Therefore, VACS manages to strike a good balance between accuracy and energy consumption.



(a) Accuracy comparison (b) Energy consumption comparison

Fig. 3: Comparison results

Effect of ω and Volatile Bandit. Our evaluation results show that (1) by setting ω to an appropriate value, it is possible to ensure high accuracy while maximizing the reduction in energy consumption costs, and (2) Volatile Bandit indeed helps improve utility. Please refer to [31] for more details.

Conclusion and Future Work. In this work, we formulate a mixed planning program that aims to maximize the analytics accuracy while minimizing the energy consumption. Due to the dynamic availability of edge servers and the variability of server performance, it is difficult to predict candidate server performance in advance and make offloading and configuration selections accordingly. We propose a volatile MAB-based configuration selection algorithm, VACS, to enhance online learning efficiency by fully utilizing the information obtained. Rigorous proof measures the gap between online decisions and the optimum. Extensive experiments validate the effectiveness of the algorithm. For future work, we plan to investigate neural codecs for machine-centric video transmission and analytics.

Acknowledgments. This work was supported in part by NSFC (62202233), Double Innovation Plan of Jiangsu Province (JSSCBS20220409), Grant from State Key Laboratory for Novel Software Technology, Nanjing University (KFKT2024B18).

REFERENCES

[1] J. Wu, C. Leng, Y . Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *IEEE CVPR*, 2016, pp. 4820–4828.

[2] K. Du, A. Pervaiz, X. Y uan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *ACM SIGCOMM*, 2020, pp. 557–570.

[3] G. Raghavan, A. Salomaki, and R. Lencevicius, "Model based estimation and verification of mobile device performance," in ACM EMSOFT, 2004, pp. 34–43.

[4] C. Wang, S. Zhang, Y . Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *IEEE INFOCOM*, 2020, pp. 257–266.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "Y ou only look once: Unified, real-time object detection," in *IEEE CVPR*, 2016, pp. 779–788.

[6] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," 2013.

[7] T.-Y. Lin, P . Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE CVPR*, 2017, pp. 2117–2125.

[8] H. Qian and D. Andresen, "Reducing mobile device energy consumption with computation offloading," in *IEEE SNPD*, 2015, pp. 1–8.

[9] H. Li and L. Chen, "Rssi-aware energy saving for large file downloading on smartphones," *IEEE LES*, vol. 7, no. 2, pp. 63–66, 2015.

[10] Y . Sun, S. Zhou, and J. Xu, "Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE JSAC*, vol. 35, no. 11, pp. 2637–2646, 2017.

[11] Y. Chen, S. Zhang, M. Xiao, Z. Qian, J. Wu, and S. Lu, "Multi-user edge-assisted video analytics task offloading game based on deep reinforcement learning," in *IEEE ICPADS*, 2020, pp. 266–273.

[12] M. Hu, Z. Xie, D. Wu, Y . Zhou, X. Chen, and L. Xiao, "Heterogeneous edge offloading with incomplete information: A minority game approach," *IEEE TPDS*, vol. 31, no. 9, pp. 2139–2154, 2020.

[13] Y. Chen S. Zhang, Y. Jin, Z. Qian, M. Xiao, N. Chen, Z. Ma, "Learning for Crowdsourcing: Online Dispatch for Video Analytics with Guarantee," *IEEE INFOCOM*, 2022, pp. 1908-1917.

[14] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *ACM SIGCOMM*, 2018, pp. 253–266.

[15] 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 *NSDI*, 2017, pp. 377–392.

[16] Q. Liu, S. Huang, J. Opadere, and T. Han, "An Edge Network Orchestrator for Mobile Augmented Reality," in *IEEE INFOCOM*, 2018, pp. 756–764.

[17] Z. Bnaya, R. Puzis, R. Stern, and A. Felner, "Social network search as a volatile multi-armed bandit problem," *Human*, vol. 2, no. 2, pp. pp–84, 2013.

[18] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P . Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu, "The 2019 ai city challenge," in *IEEE CVPR* Workshops, 2019, p. 452–460.

[19] G. J. et al., "ultralytics/yolov5: v3.0," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3983579

[20] H. Qian and D. Andresen, "Reducing mobile device energy consumption with computation offloading," in *IEEE SNPD*, 2015, pp. 1–8.

[21] H. Li and L. Chen, "Rssi-aware energy saving for large file downloading on smartphones," *IEEE LES*, vol. 7, no. 2, pp. 63–66, 2015.

[22] J. Li, L. Liu, H. Xu, S. Wu, and C. J. Xue, "Cross-camera inference on the constrained edge," in *IEEE INFOCOM 2023*. IEEE, 2023, pp. 1–10.

[23] R. Xu, S. Razavi, and R. Zheng, "Edge Video Analytics: A Survey on Applications, Systems and Enabling Techniques," IEEE Communications Surveys & Tutorials, 2023.

[24] J. Ye, H. Yeo, J. Park, and D. Han, "AccelIR: Task-Aware Image Compression for Accelerating Neural Restoration," in *IEEE/CVF CVPR*, 2023, pp. 18 216–18 226.

[25] N. Chen, S. Quan, S. Zhang, Z. Qian, "Cuttlefish: Neural Configuration Adaptation for Video Analysis in Live Augmented Reality," in *TPDS*, 2021, pp. 830–841.

[26] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European conference on information retrieval*. Springer, 2005, pp. 345–359.

[27] K. Du, A. Pervaiz, X. Y uan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *ACM SIGCOMM*, 2020, pp. 557–570.

[28] Z. Ming, J. Chen, L. Cui, S. Yang, Y. Pan, W. Xiao, and L. Zhou, "Edge-based video surveillance with graph-assisted reinforcement learning in smart construction," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9249–9265, 2022.

[29] R. Zhang, Y. Zhou, F. Wang, and Z. Wang, "Maxim: Drl-based cross-camera streaming configuration for real-time video analytics," in *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 01–06.

[30] Y. Zhou, H. Sun, Y. Jin, Y. Zhu, Y. Li, Z. Qian, S. Zhang, and S. Lu, "Inference replication at edges via combinatorial multi-armed bandit," *Journal of Systems Architecture*, vol. 129, p. 102636, 2022.

[31] https://cs.nju.edu.cn/sheng/SupplementalVACS.pdf.