IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 May 2023, accepted 13 June 2023, date of publication 19 June 2023, date of current version 22 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3287496

RESEARCH ARTICLE

Trackez: An IoT-Based 3D-Object Tracking From 2D Pixel Matrix Using Mez and FSL Algorithm

NURUZZAMAN FARUQUI[1], **MD ALAMGIR KABIR**[2], **(Member, IEEE)**,
MOHAMMAD ABU YOUSUF[3], **MD. WHAIDUZZAMAN**[4], **(Senior Member, IEEE)**,
ALISTAIR BARROS[4], **AND IMRAN MAHMUD**[1]

[1]Department of Software Engineering, Daffodil International University, Birulia, Dhaka 1216, Bangladesh
[2]Division of Computer Science and Software Engineering, Mälardalen University, 722 20 Västerås, Sweden
[3]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh
[4]School of Information Systems, Queensland University of Technology, Brisbane, QLD 4000, Australia

Corresponding author: Md Alamgir Kabir (md.alamgir.kabir@mdu.se)

ABSTRACT The imaging devices sense light reflected from objects and reconstruct images using the 2D-sensor matrix. It is a 2D Cartesian coordinate system where the depth dimension is absent. The absence of a depth axis on 2D images imposes challenges in locating and tracking objects in a 3D environment. Real-time object tracking faces another challenge imposed by network latency. This paper presents the development and analysis of a real-time, real-world object tracker called Trackez, which is capable of tracking within the top hemisphere. It uses Machine Vision at the IoT Edge (Mez) technology to mitigate latency sensitivity. A novel algorithm, Follow-Satisfy-Loop (FSL), has been developed and implemented in this paper that optimally tracks the target. It does not require the depth-axis. The simple and innovative design and incorporation of Mez technology have made the proposed object tracker a latency-insensitive, Z-axis-independent, and effective system. The Trackez reduces the average latency by 85.08% and improves the average accuracy by 81.71%. The object tracker accurately tracks objects moving in regular and irregular patterns at up to $5.4ft/s$ speed. This accurate, latency tolerant, and Z-axis independent tracking system contributes to developing a better robotics system that requires object tracking.

INDEX TERMS Machine vision, the IoT edge, latency sensitivity, object tracking, 2D coordinate, 3D coordinate, Mez.

I. INTRODUCTION

Real-time object trackers are used in various sectors, including smart surveillance [1], robotics [2], medical imaging [3], autonomous vehicle [4], sports analysis support system [5], and augmented reality [6]. The actuator within an object tracker responds dynamically to feedback from object recognition algorithms that operate remotely in the cloud [7]. The camera, strategically positioned on the tracker, transforms the optical signals reflected off the objects into electrical signals for further processing [8], which are transmitted to the cloud server through an IoT device [9], [10]. The objects are in a 3-Dimensional (3D) environment, whereas the equivalent electrical signal represents a 2-Dimensional (2D) matrix of pixels. The Cartesian location of an object in the 2D pixel matrix is not the same in a 3D environment [11]. Trackez solves a critical challenge in object tracking within 2D images. In this context, tracking involves identifying the pixel location of an object on a 2D Cartesian coordinate system, determined by the (x, y) coordinates. However, the calculation based on these coordinates alone fails to accurately represent the object's position when it moves along the depth axis or the Z-axis, which reflects the object's distance from the camera. Trackez effectively addresses this problem inherent to object moving tracking from 2D images. Consequently, pinpointing the object within the pixel matrix and instructing the actuator based on the pixel values tends to result in inaccurate object

The associate editor coordinating the review of this manuscript and approving it for publication was Gongbo Zhou.

tracking. This challenge is amplified when dealing with non-stationary objects. Network latency and processing time further compound the issue by introducing time delays in real-time video communication [12]. As a result, the moving objects are not at the location where the tracker locates them. This paper presents an innovative approach to accurately track objects in a 3D environment from the 2D-pixel matrix. It further mitigates the impact of latency and processing delay resulting in an efficient and accurate real-time object tracker.

The primary aim of this study is to create a reliable and practical real-time object tracking system. To achieve this, we employed GoogleNet, a pre-trained Convolutional Neural Network (CNN), renowned for its capacity to identify objects. As a robust 22-layer deep CNN, GoogleNet has been trained on over a million images, and it demonstrates impressive accuracy in classifying up to 1000 different objects [13]. This paper addresses two primary challenges. The first is the inherent limitations of pixel location-based tracking systems. Our solution to this problem is the innovation of the Follow-Satisfy-Loop (FSL) algorithm, which improves tracking accuracy. The second challenge we confront is latency and its effect on real-time object tracking. We designed our system architecture and proposed methodology specifically to reduce the impact of latency and enhance tracking precision. To this end, we utilized Machine Vision at the IoT Edge (Mez), a publish-subscribe messaging system developed by A. George et al. [14]. The main difference between the traditional computer vision approach and Mez is that this system can dynamically adjust the quality of video frames in real-time. Remarkably, it can tolerate latency variations up to $10x$, ensuring that our object tracker maintains accuracy even under less-than-ideal conditions.

The research methodology developed and demonstrated in this paper has been founded on practical empirical data tracked and preserved during the experiment. The core contributions of this research are:

- Design and development of the physical model of a real-time object tracker.
- Statistical and empirical analysis on the system response to discovering the major impediments of tracking objects in real-time with acceptable accuracy,
- Development of the novel FSL algorithm to overcome the major impediments,
- Overcoming the real-time object tracking system challenges by successfully integrating Mez technology as the communication medium between the object tracker and cloud server,
- Improving the object tracking accuracy of Trackez by 81.71% and
- Reducing the network latency in video communication by 85.08% while maintaining acceptable frame quality for object recognition.

The remainder of this paper is structured into six distinct sections. Section two delves into a comprehensive literature review, identifying shortcomings in current cutting-edge research and pinpointing the research gap. The proposed methodology is elucidated in the third section. In the fourth section, we present our experimental findings and evaluation. The fifth section discusses the limitations of the experimental analysis featured in this paper and outlines future directions for this research. The paper culminates with a conclusion in the sixth and final sections.

II. LITERATURE REVIEW

Hyun et al. [15] present Sparse Graph Tracker (SGT), an online graph tracker leveraging higher-order relational features for video object tracking. SGT graphs video data to fix tracklet disconnections produced by low-confidence detections in older methods. SGT's capacity to track low-scoring and missed detections with massive top-K-scored detections improves real-time inference and MOTA performance on numerous datasets. However, this method does not involve latency sensitivity issue which has been done in the proposed paper. Pang et al. [16] highlight failure instances and recommend improvements in four components of "tracking-by-detection" 3D multi-object tracking (MOT) systems. SimpleTrack, their baseline technique, yields state-of-the-art performance on Waymo Open Dataset and nuScenes with modest tweaks. The authors advocate 3D MOT research and wonder whether standards represent real-world problems. This methodology focuses on 3D object tracking only, which is computationally expensive. The proposed paper discovers the FSL algorithm to accurately track objects in 3D environment from 2D images. Chu et al. [17] create TransMOT, a graph transformer-based video MOT method. End-to-end learning using weakly filtered detection predictions allows TransMOT to function well in complicated circumstances. However, this study does not take into account real-time tracking which is highly dependent on latency sensitivity. The methodology we suggest not only accurately detects objects, but also preserves real-time responsiveness by effectively addressing issues related to latency sensitivity.

Hu et al. [18] construct SiamMask, a real-time framework for visual object tracking and video object segmentation, reaching state-of-the-art performance at 55 frames per second. It enhances offline binary segmentation training and can cascade multiple object tracking and segmentation. However, making decision at the IoT edge from 55 frames per second significantly degrades the performance of the system. This issue has been explored and solved in the proposed methodology. Dunnhofer et al. [19] investigate 42 First Person Vision (FPV) single-item tracking methods using the TREK-150 dataset and new performance measures. Despite limitations, trackers improve FPV downstream tasks involving short-term object tracking, according to the study. This paper exhibits strong performance. However, it is not suitable for remote operation due to its sensitivity to latency. The system discussed in this paper manages to track objects with a comparable degree of accuracy, while tolerating a tenfold increase in latency sensitivity. Meimetis [20] proposes a real-time multiple-object tracking system employing a modified Deep SORT algorithm to initialize objects. YOLO detection and the framework track cars and people. Custom training YOLO on the UA-DETRAC dataset improves detection and execution performance and introduces a vehicle dataset with 7 scenes, 11,025 frames, and 25,193 bounding boxes. This approach primarily targets individuals and vehicles, thereby restricting the research's application domain. The proposed tracker can accurately track up to 1,000 distinct objects, significantly expanding its potential applications across various sectors.

Wang et al. [21] present a joint detection and association network (JDAN) for end-to-end multi-object tracking (MOT). Optimizing both submodules simultaneously streamlines MOT by eliminating complicated method design and manual tweaking. The approach develops pseudo-labels to reconcile item detection and association data. Detection findings and pseudo-association labels optimize submodules. The recommended technique outperforms previous and current approaches on two MOT challenge datasets. Ussa et al. [22] offer a real-time, hybrid neuromorphic architecture for object tracking and categorizing low-power, embedded devices. Hybrid frames and event strategies save energy and increase performance. The energy-efficient deep network (EEDN) pipeline rises from the frame-based region proposal and hardware-friendly object tracking categorization. The study reveals that the system can handle real-world monitoring circumstances without affecting performance. RGB cameras function with neuromorphic technology. Although these approaches receive commendation for accurately tracking objects, they encounter performance issues when employed in IoT devices. Object detection at the IoT edge, communicated via the internet, presents a far greater challenge compared to on-premise object detection and tracking. The proposed methodology effectively addresses and overcomes these challenges.

The potential limitations in the applications of the state-of-the-art methodologies published in recent literature are summarized in table 1. The explored methodologies have been thoroughly studied and analyzed in this research. It has been discovered that the objectives of the researchers listed in table 1 have been obtained. However, there are some common and exclusive limitations of these approaches. These limitations have been studied to discover the research gap and develop the proposed method presented in this paper.

A. RESEARCH GAP ANALYSIS AND PROPOSED SOLUTION

Real-time object tracking in the real world is a tightly coupled research field that comes immediately next to object detection and tracking in software environments. The existing literature review presented in section II and summarized in table 1 demonstrate a common pattern of innovation and improvement of object detection and tracking in software environments. The challenges of utilizing the computational intelligence of object-tracking algorithms and frameworks to control hardware and track objects in real-time with acceptable accuracy have not been thoroughly studied [23]. Although Pang et al. [16] analyzed some significant limitations, the challenges of 2D-3D coordination equivalency from 2D pixel matrix to real-world is still a research gap. Moreover, the impact of network latency is a critical performance assessment element for object trackers when applied physically. This impact isn't discernible in a computing environment due to the negligible delays within the motherboard's communication bus in the experimental setting. However, once the physical object tracker is installed and linked to the object-tracking algorithms and frameworks, the importance of latency sensitivity becomes markedly visible. This factor has been overlooked in most of the object trackers [7]. This omission constitutes a substantial research gap in real-time, real-world object-tracking studies.

The methodology presented in this paper was developed through a comprehensive analysis of the shortcomings inherent to current 3D real-time object trackers, with a particular focus on their response within physical systems. The primary objective of this research is to address and overcome the deficiencies identified during our extensive literature review, with a strong ambition to make a substantial contribution to the field of experimentation by enhancing the quality, performance, and reliability of real-time object trackers. The physical model proposed herein—detailed in Section III-A—offers a cost-effective, robust, and efficient design. A notable deficiency in most state-of-the-art studies, which we identified during our literature review, is the lack of integration of physical models—a gap our study aims to bridge. Latency, particularly within real-time video communication-based tracking, is a recurring performance bottleneck that remains largely unexplored in the existing literature. In our research, we leverage the application of Mez [14], which permits latency tolerance of up to 10x variations. Finally, we address the complex challenge of object tracking in a 3D environment from a 2D image. This hurdle is overcome through the development of our innovative Follow-Satisfy-Loop (FSL) algorithm, thereby demonstrating the potential to significantly advance the field.

III. PROPOSED METHOD

The methodology proposed in this research conducts an exhaustive experimental analysis of an object tracker, with a primary goal of identifying the system's limitations through its responses. Subsequently, the root causes of these limitations are investigated meticulously. This research methodology's focal point is the innovation of solutions to overcome these identified limitations, with the aim of developing an object tracker that is not only efficient and accurate but also capable of operating in real-time. Such an object tracker holds vast potential, with the ability to be incorporated into a multitude of applications that span various sectors, including, but not limited to, robotics, manufacturing, and healthcare. The degradation of accuracy due to latency, alongside the miscalculation of the location of

TABLE 1. Summary of the comparative literature review to discover the objectives, methods, and limitations.

Authors	Objectives	Methods	Limitation
J. Hyun et al. [15]	Increasing tracking confidence score by using SGT	Sparse Graph Tracker (SGT)	Absence of physical model integration.
P. Chu et al. [17]	Effectively tracking objects by trajectories	Spatial-Temporal Graph Transformer	The challenge of practical application in the real world has not been addressed.
Z. Pang et al. [16]	Analyzing the pros & cons of 3D multi-object tracking	Tracking-by-detection	The depth depth axis challenge has not been studied
W. Hu et al. [18]	Real-time visual object tracking and segmentation	Fully-convolutional Siamese	The latency sensitivity in real-time object tracking has been analyzed
M. Dunnhofer et al. [19]	Analysis of object tracking algorithms to discover the challenges	42 Algorithms with First Person Vision (FPV)	The feasible practical solution to the challenges have not been experimented
D. Meimetis et al. [20]	Implementation of real-time object tracker from traffic videos	Modified Deep SORT & YOLO detection methods	The research didn't analyze and solved the impact of network latency in real-time video transmission
H. Wang et al. [21]	Developing an object tracking network	Joint Detection and Association Network (JDAN)	Analysis from software context ignoring the real-world application in physical device
A. Ussa et al. [22]	Developing a real-time object tracking framework using a hybrid neuromorphic approach	TrueNorth (TN) classification via EEDN	The hardware response to the proposed tracker has not been analyzed, and the network latency has been ignored

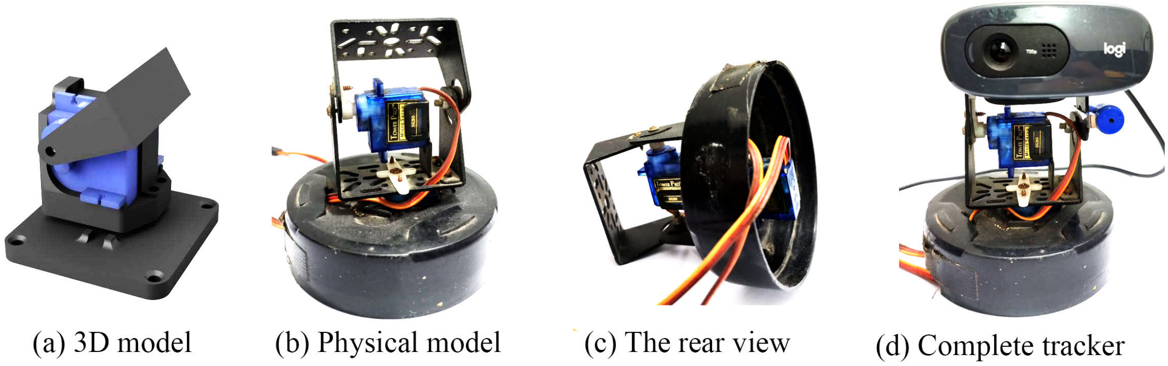

(a) 3D model
(b) Physical model
(c) The rear view
(d) Complete tracker

FIGURE 1. The 3D model and physical model of the proposed object tracker.

objects in a 3D environment based on a 2D pixel matrix, are highlighted as two principal challenges in the present system. The methodology proposed as a solution to these issues incorporates the use of Mez technology and the innovative development of the FSL algorithm. With these advancements, the system will not only be able to handle these challenges effectively but also enhance its overall tracking accuracy and efficiency.

A. OBJECT TRACKER ARCHITECTURE

Figure 1 illustrates the object tracker developed for this experiment. Figure 1(a) is the 3D model prepared before the physical implementation of the device. Because of resource constraints, the original plan was modified and implemented as shown in Figure 1(b). The rear view of the object tracker is illustrated in Figure 1(c). The USB camera has been installed, with a laser pointer affixed to the finalized model of the object tracker, as depicted in Figure 1(d). The object tracker is powered by an external power source, which is not attached to the physical model.

1) THE INFRASTRUCTURAL FRAME

The object tracker has 360-degree horizontal and 180-degree vertical freedom. That is why the proposed object tracker can track within the upper hemisphere. An aluminum pan and tilt brackets have been used to develop the infrastructural frame to gain a degree of freedom within the upper hemisphere. Figure 2 illustrates the elements of the frame and the frame itself after construction. The pan & tilt elements shown in Figure 2(a) are bolted together. After building, the complete frame illustrated in 2(b) is mounted on a plastic frame. A servo at the bottom of the plastic frame holds the entire frame and allows 360-degree horizontal movement. The tilt bracket provides support for the camera and the laser pointer. It can rotate 180 degrees vertically.

2) SERVO, CAMERA, & IoT AT THE EDGE (Mez)

The servo motor, USB camera, and IoT device connection overview are illustrated in Figure 3. The object tracker receives the optical signal using a USB web camera. The camera video stream is processed through the Machine Vision

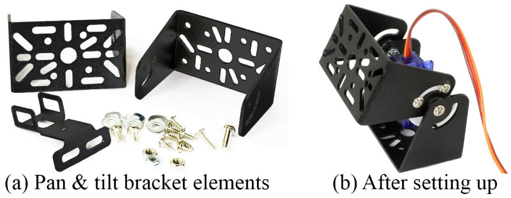


FIGURE 2. The pan & tilt brackets to build the frame.

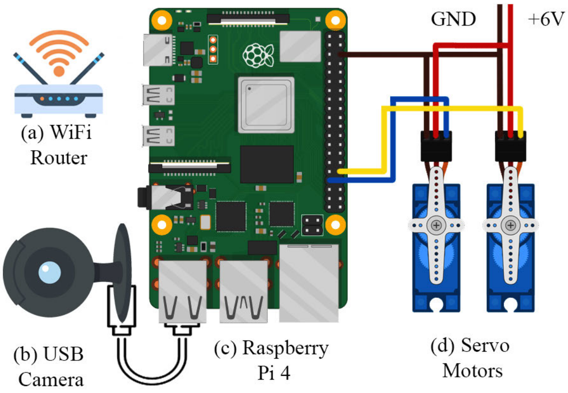


FIGURE 3. The IoT device that connections the camera and Servo motors.

at the IoT Edge (Mez) technology which has been replicated using a Raspberry Pi 4 model. The processed video is sent to the cloud over the internet. The GoogleNet, running on the cloud server, detects the object. After that, the location is tracked. The IoT device receives instructions to rotate the servo motors based on location. Finally, the servo motors are controlled by generating appropriate signals through General Purpose Input Output (GPIO) pins.

a: SERVO MOTOR

The object tracker has two TowerPro SG90 Mini Servo motors. One of these Servo motors has 180-degree freedom, and another can rotate 360 degrees. The average weight of these Servo motors is 9.02 gm. It has an operating voltage of 3.0V to 7.2V. The operating speed of these Servo motors varies depending on the input voltage. At 4.8V, it operates at 110 Rotation Per Minute (RPM). The (RPM) becomes 130 at 6V RPM. The stall torque also varies based on the operating voltage. It is 1.2kg.cm at 4.8V and 1.6kg.cm at 6.6V. An external power source of 6V is created by a parallel connection of 4 double-A batteries. It has been observed that the stall torque 1.6kg.cm at 6V.

b: THE CAMERA

The object tracker uses a C270 HD USB Webcam. The dimensions of this device, including the fixed mounting clip, are as follows: height 2.87 inches (72.91 mm), width 1.26 inches (31.91 mm), and depth 2.62 inches (66.64 mm). It also comes with a 5-foot (1.5 m) cable and weighs 2.65 ounces (75 g). In terms of technical specifications, it has a maximum resolution of 720p/30fps, a camera megapixel of 0.9, and a fixed focus lens type made of plastic. The device also has a built-in mono microphone with a range of up to 3 feet (1 m) and a diagonal field of view (dFoV) of 55°. The universal mounting clip allows for easy attachment to laptops, LCDs, or monitors, making it a versatile and practical device for video conferencing and other multimedia applications.

c: THE IoT DEVICE

The IoT device for this experiment has been constructed using a Raspberry Pi 4. It has a Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC processor that runs at a clock speed of 1.5 GHz. In this experiment, the processing power of Raspberry Pi 4 has been enhanced by using an 8GB LPDDR4-3200 SDRAM. Raspberry Pi 4 has various connectivity options, including dual-band 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless and Bluetooth 5.0. It also features Gigabit Ethernet for high-speed wired connections. The device has two USB 3.0 ports and two USB 2.0 ports, making it easy to connect external devices. We used USB 3.0 port to connect the camera and a 2.4 GHz 802.11ac wireless protocol to connect with the WiFi router. The Servo motors of the experimenting object tracker are controlled using the General Purpose Input/Output (GPIO) pins. We used a TP-Link N450 WiFi Router that uses 802.11n technology with 450Mbps bandwidth connected to the internet.

B. SYSTEM RESPONSE ANALYSIS

The methodology employed in this paper was formulated based on a detailed system response analysis. The primary objective of this paper is to construct a precise and efficient real-time object tracker. To achieve this aim, the study has adopted an unconventional yet innovative approach. Initially, a physical model of the proposed object tracker was developed and subjected to experimental testing. The methodology was subsequently shaped by comparing the system's actual response with the anticipated one, and making adjustments based on the discrepancies observed.

1) TARGET DEFINITION & CLASSIFICATION

The GoogleNet classifies 1000 categories. The proposed object tracker can track one object at a time. That is why we must specify a single class first. Even the class we specify is the target of the object tracker. The proposed object tracker can accurately track 1000 different objects. However, it is beyond the scope of any single research to experiment on all of these objects. That is why we studied the system response by taking a person as a target. This research classifies the target into three classes. They are:

- **The stationary object** (S): When the target does not move.
- **Slowly moving object** (M_s): When the target moves at less than 4.7 ft/s speed.
- **Fast-moving object** (M_f): When the target moves at more than 4.7 ft/s.

TABLE 2. The speed-accuracy relation governed by latency.

Class	Speed (ft)/s	Accuracy
S	0	1.00
M_s	1	0.42
	1.5	0.27
	2	0.24
	2.5	0.15
	3	0.09
	3.5	0.02
	4	0.01
	4.5	0.00
M_f	4.6	0.00
	4.7	0.00
	4.8	0.00
	4.9	0.00
	5	0.00
	5.1	0.00
	5.2	0.00
	5.3	0.00

According to K. Fitzpatrick et al., a human's average maximum walking speed is 4.7 ft/s. Based on the discussion presented in their paper, we considered the 4.7 ft/s as the threshold of target classification [24].

2) THE ANALYSIS

The camera and the target, illustrated in Figure 4, are calibrated properly before starting the analysis. The laser pointer center of the target is in the middle of the camera frame, and the laser is at the center as well. At this setup, the system's accuracy measured by equation 1 is 100%. We experimented from different angles. However, as long as the object is stationary, the accuracy is 100%. However, the accuracy deviates by 80% when the object is moved slowly. That means for M_s target, the accuracy is 20%. The condition worsens when the object moves faster than 4.7 ft/s. Surprisingly, the accuracy is less than for 1% for M_f class.

$$Accuracy, A(score) = (\frac{score}{10}) \times 100\% \tag{1}$$

The anomalous nature of the object tracker is worth attention. We've investigated the reasons for this drastic fall in tracking accuracy. The investigation findings have been discussed in subsection III-C.

C. FACTS FINDING & TECHNOLOGY SELECTION

The initial observation, illustrated in Figure 5 without using the Mez technology, shows that the accuracy drastically falls for moving targets. It has been tested at different speeds to determine how and why the accuracy falls. The observed data have been listed in table 2.

The empirical observation shows that the tracker tracks at the right time. However, the observations from the cloud server and physical location differ. The network latency causes this time difference in real-time [25]. There is also a sharp fall in accuracy for moving targets. Figure 5 demonstrates the relation between speed, accuracy, and latency.

The hypothesis from this observation is that the video frame size increases with the object's motion. As a result, network latency increases [26]. The GoogleNet detects and locates the object based on the video frames received by the server. However, there is a 475 ms to 2307 ms gap between the same event happening on the target location and displaying on the monitor. The frame size at different speeds is illustrated in Figure 6 supports the hypothesis.

It is evident that reducing the network latency to less than 475 ms improves the accuracy. The literature review shows that a dynamic latency-sensitive messaging system using Machine Vision at the IoT Edge (Mez) developed by A. George et al. can handle latency variations up to 10x. The Mez operates at a worst-case reduction of 4.2% of the application accuracy. The system response analysis shows that the experimenting system's performance reduces to 0% accuracy at 2.88x latency variations. Replacing the communication method with Mez can make the system tolerant of up to 10x latency variations. Considering the response analysis, facts findings, our hypothesis, and literature review, the Mez has been used as the backbone communication technology of the proposed system.

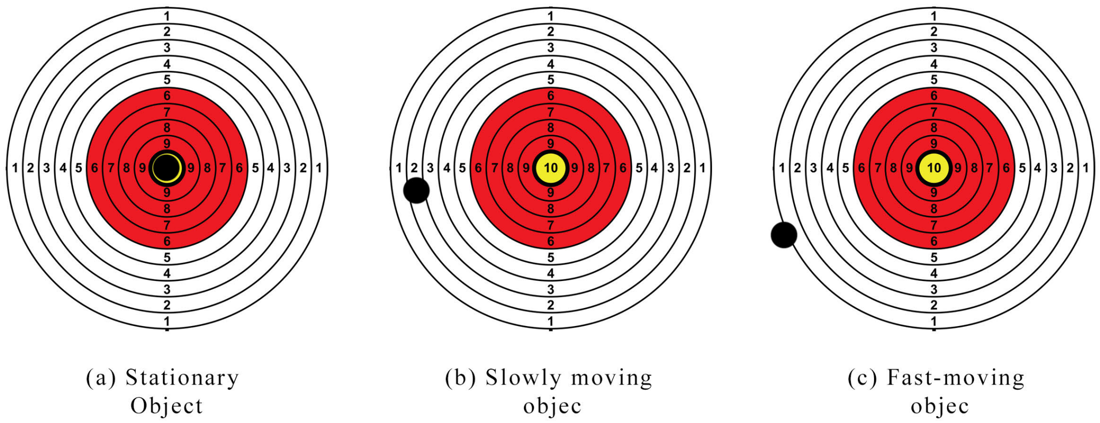
D. LATENCY CONTROLLING THROUGH MEZ

The network latency variation depends on network traffic [27]. The proposed real-time object tracker network carries only video frames. That means the size variations of the video frames are responsible for latency variations [28]. The camera used in this research has a fixed-focus camera with 0.9-megapixel sensors. It has a maximum resolution of 720p with 30 Frame Per Second (FPS). That means at the natural resolution, the video frame size is 1280 × 720.

First, we need to identify the video processing criteria to reduce the frame size without altering the overall performance. An experiment has been conducted to analyze the size of the video frames at different resolutions, and the findings have been listed in table 3. It is evident that lowering video resolution reduces the size of the video frame [29]. The imaging device used in this experiment captures the video frame in RGB (Red, Green, and Blue) colorspace. That means a single frame consists of three frames from three different channels [30]. Not every frame contains information useful in the control room. Removing useless frames before sending them to the network improves the latency [31]. Blurring video frames is another tactic to reduce the amount of information from each frame [32]. Usually, indoor videos have static backgrounds. That means information related to the background is useless in the experimental setup. Removing the background and focusing on the foreground only before sending the frames to the network improves the latency issues [33].

All possible video processing methods mentioned in the previous paragraph have been implemented in this experiment. The frame size has been resized using the equation 2.

$$(W_{new}, h_{new}) = \frac{Controller}{max(wold, h_{old})} \times (w_{old}, h_{old}) \tag{2}$$



(a) Stationary Object

(b) Slowly moving objec

(c) Fast-moving objec

FIGURE 4. The target and effect without the Mez.

TABLE 3. Frame size at different resolution and FPS.

Class	Resolution	Frame Size (KB)	30 FPS (KB)	25 FPS (KB)	20 FPS (KB)	15 FPS (KB)	10 FPS (KB)	5 FPS (KB)
s	1280x720	52.8	1584	1320	1056	792	528	264
	640x360	13.2	396	330	264	198	132	66
	320x180	3.3	99	82.5	66	49.5	33	16.5
	160x90	0.83	24.9	20.75	16.6	12.45	8.3	4.15
	80x45	0.21	6.3	5.25	4.2	3.15	2.1	1.05
Ms	1280x720	158.4	4752	3960	3168	2376	1056	792
	640x360	26.4	1188	660	792	396	264	198
	320x180	6.6	198	165	132	99	66	49.5
	160x90	1.66	49.8	41.5	49.8	24.9	16.6	8.3
	80x45	0.63	12.6	15.75	12.6	9.45	4.2	3.15
Mf	1280x720	211.2	12672	6600	9504	5544	3168	1848
	640x360	118.8	1980	2310	2112	1584	924	594
	320x180	26.4	594	742.5	396	297	132	66
	160x90	6.64	199.2	145.25	116.2	112.05	41.5	20.75
	80x45	0.84	44.1	47.25	29.4	18.9	10.5	4.2

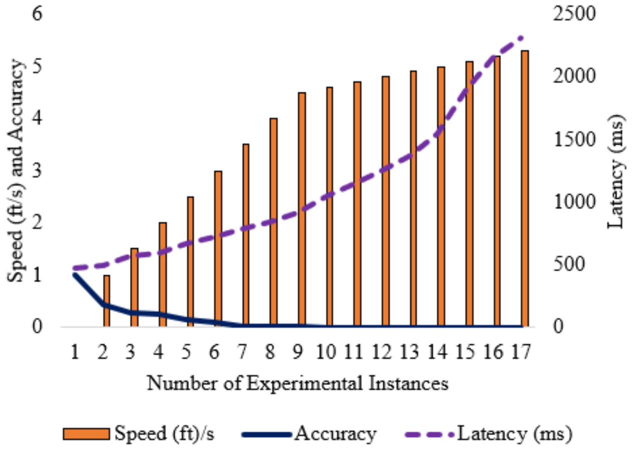


FIGURE 5. The relation among speed, accuracy, and latency without the Mez.

In equation 2, the w_{new} and h_{new} are the new width and height of the video, respectively. The w_{old} and h_{old} are the width and height before processing the frames. And the *controller* is a numeric value that controls the scaling up or scaling down the rate of the video frames.

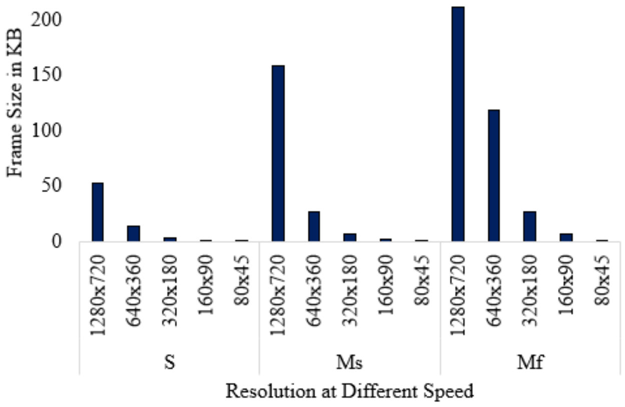


FIGURE 6. The target and effect without the Mez.

The three-channel RGB video stream is converted into a single-channel grayscale video stream using equation 3.

$$F_{gray} = 0.2126 \times F_c[0] + 0.7152 \times F_c[1] + 0.0722 \times F_c[2] \tag{3}$$

Here on equation 3, the F_{gray} is the grayscale frame converted from 3-channel RGB frame, F_c. The individual

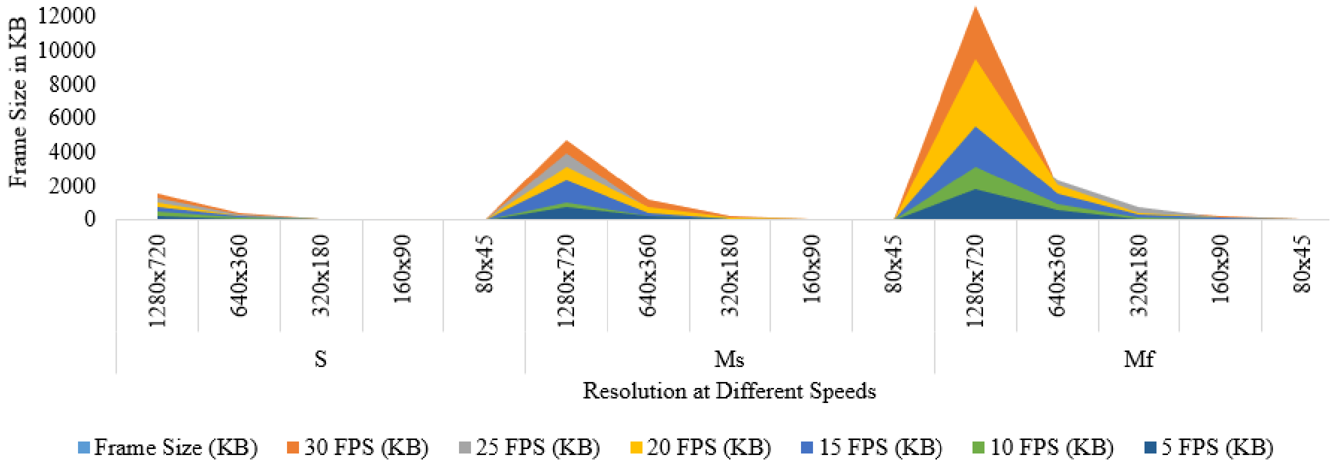


FIGURE 7. The video frame size at different speeds of the target.

channels have been fetched by array indexes 0, 1, and 2. Subsequent frame difference-based motion tracking is a simple subtraction operation in video processing. This paper has used it to remove frames carrying no information, presented in algorithm 1.

Algorithm 1 Motion-Based Frame Rejection

```
Input: Current Frame, $F_i$; Next Frame, $F_{i+1}$; Frame Difference, $F_d$
Output: Useful Frame, $F_u$;
Start
$F_d \leftarrow len(F_{i+1} - F_i)$;
if $F_d \geq MezAPI(F_d)$ then
    $F_u \leftarrow F_{i+1}$
else
    reject($F_{i+1}$)
end if
$return(F_{i+1})$
end
```

Algorithm 1 uses the Mez API to calculate the threshold to keep or reject a frame based on the tolerance level of the network. Filtering images with low-pass filters reduce the amount of information by adding sharp contours. As a result, the image size reduces. The same principle is applied to video frames as well because they are nothing but images. The Gaussian blur filter is defined by equation 4, which reduces the information on each video frame in this experiment.

$$G(x, y) = \frac{1}{2 \times \pi \times \sigma^2} \exp^{-(\frac{x^2+y^2}{2\times\sigma^2})} \tag{4}$$

Here in equation 4, x and y are the horizontal and vertical distances from the origin, respectively. The σ is the standard deviation defined by equation 5 where x_i is the current pixel value, $\bar{x}$ is the mean of all pixel values, and N is the number of pixels.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{5}$$

We have already discussed that removing static backgrounds can significantly reduce the size of the video frames. In this experiment, the Canny edge detector-based background removal approach has been used. First, all the Gaussian blur is added to the frames using equation 4. Then the intensity gradient is measured using equation 6.

$$G = \sqrt{G_x^2 + G_y^2} \tag{6}$$

After getting the intensity gradient G, we need to calculate the direction angles. It has been done using equation 7.

$$\Theta = atan2(G_x, G_y) \tag{7}$$

Once the gradient and direction are calculated, the gradient magnitude threshold is applied to avoid false responses to edge detection. We used a double threshold to determine the potential edges. After that, we applied the rate-dependent hysteresis defined by equation 8 to track the edges.

$$Y(t) = \chi_i \times X(t) + \int_a^b \Phi_d X(t-\tau)\, d\tau \tag{8}$$

Here on equation 8, χ_i is the response to $i^t h$ element, $\Phi_d \tau$ is the impulse response, and τ is the time unit in the past. The $X(t) = X_0 sin(\omega t)$ and $Y(t) = Y_0 sin(\omega t - \varphi)$. Finally, the fully connected edges are used as the mask to remove the background and keep the foreground on the frames.

1) MEZ's APPLICATION IN THE TRACKER

The video frames in Mez are the key-value(t, f) pair, the keys are the timestamps, and the values are the video frames. The original sequences of the video frame are preserved in it. This technology aims to reduce the bandwidth required to transmit video frames. The Mez architecture consists of three components. They are the message broker, memory log, and latency controller. In the proposed real-time object tracker, the message broker has been used to establish communication between the tracker and the cloud server. The memory log stores the video frames. The latency controller adjusts the video frames' quality to control the optimal video frame transmission latency. The video frames generated from the IoT cameras are temporarily stored in the memory log. The latency controller, located on the IoT camera nodes, modifies the video frames stored on the memory log according to the available bandwidth of the transmission medium [14].

E. FOLLOW-SATISFY-LOOP (FSL) ALGORITHM

The FLS algorithm operates differently from the norm; rather than directing servo motors towards specific (x_c, y_c) coordinates, it seeks the Region of Interest (ROI) center and tracks this point. This approach means that the Z-axis becomes superfluous, as the system relies on determining the distance from the ROI's center. By rotating the servo motors based on this measurement, the algorithm ensures the ROI's center aligns with the 2D sensor array's center. This methodology enhances the FLS algorithm's precision.

1) BACKGROUND OF THE FSL ALGORITHM

The Follow-Satisfy-Loop (FSL) algorithm is one of the novel contributions of this algorithm that receives predictions from GoogleNet. The GoogleNet detects the target object from a 2D frame. A bounding box is drawn around the object, the Region of Interest (ROI). The center of the ROI is the target point. The ROI's top left and bottom right coordinate is x_1, y_1 and x_2, y_2, respectively. The center of the ROI is calculated using equation 9.

$$(x_c, y_c) = (\frac{(x_1 + x_2)}{2}, \frac{(y_1, y_2)}{2}) \tag{9}$$

The (x_c, y_c) in equation 9 is the Cartesian coordinate of the target. However, these values represent the pixel location. Depending on the distance of the object from the camera, the target coordinate change even if $\frac{ds}{dx}$ and $\frac{ds}{dy}$ are zero where s is the distance. As a result, the random variations in (x_c, y_c) make the tracker unstable. The failure of the primary attempt at 2D-Cartesian coordinate-based tracking motivated the researchers of this paper to convert the Cartesian coordinate to Polar form using the equation 10.

$$T(r, \theta) = \sqrt{x^2 + y^2} \angle (tahn^{-1} \frac{y}{x}) = r \angle \theta \tag{10}$$

Theoretically, the tracker should be stable on the target when the servo motors are shifted according to the correct value of θ. However, r of equation 10 is calculated from the 2D-Cartesian coordinate system, whereas it is the z-axis of the 3D plane. As a result, the tracker again fails to track the object correctly. The failed attempts to accurately track the objects in real-time using equations 9 and 10 motivated the development of the Follow-Satisfy-Loop (FSL) algorithm.

2) PSEUDOCODE OF FSL ALGORITHM

The FSL algorithm, presented in algorithm 2, keeps the target in the center of the frame. The center of a frame with 640 × 360 is the pixel located at $(320 \times 180)^{th}$ location. If the object is on the right side of the center, the horizontal servo keeps moving counterclockwise until the object is at the center. The horizontal servo keeps rotating clockwise until the object falls at the center of the frame. The vertical servo motor rotates clockwise if the object is higher than the center and rotates counterclockwise if the object is lower than the center. The algorithm is satisfied when the object is at the center and the loop stops. Otherwise, the loop continues, and the tracker keeps following the object.

Algorithm 2 The FSL Algorithm

```
Input: Tracker Video Stream, $v_s$; HLS Request, $H_l$; Target Object, $T_{obj}$; Frame Center $(x_f, y_f)$
Initiate: Allocate Cloud Resource; Initiate GoogleNet, $G_{net}$
Output: Horizontal Servo Rotation, $H_s$; Vertical Servo Rotation, $V_s$
Start
$k \leftarrow 0$
$F[k] \leftarrow read(v_s)$
while $v_s = True$ do
    $i \leftarrow i + 1$
    Accept HLS Request
    $F[i] \leftarrow read(v_s)$
    $d \leftarrow G_{net}(F[i])$
    if $d == T_{obj}$ then
        $[x_c, y_c] \leftarrow Center(ROI(G_{net}(F[i]))$
        if $x_c > x_f$ then
            $H_s(+1)$
        end if
        if $x_c < x_f$ then
            $H_s(-1)$
        end if
        if $y_c > y_f$ then
            $V_s(-1)$
        end if
        if $y_c < y_f$ then
            $V_s(+1)$
        end if
    end if
end while
end
```

IV. EXPERIMENTAL RESULTS AND EVALUATION

This section provides a detailed analysis of the experimental results and performance assessment of the proposed tracking system, Trackez. The system demonstrates accurate object tracking capabilities up to a speed of 5.5 ft/s. However, it is observed that the tracking accuracy drops sharply when the speed surpasses this limit. A similar trend is noted with regard to latency, which also tends to increase beyond this speed.

TABLE 4. The stable and unstable range.

Attempt	Stable Range	Unstable Range	Frame Size
1	0 - 26.08	26.08 - 40	640 × 360
2	0 - 25.0	25.0 - 40	640 × 360
3	0 - 25.4	25.4 - 40	640 × 360
4	0 - 25.6	25.6 - 40	640 × 360
5	0 - 25.6	25.6 - 40	640 × 360

On the other hand, the Mez tracking system maintains stable latency even when handling speeds exceeding 5.5 ft/s.

A. EXPERIMENTAL SETUP

It has been observed that the laser used in this experiment has a stable range of up to 25 feet, illustrated in Figure 8. The experiment was conducted five times to identify the stable range. That is why Figure 8 shows five straight lines within the range. After this range, the laser pointer vibrates rapidly. At the same time, the size of the object on the camera becomes smaller. As a result, the vibrating later pointer deviates from the target. That is why the experiment was performed in a 40 feet long room. The table 4 lists the stable and unstable regions for the experimenting object tracker.

The tracking range of object trackers has always been a challenge in this research domain. The proposed system is no different. When the object is in the unstable region, the software system can track the location of the object. However, the physical device has its limitation imposed by the minimum rotation angle of the servo motor. As a result, the laser pointer keeps vibrating around the object. It is considered as a constraint of the physical device instead of the limitation of the proposed system.

The experimental setup is 40 feet long and 15 feet wide in a room with white walls. Three people wearing red (R), green (G), and blue (B) T-shirts are moving in non-regular patterns and at random speeds. The experimental setup is illustrated in Figure 9. The trackers were randomly stationed in four different locations marked on the Figure 9 by (G_1, G_2, G_3, G_4) where G represents the ground. The targets move randomly within the range of 15 to 24 feet. The speed range is 0 to 5.5 ft/s.

The Trackez aims to track moving and stationary objects using a camera connected to the IoT edge. The experiment illustrated in Figure 9 demonstrate that the Trackez is excellent at tracking moving object which is the primary objective of the paper. The increased network latency, caused by the enhancement of frame size due to motion, is effectively mitigated in the proposed system. This signifies that Trackez has successfully accomplished its research objectives.

B. EXPERIMENTAL OBSERVATION VARIABLES (EOV)

We have analyzed the before and after effects of using the Mez technology regarding system response. It has been discovered from the system response analysis that the accuracy of the object tracker is affected by network latency. The purpose of incorporating the Mez technology is to reduce the latency to increase the system's accuracy.

TABLE 5. Description of the EOVs.

EOV	Symbol	Description
Accuracy Improvement	A_{im}	This variable compares the difference between accuracy of the system before and after using the Mez technology
Latency Reduction	Lr	This is another comparing variable expresses the difference between latencies before and after using the Mez
Speed Association	S_a	It highlights the association of accuracy and speed

It is also evident that the target's speed is associated with the experimenting system's overall performance. According to these observations, we have considered the target's accuracy improvement, latency reduction, and speed as the experimental observation variables. The descriptions of these variables are listed in table 5.

C. OVERALL PERFORMANCE ANALYSIS

The performance of the Trackez is listed in table 6. The experimental results demonstrate an average accuracy improvement of 81.71% with an average of 85.5% latency reduction. The experimenting system has no performance issue at $Speed = 0\ ft/s$. That is why there is no difference between the accuracy before and after applying the Machine Vision at the IoT Edge (Mez). However, in other cases, when $Speed > 0\ ft/s$, the accuracy drastically falls, which is illustrated in Figure 10. The accuracy drops to 0% from 40% when the speed crosses 3.5ft/s.

1) ACCURACY IMPROVEMENT ANALYSIS

After using the Mez, the system's accuracy increases rapidly, which is illustrated in 11. It is observed that the Mez is also affected by the speed variation. When the speed of the target increases, the accuracy falls. However, the fall is gradual, with an insignificant slope. The difference between the accuracy before and after applying Mez is significant on 11. The drastic fall in accuracy, illustrated in Figure 10, makes any object tracker impractical. It is optional to maintain a 100% accuracy. However, maintaining consistency is essential. Figure 11 shows that the accuracy does not fall instantly after using the Mez. Instead, it maintains a gradual change with a small slope. The linear nature is also observable in Figure 11. The comparison between the fall of accuracy before and after using the Mez demonstrates the stability in performance improvement of the proposed real-time object tracker evaluated from the network latency perspective.

Performance degradation and stability are two of the issues this research deals with. The system response illustrated in Figure 11 proves the stable nature of the system at varying speeds of the target. The overall performance improvement in accuracy is illustrated in Figure 12. It has been observed that the accuracy increases significantly in the M_s region. This implies this region is more sensitive to speed than the other two regions. The s region demonstrates insignificant

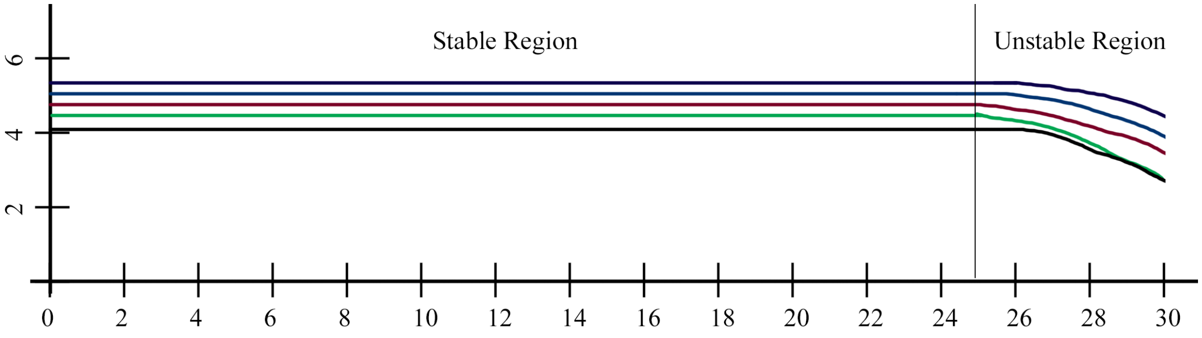


FIGURE 8. The stable range of the laser on the video frame.

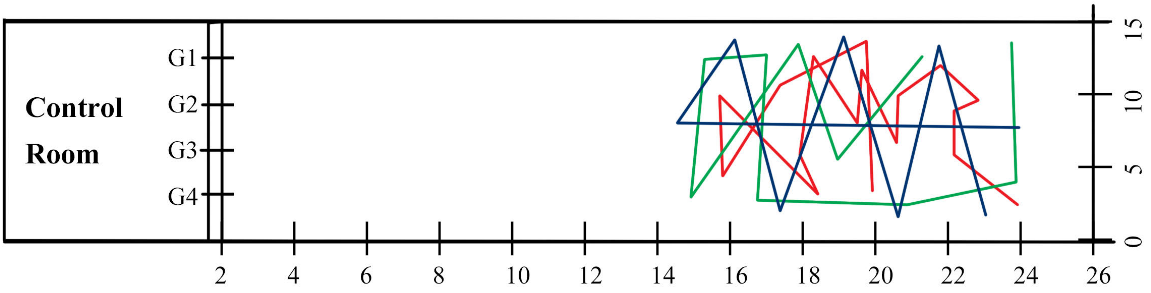


FIGURE 9. The target movement in experimental setup.

TABLE 6. The overall result of the experiment.

Class	Speed (ft)/s	Accuracy	Improved Accuracy	A_{im} (%)	Latency (ms)	Mez Latency (ms)	Lr (%)
S	0	1	1.00	0.00	475	66.50	86.00
8*Ms	1	0.42	1.00	57.97	497	61.63	87.60
	1.5	0.27	0.99	72.01	566	70.47	87.55
	2	0.24	0.99	74.87	591	74.41	87.41
	2.5	0.15	0.98	82.94	667	86.18	87.08
	3	0.09	0.97	88.04	722	93.14	87.10
	3.5	0.02	0.96	94.14	790	103.51	86.90
	4	0.0089	0.96	94.89	840	112.73	86.58
	4.5	0.0046	0.94	93.61	912	125.45	86.24
8*Mf	4.6	0.003	0.94	93.69	1046	140.82	86.54
	4.7	0.0022	0.93	92.98	1140	170.77	85.02
	4.8	0.0009	0.92	92.38	1246	197.20	84.17
	4.9	0.0002	0.91	91.00	1368	219.45	83.96
	5	0	0.91	90.71	1541	276.16	82.08
	5.1	0	0.90	90.38	1894	345.28	81.77
	5.2	0	0.90	90.04	2145	409.76	80.90
	5.3	0	0.89	89.44	2307	472.52	79.52

behavior. And the rate of change of improvement in M_f is nominal. The improvement curve demonstrates near-linear characteristics. And the overall accuracy improvement is 81.71%. The initial analysis presented in this paper hypothesized that the accuracy degradation when the speed of the target increases is caused by network latency. The application of Mez reduces the latency sensitivity. As a result, the system's accuracy improves rapidly, especially in the M_s region. The accuracy becomes stable in the M_f region. From the accuracy and stability perspective, the proposed real-time object tracker using Mez exhibits acceptable performance.

2) LATENCY REDUCTION ANALYSIS

Prior to implementing Mez technology, the average latency stood at 1102.76 ms. This latency was the principal cause of the decrease in accuracy observed within the tracking system. However, upon the application of Mez technology, there was a significant drop in average latency to 178.00 ms, which translates into an average reduction of 85.08%. The latency for s class is 475 ms without the Mez. It is the lowest latency of the system. At the highest value of speed in M_f, the latency is 2307 ms. After using the Mez, the latency range is 66.50 ms to 472.52 ms. The latency reduction analysis is illustrated in Figure 13, demonstrating the significant latency reduction.

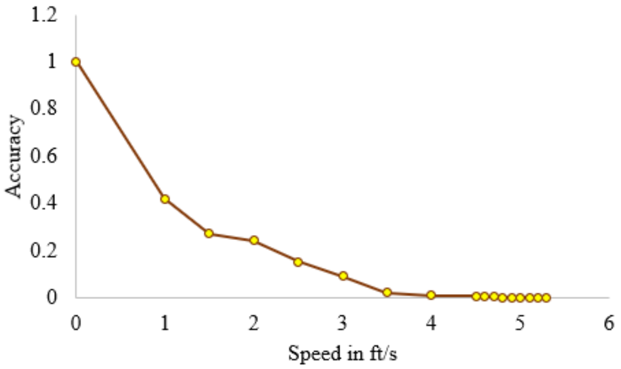


FIGURE 10. Fall of accuracy.

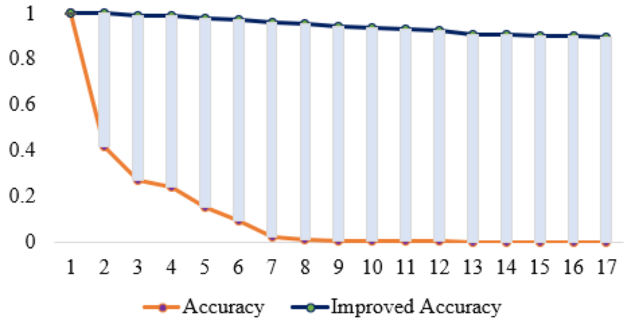


FIGURE 11. Increase of system's accuracy.

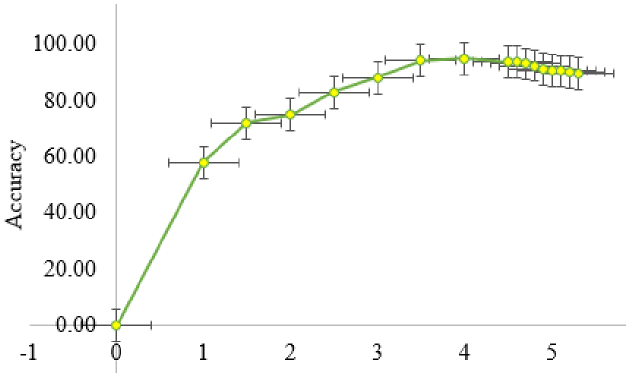


FIGURE 12. The accuracy improvement analysis.

The latency reduction has been further analyzed in the percentage range in Figure 14. One of the primary performance limiting factors of the system is high latency in the M_f region. It has been observed that the Mez rapidly reduces the latency of the M_f region.

The experimental result demonstrates that the Trackez accurately tracks the object up to a speed range of 5.4 ft/s. Within this speed limit, the system is scalable. Moreover, GoogleNet minimizes the images by repeated convolution. As a result, even large-scale objects are accurately tracked by the proposed tracker.

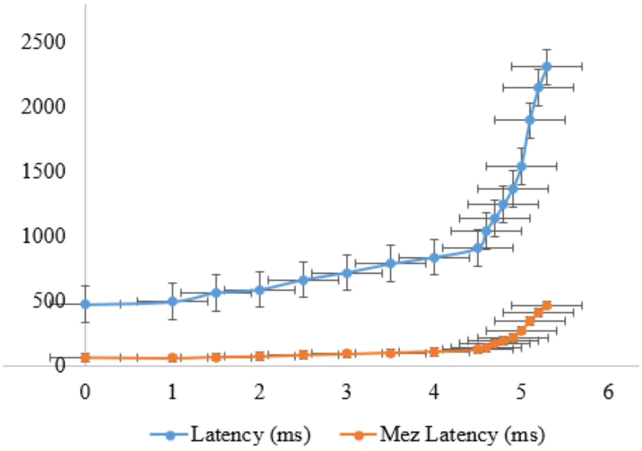


FIGURE 13. Latency comparison and analysis.

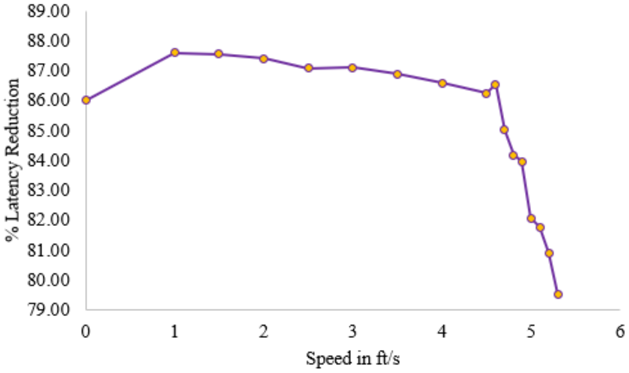


FIGURE 14. The latency reduction analysis in percentage.

V. LIMITATION AND FUTURE WORK

Although the proposed object tracker overcomes the challenge of tracking objects in a 3D environment from the 2D-pixel matrix, it is not immune to several limitations.

A. ACTUAL vs. EXPERIMENTAL SETUP

The proposed system has been experimented with in an experimental environment. There are differences between the actual and experimental environments. The object trackers have been designed for robotics applications with possibilities for vibration, momentum, and external adversarial impacts. These affect the system's accuracy, which was not studied during the experiment. An ongoing experiment, an extension of this study, analyzes these impacts, which will be published in subsequent papers.

B. IMPRACTICAL EFFECTIVE RANGE

This experiment uses a short-range laser to track the object. The stability at the beginning and end of the laser is different. The longer the range, the more sensitive it becomes to movement. The experimental setup is 24 feet long. The data presented in this paper are collected from within this range. That means the effect of the tracker in the long-range is unexplored. It is a major limitation of the proposed system. However, this paper's researcher considers it an opportunity to extend the study and analyze the performance of the tracker in a longer range.

C. CAMERA MOUNTED ON THE SYSTEM

The proposed object tracker has a camera mounted on the actuator. It tracks the object by keeping it at the center of the camera frame. When the tracker tracks the object that is at the maximum range of the tracker, there is a maximum blind spot on the other side. The solution to this problem is to keep the camera stable and move the tracker only. However, the proposed FSL algorithm works only for camera-mounted object trackers. It is a major limitation of the proposed system. However, it creates opportunities to conduct more experiments and discover an efficient and effective way to keep the camera stationary while tracking objects accurately.

D. CLOUD RESOURCE OPTIMIZATION

The proposed object tracker uses GoogleNet to recognize objects which run from a cloud server. Undoubtedly, GoogleNet is one of the best-performing CNN for object recognition. However, it is a computationally expensive network. There are scopes of developing and using lightweight CNNs to minimize cloud resources. However, it has not been addressed in this paper. The subsequent research conducted in this domain will address cloud resource optimization, which is the future scope of this existing research.

E. PARTIAL VIEW PROBLEM

The proposed Trackez uses the pre-trained GoogleNet that recognizes 1,000 different objects. However, it has some limitations. It sometimes misclassifies objects when there is a partial view of the object is available. As a result, Trackez inherits this limitation. It fails to track objects which GoogleNet fails to recognize.

VI. CONCLUSION

The primary challenge of developing a real-time object tracker is the time delay in real-time video feed transmission. Timing and accuracy are the two most essential factors in automatic object tracking. However, latency caused by traffic-intensive video feed from HD cameras mounted on the tracker becomes the primary impediment to the practical implementation of the system. This impediment has been effectively overcome by changing the communication technology between the object tracker and the cloud server. The adaptive distributed messaging system developed by George et al., capable of adjusting the video quality without altering the application level accuracy threshold, has been used. It is a Machine Vision at the IoT Edge (Mez) technology that reduces the latency by 85.08%. The Mez increases the overall accuracy of the proposed real-time object tracker by 81.71%.

Trackez presents a promising solution for latency-sensitive applications that rely on object tracking. Conventionally, the measurement of the distance between an object and the camera, and the utilization of this depth value for precise object tracking necessitates extra equipment, which invariably leads to additional costs. However, Trackez eliminates the need to determine the distance between an object and the camera for detection purposes. Consequently, it not only diminishes the cost of equipment but also reduces the computational burden on the processor. Trackez has propelled object-tracking research to an advanced stage where it is no longer restricted to purely software-based solutions. This full-fledged object tracker operates using a physical device in real time, adeptly managing latency issues encountered during real-time object tracking.

The world is becoming interconnected faster than ever. More than 63.1% of the global population uses the internet [34]. As a matter of fact, the Internet of Things (IoT) market has grown from \$326.9 billion in 2021 to \$396.34 billion in 2022. That means there is a 21.2% compound annual growth rate (CAGR) in this sector [35]. Along with humans, devices are interconnected. It increases the demand for ultra-real-time communication with minimal latency. The remarkable improvement in the real-time object tracker through Mez shows it has significant potential in latency-sensitive communication in image processing and Machine Learning-based technologies such as real-time surveillance, drone control, virtual reality, video conferencing, video calls, Robot Operating Systems (ROSs), space exploration, autonomous vehicle, ocean exploration, and so on.

However, the Trackez is not beyond limitations. It has been experimented with in an experimental setting with uniform background. The practical application of it may not always have uniform background and the effects of this possibility have not been addressed in this study. Moreover, the range of the Trackez is limited to 24 feet. Additional computation, for example, digital or optical zoom, is necessary for long-distance tracking. It has not been explored in this paper. The Trackez uses a camera mounter device architecture where the camera moves with the laser pointer. The momentum and additional jerking are potential threats to the stability of the tracker. No measurement has been taken to fix this issue. In addition, the partial view problem has not been addressed in the proposed system.

These limitations are the future scope of conducting more research to further improve the performance of Trackez and make it a more reliable system. The practical experimental analysis as a part of the robotics system is a potential application of Trackez which will be explored in the future. The digital zooming-based range enhancement of the Trackez is another potential future research scope. There are opportunities to further improve the system architecture by separating the camera from the actuator. It ensures more stability and overcomes momentum and shaking problems. Estimating objects from a partial view is another challenging future research scope on Trackez. The current version of the Trackez is an accurate, efficient, and effective object tracker.

The subsequent versions which will incorporate the potential improvements will be better, faster, and more application friendly.

REFERENCES

[1] L. P. O. Paula, N. Faruqui, I. Mahmud, M. Whaiduzzaman, E. C. Hawkinson, and S. Trivedi, "A novel front door security (FDS) algorithm using GoogleNet-BiLSTM hybridization," *IEEE Access*, vol. 11, pp. 19122–19134, 2023.

[2] P. Chakraborty, M. A. Yousuf, M. Z. Rahman, and N. Faruqui, "How can a robot calculate the level of visual focus of human's attention," in *Proc. Int. Joint Conf. Comput. Intell., (IJCCI)* Cham, Switzerland: Springer, 2020, pp. 329–342.

[3] N. Faruqui, M. A. Yousuf, M. Whaiduzzaman, A. K. M. Azad, A. Barros, and M. A. Moni, "LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data," *Comput. Biol. Med.*, vol. 139, Dec. 2021, Art. no. 104961.

[4] Y. Lee and B. You, "Free space detection algorithm using object tracking for autonomous vehicles," *Sensors*, vol. 22, no. 1, p. 315, Dec. 2021.

[5] A. Cioppa, S. Giancola, A. Deliège, L. Kang, X. Zhou, Z. Cheng, B. Ghanem, and M. Van Droogenbroeck, "SoccerNet-tracking: Multiple object tracking dataset and benchmark in soccer videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3490–3501.

[6] J. S. Devagiri, S. Paheding, Q. Niyaz, X. Yang, and S. Smith, "Augmented reality and artificial intelligence in industry: Trends, tools, and future challenges," *Expert Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 118002.

[7] A. George and A. Ravindran, "Scalable approximate computing techniques for latency and bandwidth constrained IoT edge," in *International Summit Smart City 360°*. Cham, Switzerland: Springer, 2021, pp. 274–292.

[8] Y. Liu and W. He, "Signal detection and identification in an optical camera communication system in moving state," *J. Phys., Conf. Ser.*, vol. 1873, no. 1, Apr. 2021, Art. no. 012015.

[9] S. N. Jyothi and K. V. Vardhan, "Design and implementation of real time security surveillance system using IoT," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Oct. 2016, pp. 1–5.

[10] A. George and A. Ravindran, "Distributed middleware for edge vision systems," in *Proc. IEEE 16th Int. Conf. Smart Cities, Improving Quality Life Using ICT IoT AI (HONET-ICT)*, Oct. 2019, pp. 193–194.

[11] M. S. John, M. B. Cowen, H. S. Smallman, and H. M. Oonk, "The use of 2D and 3D displays for shape-understanding versus relative-position tasks," *Human Factors, J. Hum. Factors Ergonom. Soc.*, vol. 43, no. 1, pp. 79–98, Mar. 2001.

[12] A. George, "Distributed messaging system for the IoT edge," Ph.D. thesis, Dept. Elect. Comput. Eng., Univ. North Carolina Charlotte, Charlotte, NC, USA, 2020.

[13] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103514.

[14] A. George, A. Ravindran, M. Mendieta, and H. Tabkhi, "Mez: An adaptive messaging system for latency-sensitive multi-camera machine vision at the IoT edge," *IEEE Access*, vol. 9, pp. 21457–21473, 2021.

[15] J. Hyun, M. Kang, D. Wee, and D. Yeung, "Detection recovery in online multi-object tracking with sparse graph tracker," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4839–4848.

[16] Z. Pang, Z. Li, and N. Wang, "SimpleTrack: Understanding and rethinking 3D multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 680–696.

[17] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "TransMOT: Spatial–temporal graph transformer for multiple object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4859–4869.

[18] W. Hu, Q. Wang, L. Zhang, L. Bertinetto, and P. H. S. Torr, "SiamMask: A framework for fast online object tracking and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3072–3089, Mar. 2023.

[19] M. Dunnhofer, A. Furnari, G. M. Farinella, and C. Micheloni, "Visual object tracking in first person vision," *Int. J. Comput. Vis.*, vol. 131, no. 1, pp. 259–283, Jan. 2023.

[20] D. Meimetis, I. Daramouskas, I. Perikos, and I. Hatzilygeroudis, "Real-time multiple object tracking using deep learning methods," *Neural Comput. Appl.*, vol. 35, no. 1, pp. 89–118, Jan. 2023.

[21] H. Wang, X. He, Z. Li, J. Yuan, and S. Li, "JDAN: Joint detection and association network for real-time online multi-object tracking," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 19, no. 1s, pp. 1–17, Feb. 2023.

[22] A. Ussa, C. S. Rajen, T. Pulluri, D. Singla, J. Acharya, G. F. Chuanrong, A. Basu, and B. Ramesh, "A hybrid neuromorphic object tracking and classification framework for real-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 17, 2023, doi: 10.1109/TNNLS.2023.3243679.

[23] M. Mendieta, C. Neff, D. Lingerfelt, C. Beam, A. George, S. Rogers, A. Ravindran, and H. Tabkhi, "A novel application/infrastructure co-design approach for real-time edge video analytics," in *Proc. SoutheastCon*, Apr. 2019, pp. 1–7.

[24] K. Fitzpatrick, M. A. Brewer, and S. Turner, "Another look at pedestrian walking speed," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1982, no. 1, pp. 21–29, Jan. 2006.

[25] S. Wu, L. Hou, G. Zhang, and H. Chen, "Real-time mixed reality-based visual warning for construction workforce safety," *Autom. Construct.*, vol. 139, Jul. 2022, Art. no. 104252.

[26] W. G. de Morais, C. E. M. Santos, and C. M. Pedroso, "Application of active queue management for real-time adaptive video streaming," *Telecommun. Syst.*, vol. 79, no. 2, pp. 261–270, Feb. 2022.

[27] A. George and A. Ravindran, "Latency control for distributed machine vision at the edge through approximate computing," in *Proc. Int. Conf. Edge Comput.* Cham, Switzerland: Springer, 2019, pp. 16–30.

[28] J. Wu, C. Yuen, and J. Chen, "Leveraging the delay-friendliness of TCP with FEC coding in real-time video communication," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3584–3599, Oct. 2015.

[29] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Trans. Comput. Imag.*, vol. 2, no. 2, pp. 109–122, Jun. 2016.

[30] S. Süsstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces," in *Proc. Color Imag. Conf.* Springfield, VA, USA: Society for Imaging Science and Technology, 1999, pp. 127–134.

[31] G. Cena, S. Scanzio, and A. Valenzano, "Seamless link-level redundancy to improve reliability of industrial Wi-Fi networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 608–620, Apr. 2016.

[32] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1954–1963.

[33] Z. Fu, Y. Chen, H. Yong, R. Jiang, L. Zhang, and X. Hua, "Foreground gating and background refining network for surveillance object detection," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6077–6090, Dec. 2019.

[34] B. Adeyeye, S. E. Ojih, D. Bello, E. Adesina, D. Yartey, C. Ben-Enukora, and Q. Adeyeye, "Online learning platforms and covenant university students' academic performance in practical related courses during COVID-19 pandemic," *Sustainability*, vol. 14, no. 2, p. 878, Jan. 2022.

[35] D. Rai, S. S. Rajput, and R. Kumar, "Attacks, security concerns, solutions, and market trends for IoT," in *5G Beyond*. Boca Raton, FL, USA: CRC Press, 2022, pp. 13–28.

NURUZZAMAN FARUQUI received the B.Sc. degree in electrical and electronics engineering (EEE) from North South University, Bangladesh, in 2016, and the master's degree in information technology from the Institute of Information Technology (IIT), Jahangirnagar University (JU), Bangladesh, in 2018, with a 4/4 CGPA.

He is currently a Senior Lecturer with the Department of Software Engineering (SWE), Daffodil International University, Bangladesh. He is also a research coordinator of the department. He is globally recognized for his educational video content on neural networks using MATLAB. He is a YouTuber and the author. He has authored three books. His current research interests include artificial intelligence, machine learning, deep learning, cloud computing, and image processing.

Mr. Nuruzzaman is a member of the Institution of Engineers, Bangladesh (IEB), and the Bangladesh Society for Private University Academics (BSPUA).

MD ALAMGIR KABIR (Member, IEEE) received the master's degree in software engineering from Wuhan University, China, in 2017, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2021. He is currently a Postdoctoral Researcher with the Artificial Intelligence and Intelligent Systems Research Group, Mälardalen University, Sweden. He is an inquisitive individual, driven to develop novel techniques that facilitate efficient, explainable, and robust AI. During his graduate studies, he conducted research on various topics in software engineering, including software testing, verification and validation, and software metrics. He has published several research papers in top-tier conferences and journals in his field. His current research interests include developing new techniques for artificial intelligence and intelligent systems.

MOHAMMAD ABU YOUSUF received the B.Sc.(Eng.) degree in computer science and engineering from the Shahjalal University of Science and Technology, Sylhet, Bangladesh, in 1999, the M.Eng. degree in biomedical engineering from Kyung Hee University, South Korea, in 2009, and the Ph.D. degree in science and engineering from Saitama University, Japan, in 2013. In 2003, he joined as a Lecturer with the Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh. In 2014, he was with the Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, Bangladesh, where he is currently a Professor. His current research interests include medical image processing, human–robot interaction, computer vision, and natural language processing.

MD. WHAIDUZZAMAN (Senior Member, IEEE) received the bachelor's degree in electronics and computer science and the M.Sc. degree in telecommunication and computer network engineering from London, U.K., and the Ph.D. degree from the University of Malaya, Malaysia.

He serves as a Professor with the Institute of Information Technology (IIT), Jahangirnagar University. Currently, he is working on ARC-funded projects with the Queensland University of Technology, Australia. His current research interests include mobile cloud computing, vehicular cloud computing, fog computing, the IoT, and microservices.

Dr. Whaiduzzaman received the *Journal of Network and Computer Applications* (Elsevier) Best Paper Award in Paris, France.

ALISTAIR BARROS is currently a Professor of information systems and the Head of the Services Computing Program, School of Information Systems, Queensland University of Technology. He has 32 years of ICT experience across industry, industrial research and development, and academic roles, including the Global Research Leader and the Chief Development Architect with SAP AG. His current research interests include cloud, enterprise systems and microservices engineering, evolution, and provisioning using model-based techniques.

IMRAN MAHMUD received the master's degree in software engineering from the University of Hertfordshire, U.K., in 2008, and the Ph.D. degree in technology management from Universiti Sains Malaysia, in 2017.

He is currently the Head and an Associate Professor with the Department of Software Engineering, Daffodil International University, Bangladesh. He is also a Visiting Professor with the Graduate School of Business, Universiti Sains Malaysia. He is also with Technostress, where he is focusing on game addiction and online learning continuance intention. Previously, he was a Senior Lecturer with the Graduate School of Business, Universiti Sains Malaysia. He was a Visiting Lecturer with the Institute of Technology, Bandung, Indonesia, and the Hong Kong Management Association, Hong Kong.

Dr. Mahmud achieved several awards, including the Hall of Fame and Prestigious Publication Award from Universiti Sains Malaysia, the Young Researcher from Kasetsart University, Thailand, and the Young Scientist in Technology Management from the Venus International Foundation, India.