March 2021

CONFIGURATION-ADAPTIVE EDGE-ASSISTED INDUSTRIAL WIRELESS CAMERAS WITH DEEP REINFORCEMENT LEARNING

HP INC

Configuration-Adaptive Edge-Assisted Industrial Wireless Cameras with Deep Reinforcement Learning

Abstract

Embodiments of the present disclosure provide a method and system for designing and implementing an industrial smart camera that uses low-power wireless communications and edge computing to achieve cordless and energy-efficient visual sensing. The smart camera applies deep reinforcement learning to adapt the camera configuration to maintain the desired visual sensing performance with the minimum energy consumption under dynamic variations of application requirement and wireless channel conditions.

Field of Technology

This disclosure relates to a method and system for designing an industrial smart wireless camera to implement cordless and energy-efficient industrial computer vision systems.

Overview

Computer vision has become an essential component of the automated inspection processes in smart manufacturing systems. Application examples include product quality inspection, manufacturing system fault diagnosis, and activity monitoring. Without relying on cables for network connectivity and power supply, the wireless cameras offer several benefits such as easy deployment, mobility support, and unobtrusiveness to the ongoing industrial processes. As today's wireless cameras become smaller in form factors, they are promising for swift ad hoc deployments in industrial event-based or schedule-based diagnostic tasks.

An industrial visual sensing system in general involves compute-intensive image processing. Deep learning has been increasingly used for industrial computer vision applications. However, the execution of deep models imposes high demand on computing resources. On the other hand, to achieve the cordless setting for agility, the wireless cameras are often powered by batteries with finite capacities. Therefore, running the compute-intensive deep models on the wireless cameras is not desirable since otherwise bulky batteries or wired power supply will be needed.

The present disclosure provides a method for implementing an industrial smart wireless camera that uses a wall-powered wireless edge node with sufficient computing resources to support the front-end wireless cameras in facilitating deep model execution. Specifically, the wireless camera captures images, performs the local image processing to generate smaller representations of the captured images, and transmits the representations to the edge node for advanced processing. Up on receiving the image data from the camera, the edge node reconstructs images and performs advanced visual processing using the deep model. The disclosed smart camera is capable of dynamically updating the configuration for various parameters such as frame capture rate, image resolution and local image processing mode.

Due to dynamic changes of visual sensing performance application requirement and wireless channel conditions, adaptation of the camera's configuration is desired in industrial visual sensing systems. A few existing works [1] propose hysteresis-based control approaches to adapt the configuration of the camera. However, the hysteresis-based approach often has poor performance under dynamic application requirements and wireless channel conditions. To solve the problem, the present disclosure applies the model-free deep reinforcement learning to learn the optimal configuration adaption policy that can achieve desired visual sensing performance with the minimum camera energy consumption under variations of industrial application requirements and wireless channel conditions.

Detailed Description

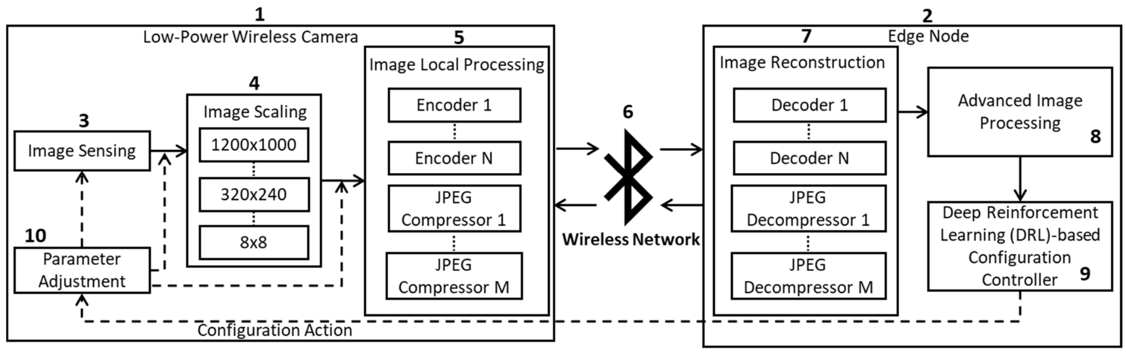

FIG. 1.

Fig. 1 overviews the design of the disclosed smart wireless camera system which has two main components: the low-power wireless camera 1 and the edge node 2. The wireless camera 1 is prototyped by an off-the-shelf battery-powered wireless camera, called the ESP32-CAM [2], consisting of an ov2640 camera module that supports an image resolution up to 1600x1200. The ESP32-CAM supports the TensorFlow Lite (TFLite) Micro, a deep learning library tailored for microcontrollers which allows to run the machine learning models on the microcontroller. Moreover, a Raspberry Pi 4 unit is used to prototype the functions of the edge node 2. The following sections will present the detailed implementation of the end-to-end image processing pipeline of the smart wireless camera system in the present disclosure.

The camera first performs image sensing 3 that writes captured image frames to the SRAM memory of ESP32-CAM. The OV2640 camera sensor of ESP32-CAM can support image capture with 8 resolution levels between 160x120 and 1600x1200. The image solution has significant impacts on the visual sensing performance and camera energy consumption. To support any image resolution that may be lower than the minimum of 160x120 supported by OV2640, a bilinear scaling-based image resizing module 4 is implemented to adjust resolution of the captured image to a desired resolution.

Captured images are first processed at the local image processing module 5 before transmitting the image data to the edge node. Specifically, the local image processing module 5 supports multiple JPEG compression modes with the quality index from 0 to 80. In addition, the convolutional autoencoder is also implemented in the disclosed smart camera. A convolutional autoencoder has two parts: encoder and decoder. The encoder, consisting of three convolution layers and one output layer is employed in the local image processing module 5 of the wireless camera to exact the high-level representation of a raw image. The decoder with nine convolution layers is deployed at the edge node to reconstruct the approximation of the image based on the data representation received from the camera. The rectified linear units are used as the activation function for convolution layers in both encoder and decoder, while the sigmoid activation is used at the output layers. The disclosed camera supports multiple autoencoder modes with different size of filters in convolution layers and output layers of the encoder.

The compressed image representation data that are the output of the local image processing module 5 are transmitted to the edge node using the Bluetooth low energy (BLE)-based wireless network 6. At the edge node, the received image data are fed into the image reconstruction module 7 to reconstruct the approximation of the original image. Then, the reconstructed images are further processed at the advanced image processing module 8. Deep neural networks are implemented to classify or recognize objects in the images.

At the edge node, a deep reinforcement learning (DRL)-based configuration controller 9 is implemented to adapt the configuration for the camera's parameters in response to changes of visual sensing performance requirement and industrial wireless channel condition. Specifically, at the beginning of every adaptation period *T*, the controller 9 obtains the current system state denoted by *x*, including the result of the advanced image processing module 8 and the received signal strength indicator of the radio signal sensed by the edge node. Based on the current system state, the controller determines a configuration action, denoted by *a*. Then, the edge node transmits the configuration action to the camera. Up on receiving the configuration action, at the camera, the parameter adjustment module 10 sets the frame capture rate, image resolution and local image pre-processing mode (i.e., the JPEG compression modes with a quality index or autoencoder modes with a setting for its hyperparameters) to desired values for the camera operation during the period of *T*.

The main objective of the deep reinforcement learning (DRL)-based configuration controller 9 is to find an adaptation policy that determines the action *a* based on the state *x* to maximize the expected reward denoted *E[r]* by over a long run. The reward function *r* is defined as $r = -e(x, a) - p(x, a)$, where $e(x, a)$ and $p(x, a)$ are the total energy consumed by the camera for the image sensing, scaling, local processing and data transmission over the adaptation period of *T* and penalty, respectively, given a pair of state *x* and action a. Let $Lmax(x, a)$ and φ denote the maximum image processing latency and the number of images that are captured and correctly recognized during the period of *T,* respectively. The penalty is defined as $p(x, a) = \lambda 1. N(Lmax(x, a) - Lth) + \lambda 2. N(\varphi req - \varphi)$, where $\lambda 1, \lambda 2$ are configurable weights, Lth is the upper bound threshold for the maximum latency, φreq is the number of correctly recognized images required over the period of *T,* and $N(X) = \frac{\max(X,0)}{Xmax}$.

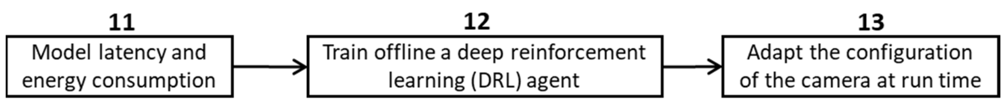

FIG. 2

For the configuration adaptation of the disclosed smart camera, the typical online training of the deep reinforcement learning agent [3] has the following two issues. First, it may take a long time to converge, which may lead to the camera's excessive battery energy consumption. Second, during the online learning phase, measuring the camera's power and image processing accuracy is cumbersome or infeasible. The camera is not capable of metering its power in real time, which requires an external power meter. Moreover, the image classification accuracy cannot be obtained during the online learning due to the lack of ground truth labels.

To address these issues, the present disclosure adopts an offline training approach as illustrated in Fig. 2. At the first phase 11, the image processing latency and energy consumption models are built based on real data traces collected from the deployment field. In the second phase, the real data traces and models built in the first phase are used to drive the offline training of the deep reinforcement learning agent. After the completion of the offline training, the trained deep reinforcement learning agent is commissioned to adapt the configuration of the camera for visual sensing tasks at run time.

References

[1] C. Josephson, L. Yang, P. Zhang, and S. Katti, "Wireless computer vision using commodity radios," in proceedings of 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 229-240, 2019.

[2] ESP32, https://www.espressif.com/en/products/socs/esp32.

[3] V. Mnih et al., "Human level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.

Disclosed by Kok Loon Ho, Rui Tan, Siyuan Zhou, Duc Van Le and Qiping Yang, HP Inc.