Special Section: Security and Privacy-Aware Emerging Computing

Real-Time Physical Threat Detection on Edge Data Using Online Learning

Utsab Khakurel and Danda B. Rawat
Howard University, USA

***Abstract*—Sensor-powered devices offer safe global connections; cloud scalability and flexibility, and new business value driven by data. The constraints that have historically obstructed major innovations in technology can be addressed by advancements in Artificial Intelligence (AI) and Machine Learning (ML), cloud, quantum computing, and the ubiquitous availability of data. Edge AI (Edge Artificial Intelligence) refers to the deployment of AI applications on the edge device near the data source rather than in a cloud computing environment. Although edge data has been utilized to make inferences in real-time through predictive models, real-time machine learning has not yet been fully adopted. Real-time machine learning utilizes real-time data to learn on the go, which helps in faster and more accurate real-time predictions and eliminates the need to store data eradicating privacy issues. In this article, we present the practical prospect of developing a physical threat detection system using real-time edge data from security cameras/sensors to improve the accuracy, efficiency, reliability, security, and privacy of the real-time inference model.**

ANY SYSTEM OF PHYSICAL hardware or devices that receive and transmit data via networks without human involvement is referred to as an Internet of Things (IoT) system [8]. Some examples of IoT applications include smart cities, healthcare, smart farming, retail and logistics, and smart homes. IoT devices are widely utilized to automate processes, improve performance, and save energy. 11.7 billion (54%) of the 21.7 billion connected devices in use in 2021 were Internet of Things (IoT) devices. According to [1], there will be 30 billion (or 75%) active IoT devices by 2025, or an average of 4 devices per person. IoT and sensor devices are generating data exponentially, which can be used to predict trends. Edge computing is the architecture in which the processing and storing of data happens close to the data source. Real-time processing at the edge is more efficient, accurate, and necessary as the majority of the data is created there. The three components of the edge computing architecture are the cloud layer, the boundary layer, and the terminal layer [2].

Digital Object Identifier 10.1109/MCE.2022.Doi Number
Date of publication 00 xxxx 0000; date of current version 00 xxxx 0000

Edge AI deploys AI applications on edge devices near the data source, enabling deployment of AI in resource-constrained environments with real-time insights, reduced latency, cost, and increased privacy [10]. YOLO (You Only Look Once), a state-of-the-art object detection system, provides high accuracy and real-time object detection at up to 45 fps, making it useful for Edge AI applications in autonomous vehicles and surveillance systems [4]. Edge AI can be used to process real-time data from IoT sensors to detect potential security systems threats. YOLO object detection algorithms can identify physical threats on the edge and take immediate action. Few physical security systems in the market can perform real-time detection using visual, night vision, and thermal video frames [9]. Current real-time AI detection systems require batch-trained models, which are costly and time-consuming to update, and deploying the detection model on the cloud causes data transfer latency. Deploying AI at the edge with real-time learning can improve accuracy, efficiency, security, and privacy in security systems, while addressing dynamic and evolving threats.

In this article, we propose a real-time physical threat detection framework that combines edge-based object detection AI with online learning to improve accuracy, efficiency, reliability, security, and privacy. Edge deployment leads to a fast and efficient system with minimal resources required. Online learning allows the model to learn from new data in real-time, with fast and affordable learning steps [3]. The proposed edge-based model ensures privacy protection by immediately utilizing and discarding data, while also being capable of offline operation, making it dependable and trustworthy. Additionally, the model's ability to operate through visible and thermal images further enhances its reliability. Our solution enhances physical security systems through the combination of edge computing and incremental online learning.

BACKGROUND AND RELATED WORK

Overview of Object Detection (OD)

Object Detection (OD) is a computer vision technique for detecting, locating, and classifying objects in images. Two main types of OD algorithms are two-shot detection and single-shot detection. YOLO [4] takes an image and splits it into S x S grid, where each grid will have parameters defining an object as $[P_c, b_x, b_y, b_w, b_h, c]$, where, P_c denotes the confidence score for the object in the box, (b_x, b_y) represents the center of the box relative to the grid cell, (b_w, b_h) represents the width and height relative to the whole image, and c denotes the presence of each class in the cell. YOLO is fast, accurate, and the best option for real-time object detection, utilizing anchor boxes and non-maximal suppression to improve performance [4].

Overview of Online Learning (OL)

ML has flourished with the availability of data, storage, and processors, but data generation surpasses its usage rate. Online learning trains models in real-time by incrementally learning from continuous data input, eliminating the need for data storage [3]. However, faulty data could negatively impact performance, so proper data governance is required.

Related Work and Objectives

In airport security, deep learning approaches have been utilized to detect potential threats in X-ray and Thermal Infrared (TIR) images [5]. The effectiveness of different versions of YOLO models is evaluated using Teledyne Forward-looking Infrared (FLIR) Thermal Dataset in [6]. Another study utilized YOLOv3 and demonstrated its performance using a combination of pre-training on Imagenet and a custom gun dataset [7]. Edge YOLO is a lightweight version of the YOLOv4 object detection algorithm that has been optimized for edge computing [11]. The algorithm features a trimmed-down backbone using CSPNet, enhanced feature fusion, and a connection to a cloud-based GPU workstation for model training. Edge YOLO outperforms other popular algorithms, such as YOLOv3 and MobileNetv3 SSD, in terms of both speed and accuracy on the edge.

Although research has experimented to make real-time inferences with image and video frames with object detection and edge computing as a single domain, none of the approaches focuses on utilizing the real-time data to keep the model up-to-date. Real-time training of threat detection models on edge devices is cost-effective, power-efficient, and privacy-preserving. It allows for continuous adaptation to changing data, improving the model's performance. Online learning is well-suited for edge devices as it requires minimal storage, is maintainable, and fast [3]. The proposed solution utilizes a low-power edge processor to process visual and thermal video frames in real-time. The

cloud environment label annotates input data and updates the online feature store to update the weights of the local model copy (online model) in real-time. The edge model is updated by pulling weight parameters from the online model through set triggers, improving the physical security system's speed, accuracy, and reliability. This approach combines edge object detection and online learning methods to create a scalable, secure, and fast physical threat detection model that trains and predicts on real-time data. The features of the proposed approach are as follows:

- Proposed approach uses a lightweight version of the object detection algorithm on the edge to make a real-time inference contributing to faster inference time and offline availability of the service.
- Visual and Thermal Infrared (TIR) images are used to train the model to ensure the availability of service at different weather conditions, and different time of the day.
- Online learning is utilized to train the model incrementally in real-time improving the performance of the model, and purging the need for data storage.
- Creates privacy-preserved model learning environment with no stored data.

REQUIREMENTS FOR PHYSICAL THREAT DETECTION MODEL

Selecting the proper dataset, model, and evaluation metrics are of key importance for any AI system to succeed. This section elaborates on these major components required to create the online physical threat detection system on the edge.

The Teledyne FLIR ADAS Dataset offers labeled thermal and visible images to train object detection systems using CNNs that recognize threats in both image types. The dataset includes images in various weather conditions to improve the detection system's adaptability. Common benchmark datasets such as ImageNet and COCO are often insufficient due to the limited availability of annotated data with diverse object labels. There are few labeled datasets that identify objects as threats, requiring custom images and videos with threat objects and proper label annotations to design the proposed online physical threat detection system. Pre-trained models on large datasets are frequently used and then retrained on custom datasets.

Although Faster R-CNN and RetinaNet are more accurate than YOLO, YOLO's real-time detection capability outshines other detection algorithms. In [6], YOLOv3-SPP was identified as having high mAP and precision, but not ideal for edge deployment due to its low speed and high storage requirements. YOLOv5-s is recommended for edge deployment with a compact size of 14 MB, fast speed of 41 FPS, and high mAP and precision of 0.803 and 0.638 respectively, outperforming YOLOv3-SPP. GhostNet [12] and coordinate attention (CA) [13] methods can make YOLOv5-s even more lightweight but equally efficient. Using both techniques, [14] generated reduced boundary box loss, classification loss, and object confidence loss in self-constructed data. [11] demonstrated better performance of the YOLOv4 version with a trimmed-down CSP-darknet53 and a modified backbone neck with Spatial Pyramid Pooling (SPP) and Feature Pyramid Network (FPN) for cost-effective small object detection.

The modified GhostNet and CA backbone of YOLOv5-s can be improved with the same lightweight neck structure used in YOLOv4 to enhance our physical threat detection model performance. YOLOv5 uses batch normalization, leaky ReLU activation, Stochastic Gradient Descent optimization, binary cross-entropy for classification loss, and mean squared error for coordinate regression. Mean Average Precision (mAP), Intersection over Union (IoU), Precision, Recall, and Frames Per Second (FPS) are used to evaluate the proposed system.

DESIGN OF THE PROPOSED FRAMEWORK

The proposed real-time physical threat detection framework with online learning is presented in Figure 1. It is composed of two parts: learning and implementation phases, which are explained in detail in this section.

Learning Phase

The learning phase consists of the process of preprocessing the dataset, using the data to train the YOLO model, and using the validation dataset to test the model.

To improve the algorithm's ability to predict images, data augmentation is necessary for both the training and validation datasets. This includes random crop, rotation, flips, saturation, and exposure shifts. Manual labeling is required for the augmented images and newly captured images containing threat labels. A minimum of 1500 images per class and 10000 labeled objects per class are recommended, and adding background images with no objects can help reduce

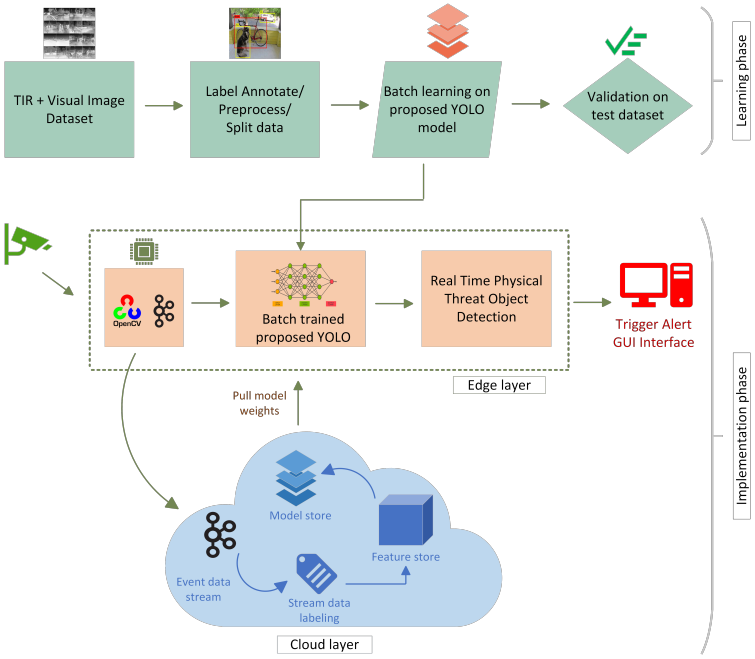

Figure 1. Real-time Physical Threat Detection framework utilizing Online Learning.

false positives.

To make the object detection model operational, the proposed YOLO model is fed with annotated thermal and visual images. Pre-trained weights are suggested for small or medium datasets. The YOLOv5 model's default settings obtain good results on large and well-labeled datasets with 300 epochs recommended. Overfitting can occur if the model is too complex. Regularization techniques such as L1 (Lasso) and L2 (Ridge) regularization can be used in such cases. Cross-validation can also be used to avoid overfitting. The batch size should always be as large as the hardware allows as a lower batch size generates poor batch normalization statistics. YOLOv5 has approximately 30 hyperparameters for training. A Genetic Algorithm (GA) is provided to optimize these hyperparameters, producing an optimal value by repeatedly mutating parent hyperparameters. For best results, 300 generation cycles are recommended, and the hyperparameters from the best-performing cycle are chosen for the model.

The performance of the YOLO model on the test image dataset can be evaluated using the validation dataset. This evaluation involves estimating the model's localization and classification errors, using the evaluation metrics discussed previously. The resulting Average Precision (AP) of each class, along with the $mAP_{0.5:0.95}$ score, provides valuable insights into the model's performance, thereby helping to identify any overfitting or underperforming issues.

Implementation Phase

The implementation phase comprises of the edge application and the cloud application. The layers are discussed in detail below.

Edge Application Edge computing devices such as System on Chip (SoC), Field Programmable Gate Arrays (FPGA), Application Specific Integrated Circuits (ASIC), Central Processing Units (CPU), and Graphic Processing Units (GPU) are available for deployment. SoC is particularly notable due to its energy efficiency, small size, and high throughput. It can also use migration tools to convert deep learning frameworks into TensorRT for accelerated model inference. Among the NVIDIA Jetson series, Jetson TX2 NX and Jetson Nano are the most cost-effective.

Given adequate CUDA cores and storage capacity, Kafka is an appropriate option for the edge detection model which can operate without connecting to the cloud. Kafka provides real-time edge processing, low latency, and cost-efficient data processing in a reliable, scalable, and fault-tolerant way. Kafka clusters can be configured as a single node or a cluster with multiple brokers depending on the need for availability and resource constraints of the edge hardware. Data

retention in Kafka is configurable, and cluster linking enables connections between small Kafka clusters at the edge and bigger Kafka clusters in the cloud using the Kafka protocol.

OpenCV converts the video stream into frames and resizes if necessary. Producers populate frames into the Kafka topic, while consumers collect data to feed into the batch-trained and validated YOLO model initially deployed on the edge. The threat detection model makes real-time inferences, and if an object meets the confidence threshold for a threat, the GUI Interface is notified and the sound system triggers an alarm. The input image should match the trained dataset for optimal results. The local model updates itself by pulling new weights from the real-time model deployed in the cloud, resulting in accurate predictions.

Cloud Application The cloud application utilizes a Kafka cluster, active learning, an online feature store, and a copy of the initial YOLO model deployed at the edge. AWS offers suitable infrastructure to host clusters, create data labeling pipelines, and store online models. The replicated data streams from the edge are held in larger Kafka clusters in the cloud. The data in the Kafka cluster is partitioned into train and test topics. The data stream from the train and test topic is sent to the data labeling pipeline. A streaming labeling job is created, where human workers label the data in real-time. Streaming jobs work in a sliding window manner, and any jobs that pile up are stored in a queuing service. The labeled jobs are output through a stream channel.

Labeled data is utilized to build a feature pipeline that captures real-time features and stores them in an online feature store. The pipeline ensures thorough processing, validation, and transformation of the data into a usable format for inference or training. The in-memory database is used for the feature store to attain high throughput and low latency. The features are updated continuously in a streaming fashion to keep up with real-time data and also provide context.

The feature store employs automated stateful training to continuously train the YOLO model with real-time data and updates the online model in the model store accordingly. At the edge, the local model pulls updated model parameters based on user-defined triggers. The feature store can track the model's lineage, but evaluation of the online model using test stream data is not yet possible. Incorporating explore-exploit strategies from bandit algorithms into the feature store can be a data-efficient solution, compared to traditional A/B testing.

Our proposed approach theoretically ensures fast and accurate real-time inference at the edge by leveraging privacy-aware stateful learning. The edge device can run a lightweight threat detection model independently, providing efficiency, accuracy, security, and reliability without needing to connect to the cloud.

DISCUSSION

This article proposes an online learning-enabled real-time edge threat detection model for fast, accurate, reliable, and privacy-aware detection. However, further experimentation and validation are needed to fully realize the potential of the proposed framework. In this section, we provide insights into expected outcomes.

The study in [6] evaluates multiple versions of YOLO for multi-object detection and finds YOLOv5-s to have the best overall performance in terms of mAP, precision, speed, and storage when tested on the FLIR ADAS thermal dataset, with values of 0.803, 0.638, 41 FPS, and 14MB respectively. [14] supports the improved performance of the YOLOv5-s_Ghost_CA model for facial expression detection, with a boost in $mAP_{0.5}$ from 98.4 to 98.8 and maintained $mAP_{0.5:0.95}$. This results in reduced weights (15.4 MB to 8 MB), parameters (7.02 M to 3.70 M), and computation cost (15.8 to 8.1 GFLOPs), with improved inference time (115 FPS to 123 FPS). EdgeYOLO [11] improved FPS to 26.6 from 4.9 on YOLOv4, with COCO2017 dataset, powered on Jetson Xavier. With the KITTI dataset, EdgeYOLO improved FPS to 40.6 from 5.2 on YOLOv4, powered on Jetson Xavier. Jetson Nano, with its lower memory and computation power, showed lower FPS than Jetson Xavier, but still demonstrated improved speed between YOLOv4 and EdgeYOLO. Our framework utilizes backbone of the YOLOv5-s_Ghost_CA model [14] with modified neck from EdgeYOLO [11] combined with online learning. The proposed model combining lightweight YOLOv5-s with real-time stateful learning, has the potential to optimize weights, parameters, and computation costs while maintaining a high mAP score. Moreover, it can improve latency and reduce energy consumption, based on the promising results of these approaches.

CONCLUSION

This article proposes a real-time online physical threat detection model using edge AI computing and

online learning. The experimental nature of real-time machine learning, manual data generation, and multiple parameters pose a challenge in achieving optimal results. Repeated experimentation using optimized models and advanced edge hardware could soon realize this vision, enhancing speed, accuracy, reliability, and privacy in computing technology.

ACKNOWLEDGMENTS

This research was funded by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory and in part by the US NSF grant CNS/SaTC 2039583.

REFERENCES

1. IANS, “At 12 billion,IoT connections to surpass non-IoT devices in 2020,” 2020. The Times of India. Available: https://telecom.economictimes.indiatimes.com/news/at-12-billion-iot-connections-to-surpass-non-iot-devices-in-2020/79318722. Accessed: Oct. 30, 2022.
2. W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, ”Edge computing: A survey”, *Future Generation Computer Systems*, vol. 97, Pages 219-235, 2019.
3. S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao,“Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, 2021. https://doi.org/10.1016/j.neucom.2021.04.112.
4. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2015. ArXiv vol. abs/1506.02640. [Online]. Available: https://arxiv.org/abs/1506.02640.
5. Dhiraj and D. K. Jain, “An evaluation of deep learning based object detection strategies for threat object detection in baggage security imagery,” *Pattern Recognition Letters*, vol. 120, pp. 112–119, 2019. https://doi.org/10.1016/j.patrec.2019.01.014.
6. C. Jiang, H. Ren, X. Ye, J. Zhu, H. Zeng, Y. Nan, M. Sun, X. Ren, and H. Huo, “Object detection from uav thermal infrared images and videos using yolo models,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102912, 2022. https://doi.org/10.1016/j.jag.2022.102912.
7. A. Warsi, M. Abdullah, M. N. Husen, M. Yahya, S. Khan and N. Jawaid, ”Gun Detection System Using Yolov3,” *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, 2019, pp. 1-4, doi: 10.1109/ICSIMA47653.2019.9057329.
8. P. K. Donta, S. N. Srirama, T. Amgoth, and C. S. R. Annavarapu, ”Survey on recent advances in IoT application layer protocols and machine learning scope for research directions,” *Digital Communications and Networks*, Vol. 8, Issue 5, pp. 727-744, 2022.
9. B. Dowlen, “Scylla AI-powered Gun Detection System integrated with Intrepid Networks Response Platform”, 05-Aug.-2021. [Online]. Available: https://www.einnews.com/pr_news/547342913/scylla-ai-powered-gun-detection-system-integrated-with-intrepid-networks-response-platform. [Accessed: 05-Feb.-2023].
10. C. Zhuoqing, S. Liu, X. Xiong, Z. Cai, and G. Tu. (2021). ”A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things”. *IEEE Internet of Things Journal*. PP. 1-1. 10.1109/JIOT.2021.3088875.
11. S. Liang, H. Wu, L. Zhen, Q. Hua, S. Garg, G. Kaddoum, M. M. Hassan, and K. Yu, ”Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345-25360, Dec. 2022, doi: 10.1109/TITS.2022.3158253.
12. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, ”GhostNet: More Features From Cheap Operations”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580-1589.
13. Q. Hou, D. Zhou, and J. Feng, ”Coordinate Attention for Efficient Mobile Network Design,” *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 13708-13717, doi: 10.1109/CVPR46437.2021.01350.
14. Z. Li, J. Song, K. Qiao, C. Li, Y. Zhang, and Z. Li, ”Research on efficient feature extraction: Improving YOLOv5 backbone for facial expression detection in live streaming scenes”, *Front. Comput. Neurosci.*, 16:980063, 2022, doi: 10.3389/fncom.2022.980063.

Utsab Khakurel is a Computer Science Ph.D. candidate at Howard University under the supervision of Prof. Danda B. Rawat. Email: utsab.khakurel@bison.howard.edu.

Danda B. Rawat is a Professor and Engineering Associate Dean for Research, Executive Director of RITA at Howard University. Email: danda.rawat@howard.edu.