MethodsX 14 (2025) 103291

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

Innovative IoT-enabled mask detection system: A hybrid deep learning approach for public health applications ☆

Parul Dubey [a,*], Vinay Keswani [b], Pushkar Dubey [c], Gunjan Keswani [d], Dhananjay Bhagat [e]

[a] Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India
[b] Department of Electronics and Telecommunication Engineering, G H Raisoni College of Engineering, Nagpur, India
[c] Department of Management, Pandit Sundarlal Sharma (Open) University, Chhattisgarh, India
[d] Department of Computer Science & Engineering and Emerging Technologies, School of Computer Science and Engineering, Ramdeobaba University, Nagpur, India
[e] Dr. Vishwanath Karad MIT World Peace University, Pune, India

ARTICLE INFO

Method name:
Adaptive Flame-Sailfish Optimization (AFSO)

Keywords:
Face mask detection
Internet of things
Deep learning
Adaptive optimization
Real-time health monitoring
Hybrid framework

ABSTRACT

The integration of IoT and deep learning has revolutionized real-time monitoring systems, particularly in public health applications such as face mask detection. With increasing public reliance on these technologies, robust and efficient frameworks are critical for ensuring compliance with health measures. Existing models, on the other hand, often have problems, such as being slow to compute, not being able to work well in a wide range of environments, and not being able to adapt well to IoT devices with limited resources. These shortcomings highlight the need for an optimized and scalable solution. To address these issues, this study utilizes three datasets: the Kaggle Face Mask Dataset, the Public Places Dataset, and the Public Videos Dataset, encompassing varied environmental conditions and use cases. The proposed framework integrates ResNet50 and MobileNetV2 architectures, optimized using the Adaptive Flame-Sailfish Optimization (AFSO) algorithm. This hybrid approach enhances detection accuracy and computational efficiency, making it suitable for real-time deployment. The novelty of the paper lies in combining AFSO with a hybrid deep learning architecture for parameter optimization and improved scalability. Performance metrics such as accuracy, sensitivity, precision, and F1-score were used to evaluate the model. The proposed framework achieved an accuracy of 97.8 % on the Kaggle dataset, significantly outperforming baseline models and demonstrating its robustness and efficiency for IoT-enabled face mask detection systems.

- The article introduces a novel hybrid framework that combines ResNet50 and MobileNetV2 architectures optimized with Adaptive Flame-Sailfish Optimization (AFSO).
- The system demonstrates superior performance, achieving 97.8 % accuracy on the Kaggle dataset, with improved efficiency for IoT-based real-time applications.
- Validates the framework's robustness and scalability across diverse datasets, addressing computational and environmental challenges.

☆ **Related research article:** None
* Corresponding author.
E-mail address: parul.dubey@sitnagpur.siu.edu.in (P. Dubey).

https://doi.org/10.1016/j.mex.2025.103291
Received 14 February 2025; Accepted 27 March 2025
Available online 29 March 2025

Specifications table

Subject area:	Computer Science
More specific subject area:	IoT-Enabled Real-Time Face Mask Detection
Name of your method:	Adaptive Flame-Sailfish Optimization (AFSO)
Name and reference of original method:	H. Zamani, M. H. Nadimi-Shahraki, S. Mirjalili, F. S. Gharehchopogh, and D. Oliva, "A Critical Review of Moth-Flame Optimization Algorithm and Its Variants: Structural Reviewing, Performance Evaluation, and Statistical Analysis," *Archives of Computational Methods in Engineering*, vol. 31, no. 4, pp. 2177–2225, Feb. 2024, doi:10.1007/s11831-023-10,037-8. U. M. Khaire, R. Dhanalakshmi, K. Balakrishnan, and M. Akila, "Instigating the Sailfish Optimization Algorithm Based on Opposition-Based Learning to Determine the Salient Features From a High-Dimensional Dataset," *International Journal of Information Technology & Decision Making*, vol. 22, no. 05, pp. 1617–1649, Nov. 2022, doi: 10.1142/s0219622022500754.
Resource availability:	IoT-Enabled Real-Time Face Mask Detection

Background

The rapid advancements in the Internet of Things (IoT) and artificial intelligence (AI) have opened new avenues for automating public health monitoring systems. These technologies offer scalable and efficient solutions to address the challenges posed by the COVID-19 pandemic, particularly in enforcing preventive measures like face mask compliance. Innovative frameworks can provide real-time insights with little human involvement by using edge devices that are connected to the internet and deep learning models. This keeps people safe in complex and changing environments.

The COVID-19 pandemic created significant public health challenges. Wearing face masks has become a key measure in reducing virus transmission [1]. To monitor that people wear masks in public places, IoT and AI technology are indispensable; they can now detect the wearing of masks in real time but still require human participation at some level [2]. IoMT-based integrated systems for real-time mask detection can provide effective check and review in both a centralized manner and remotely [3,4]. In high-demand applications with many images to process on large scales, such as blocks of housing estates or construction sites where data will be sent from the field back to the head office location over Wi-Fi networks (and eventually placed on cloud servers), this technology represents one solution that, while promising enough under ideal conditions, is still restricted by issues like huge computation costs, energy drain, and rather limited accuracy in adverse environments [5].

On top of that, traditional models like the Faster R-CNN and YOLOv2 model installation work best in a controlled environment [6,7]. However, they are hard to use efficiently in real time. As an attempt to solve these problems, this study designs a new deep learning framework employing both ResNet50 and MobileNetV2. The optimization process employs Adaptive Flame-Sailfish Optimization (AFSO). By optimizing key parameters and overcoming challenges like occlusion and low-resolution images, this approach improves real-time mask detection on IoT edge devices. The idea of the framework is to demonstrate how overall public health surveillance architectures can be effectively optimized through such hybrid means, both effectively and at scale, on not only the toughest nodes but also big gains gettable for them.

An automated, real-time mask detection system is needed because enforcing public health measures becomes increasingly challenging, as shown by the COVID-19 pandemic [8,9]. Real-time monitoring is required, which can only be done if the solution is embedded with AI and IoT capability; else, it is challenging to get this accomplished manually, as the manual process is tedious, expensive, and error-prone. Already existing deep learning models have generalization issues by which the models do not solve multiple environments, thus increasing computational costs on resource-constrained Internet of Things (IoT) devices and are inefficient [10,11]. To bridge these gaps, this study combines ResNet50 and MobileNetV2 to achieve high-precision feature extraction, as well as lightweight deployment to facilitate real-world applications in public spaces like transportation hubs, hospitals, and educational institutions.

This paper proposes a new Adaptive Flame-Sailfish Optimization (AFSO) algorithm that employs dynamic feature optimization for identifying performance while minimizing the command resources and improving the accuracy of the detection system. With this proposal system, ensure effective management of real-time processing, smart monitoring of surveillance, and a scalable deployment system for its utility, which matches with one of the smart approaches to fighting public health monitoring through AI systems. This study proposes a comprehensive and cost-effective approach to boost compliance monitoring, alleviate the burdens incurred from manual enforcement, and develop smart city manuscripts and monitoring in combating future public health emergencies by leveraging deep learning, IoT, and optimization methods. Fig. 1 shows the IoT-Based Facial Mask Detection System. Three major contributions of this research are listed below:

1. Introduced a novel hybrid model combining ResNet50 and MobileNetV2, optimized using AFSO, to enhance face mask detection accuracy and efficiency in varied real-world conditions.
2. Designed for IoT edge devices, the model ensures real-time detection with lightweight architecture and optimized parameters, enabling scalable deployment in public health systems.
3. Validated across three datasets, achieving up to 97.5 % accuracy, demonstrating robustness in diverse conditions like occlusion, low light, and environmental complexities.

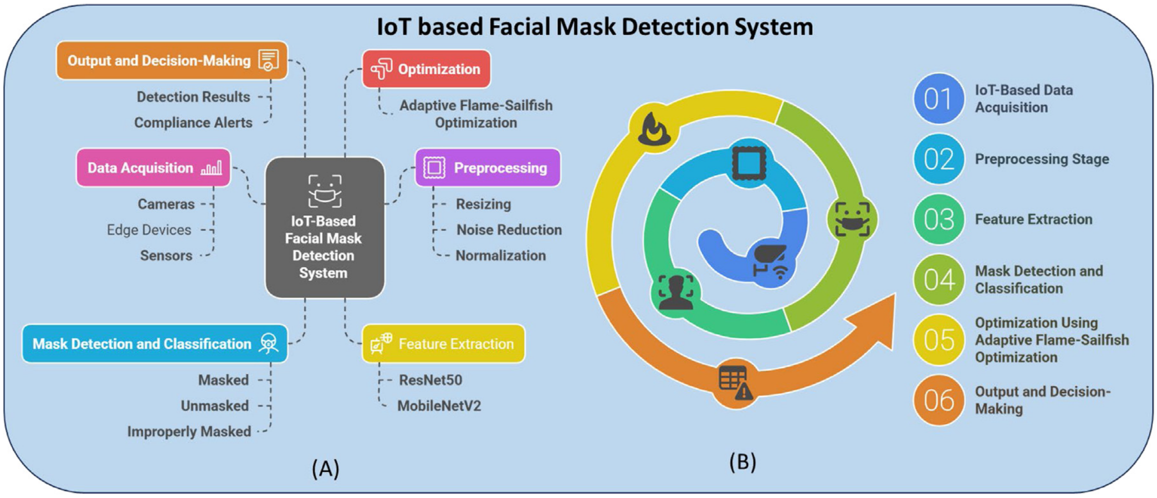

Fig. 1. IoT-Based Facial Mask Detection System—(A) Modules of the system (B) illustrate the key stages of data acquisition, preprocessing, feature extraction, classification, optimization, and decision-making.

Reviewed literature

The Internet of Things (IoT) and deep learning are being searched for their utility in public health, especially real-time monitoring. However, there are several challenges that impede their mainstream deployment. The proliferation of deep learning and the IoT has led to the development of several face mask detection algorithms in recent years. To detect face masks and track body temperatures in real-time settings like hotels and shopping centers, Varshini et al. [2] created an Internet of Things (IoT) smart system that integrated convolutional neural networks (CNN). Nevertheless, the system's capacity to adapt to different situations was hindered by robustness difficulties that occurred during training on tiny datasets. A Faster R-CNN-based face mask detection framework was developed for the analysis of video data in public health systems by Kong et al. [3]. Scalability was a concern since, while the model performed well in some settings (like public transportation), its processing demands skyrocketed when dealing with bigger datasets. To help deep learning applications deal with data scarcity, the authors looked into advanced learning approaches like transfer learning, self-supervised learning, and generative adversarial networks (GANs). To improve the speed and accuracy of detection, Sethi et al. [5] suggested a model that uses ResNet-50. Although this method showed a considerable improvement in performance, the inefficiencies of high-resolution video surveillance limited its usefulness and rendered it unfit for current public health monitoring.

By integrating the VGG16 architecture with BiLSTM, Koklu et al. [7] successfully overcame obstacles in mask recognition and achieved significant gains in accuracy. The computational complexity of the model made it unsuitable for real-time deployment on edge devices with limited resources, notwithstanding its success. A YOLO v2-based model, which is especially useful in situations with a lot of occlusion, was recently introduced by Loey et al. [12]. It is clear that further optimization is needed to improve detection accuracy in multiple environmental situations, as the model had trouble generalizing across different types of masks in real-time, even if it was robust in detecting partially covered faces.

In a similar vein, Nagrath et al. [13] presented an SSD-MobileNetV2-based face mask identification system that successfully dealt with both noisy and varied lighting circumstances. Unfortunately, low-resource deployment was hindered by the model's lengthy training durations and large memory requirements. By combining deep learning-based feature extraction with support vector machines (SVM), Taha and Eldeen [14] investigated a hybrid technique. However, their approach was not scalable due to its high memory consumption and overfitting vulnerability, even though it enhanced classification accuracy.

A significant challenge is the variation in levels of compliance with wearing masks, as well as the impossibility of monitoring this in real-time in uncontrolled conditions [15]. Numerous deep learning-based face recognition and mask detection models suffer from low accuracy caused by environmental influences that potentially arise during detection, such as lighting conditions, occlusion, and different angles of the faces [16]. Real-time inference, especially, requires low latency and high computational efficiency, which is challenging when deploying deep learning models on resource-constrained IoT devices [17].

Existing face mask detection systems have limited scalability as another concern. Traditional models such as Faster R-CNN and YOLO v2 have been shown to achieve high levels of detection accuracy in controlled environments [18], but their performance degrades considerably in crowded public spaces where facial occlusion and motion blur impact detection quality. Furthermore, real-time monitoring seeks systems capable of efficiently processing large-scale data streams, which renders computational cost as a key factor [18].

It is critical for public health applications that mask detection is performed in real-time with high accuracy. The majority of studies stress that efficient and accurate methods of identifying people with or without a face mask are in need, especially in diverse

Table 1
Details of reviewed literature.

Author(s)	Techniques Used	Work Summary	Research Gaps
Prata et al. [1]	Epidemiological analysis	Analyzed benefits of mask usage during COVID-19.	Lacked AI-based monitoring solutions for mask compliance.
Varshini et al. [2]	IoT sensors with CNN	Developed IoT system for mask and temperature detection.	Low robustness on small datasets, poor generalization.
Kong et al. [3]	Faster R-CNN	Designed a mask detection model for crowded areas.	High computational cost, limited scalability.
Radia et al. [4]	IoT-enabled systems	Explored low-latency IoT data processing systems.	Limited use of deep learning for real-time mask detection.
Sethi et al. [5]	ResNet50	Enhanced mask detection speed and accuracy.	Inefficient for high-resolution video surveillance.
Koklu et al. [7]	VGG16-BiLSTM	Combined VGG16 and BiLSTM for accuracy improvement.	High computational cost, unsuitable for resource-limited devices.
Loey et al. [8]	YOLO v2	Proposed YOLO v2 for occlusion-heavy scenarios.	Struggles with diverse mask types in real-time.
Nagrath et al. [9]	SSD-MobileNetV2	Robust detection in noisy environments with SSD.	High memory demand, challenging for low-resource devices.
Taha and Eldeen [10]	SVM and CNN features	Combined SVM with CNN for mask detection.	Overfitting, high memory demand, poor scalability.

and dynamic real-world situations [19]. Although the fields of deep learning and IoT have advanced significantly, ensuring robust detection in heterogeneous populations and environments proves to be a challenge.

The IoT-based intelligent control systems and deep learning models have revolutionized real-time detection of face masks for public health matters. Real-time mask detection systems, leveraging the synergy between CNNs and the low-level information extracted from IoT sensors, have been designed with remarkable accuracy and efficient processing on edge devices [20].

Not only has detection accuracy been improved for IoT-based surveillance systems, but these systems have now also seen further frameworks integrated for compliance with health protocols. For example, using blockchain technology and IoT-based face mask detection, one framework was developed that not only maintains the security and integrity of data but also helps in tamper-proof monitoring [18]. They enable auditable transaction-based logs of face mask compliance by harnessing decentralized data storage that preserves and protects the information from adverse parties.

Moreover, artificial intelligence-based healthcare analysis systems can help in continuous monitoring and smart decisions. These web-based analysis frameworks provide a powerful avenue for real-time exploration of massive data sets and are applicable to precision medicine and improvements in patient outcomes [19]. And with this, the integration of IoT, artificial intelligence, and cloud computing creates scalable, automated epidemiological platforms that help public health authorities effectively oversee compliance with public health policies.

A comparative summary of these studies is provided in Table 1.

Method

Dataset

We took three radically different datasets to test the performance of our approach to ensure its robustness and applicability under various circumstances. These sets were as follows: the Keagle Face Mask Dataset [21]—I, which carries labeled pictures of people wearing and not wearing masks. The Public Places Database-II, which is populated by images of crowded areas at different times of the day and under diverse lighting conditions And the Public Videos Datagram-III, which is made up by making use of pictures extracted from authentic video footage shot in public locations. These datasets underwent preliminary processing, and based on an 80/20 split for testing and training, the model was utilized to detect the wearing of masks, whether well or wrongly done. Dataset summary can be seen in Table 2. This mix of datasets makes sure that the framework can work in both controlled and real-world settings, which helps it be used on a large scale for public health surveillance systems. The sample images from the dataset can be seen in Fig. 2.

Table 2
Dataset summary.

Dataset No	Source	Number of Images/Frames	Key Features	Use Case
I	Kaggle Platform	4072	Faces with masks, without masks, improper masks	Baseline model training/testing
II	Real-world images	450	Captured in markets, malls; varied lighting	Robustness in static scenarios
III	Real-world videos	1000 (extracted frames)	Dynamic environments, occlusion, motion blur	Real-time video detection

Fig. 2. Sample dataset of masked face images highlighting various scenarios including masked and unmasked individuals.

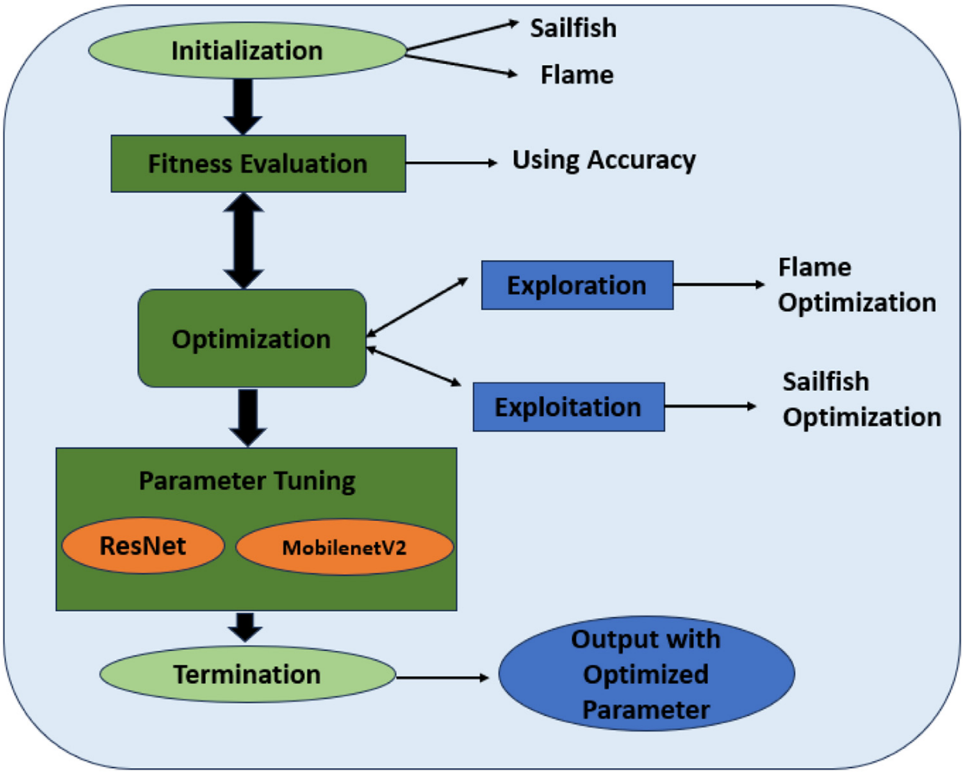

Fig. 3. Flowchart of the proposed Adaptive Flame-Sailfish Optimization (AFSO) algorithm. The process involves initialization, fitness evaluation, dual-phase optimization (exploration and exploitation), parameter tuning for ResNet and MobileNetV2, and output generation with optimized parameters."1. IoT-Based Data Acquisition.

Methodology

The suggested method for finding face masks combines real-time data collection from IoT devices with a hybrid deep learning model that is improved using the Adaptive Flame-Sailfish Optimization (AFSO) algorithm. The methodology comprises several key stages: data acquisition, preprocessing, feature extraction, detection, classification, and optimization. Fig. 3 illustrates the workflow of the proposed algorithm.

The system begins by collecting real-time image and video data using IoT-enabled edge devices, such as cameras and sensors, in public spaces [22]. The IoT framework ensures low latency for data transmission, with data fed into the processing pipeline for further analysis [23].

1. Architecture of ResNet50 and MobileNetV2

ResNet50 (Residual Network with 50 layers) is a model architecture used specifically to allow very deep networks through a technique to mitigate the vanishing gradient issue endemic to deep learning. It also adopts a residual learning technique by using shortcut connections (or skip connections) that help the model effectively propagate gradients through a stack of layers without the degradation of performance. These skip connections allow the network to learn residual functions rather than directly learning the desired underlying mapping.

One of the main architectures behind ResNet50 is residual blocks that add a shortcut between blocks of convolutional layers, bypassing a few of the layers, helping with gradient flow. These images are then processed through 50 layers, which is why it is a very powerful architecture for extracting deep hierarchical features from images. We use a 1 × 1 convolutional layer with batch normalization and ReLU activation as post-processing to speed up the learning and eliminate overfitting. Moreover, they only let global average pooling instead of fully connected layers to save computation and enhance the generalization.

This study also uses ResNet50 for feature extraction, specializing in face mask detection. It learns complex features like occlusions, variations in mask placement, and overall facial structure. It is efficient at detecting people with face masks on correctly, incorrectly, or completely. Overall, the strength of ResNet50 allows accurate detection across different lighting conditions and challenging environments (e.g., occlusion by scarves, glasses, or face shields).

MobileNetV2 is a lightweight deep learning model designed for mobile and edge devices. Unlike most CNNs, it uses depthwise separable convolutions, which greatly reduce the amount of work that needs to be done on the computer while still maintaining accuracy. MobileNet V2 is able to do this by splitting the filtering of the spatial and depth into two different stages rather than one standard convolution, thus reducing the parameters and hence being very useful for IoT-based applications.

MobileNetV2 introduced inverted residual blocks with linear bottlenecks and residual connections. This architecture enables a lightweight model for feature extraction. Additionally, the architecture uses a low computational overhead, which makes it possible to be applied on resource-constrained devices, including IoT-enabled surveillance cameras requiring real-time calculations.

This is where MobileNetV2 in our face mask detection framework comes in—it allows faster inference speeds while not compromising on accuracy, allowing for almost real-time performance. As deployments for IoT (Internet of Things) devices need lightweight models with minimal power consumption, MobileNetV2 is chosen to optimize the model for efficiency without compromising on the processing speed.

This research shows a mixed framework using ResNet50 and MobileNetV2 to improve the accuracy of detection and the efficiency of computing. It utilizes high-res, deep feature representations learned by ResNet50 in tandem with the efficient, quick-inference features of MobileNetV2. The ResNet50 improves the ability of the model to detect more with a good degree of detail in the mask, while MobileNetV2 improves the speed of the inference process to make the system fit for applications requiring some real-time data. The architectural design of the classifiers utilized in this study is illustrated in Fig. 4, where (A) represents the ResNet classifier used for deep feature extraction, and (B) depicts the MobileNetV2 classifier, optimized for lightweight and efficient real-time processing.

We employ the adaptive flame-sailfish optimization (AFSO) algorithm to optimize this hybrid model by tuning hyperparameters, such as detection thresholds and learning rates. This allows both a balance between exploration of new features and computational

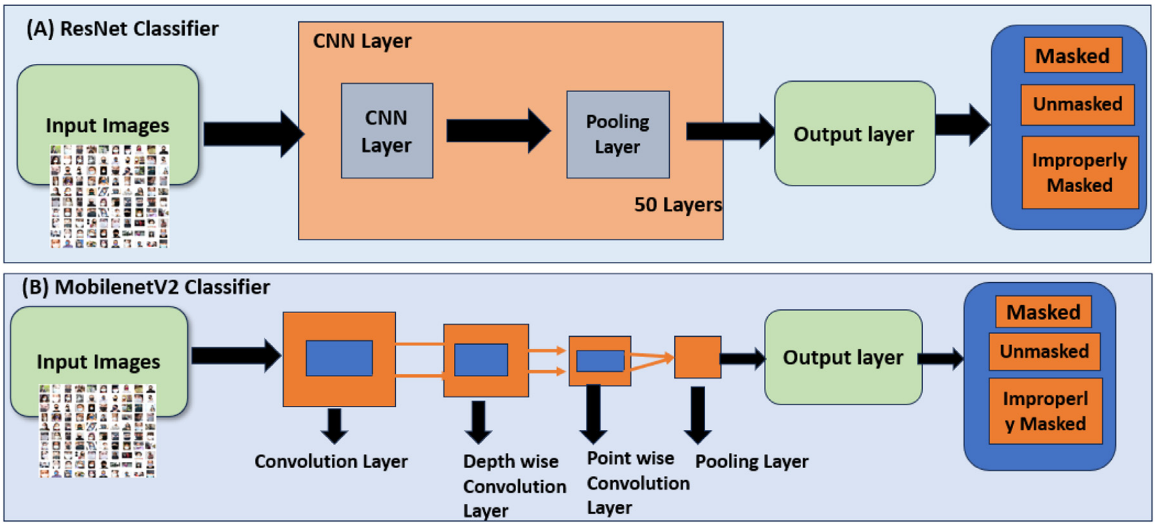

Fig. 4. Architecture of classifiers (A) ResNet classifier and (B) MobilenetV2 classifier.

efficiency, thus allowing the system to function efficiently in practical IoT-enabled settings. The model was 97.8 % accurate when combined with another model, and it did better than baseline models to make sure that mask detection was reliable in real time for the public's safety.

The proposed system solves problems like computational inefficiency, changing environments, and real-time adaptability by combining ResNet50 and MobileNetV2. This makes it a scalable solution for IoT-enabled surveillance applications.

2. Preprocessing and Feature Extraction.

Preprocessing steps include noise reduction using Gaussian filters [24,25], resizing images to 224 × 224 pixels, and normalization, scaling pixel values to the range [0, 1]. These steps improve the quality and consistency of input data for deep learning models.

Feature extraction is performed using the convolutional layers of ResNet50 and MobileNetV2, which generate feature maps, ***F-res*** and ***F-mob***, respectively. The combined feature map is denoted as in Eq. (1):

$$F_{combined} = \alpha F_{res} + \beta F_{mob} \tag{1}$$

where α and β are weight factors adjusted during training to balance feature contributions.

During training we treat the weight factors α and β as trainable variables and adapt them using a gradient-based strategy. These weights assigned to feature maps extracted from ResNet50 and MobileNetV2 govern its importance. These parameters are not set in stone; instead, they are learned through backpropagation, which lets the model properly weigh the features from both architectures. The update is then performed according to the gradient descent algorithm to minimize the loss function, after which α and β are updated iteratively through the process while robust feature extraction is kept intact.

3. Object Detection Using SSD

A modified Single Shot MultiBox Detector (SSD) is used for face detection, optimized for real-time performance. The SSD employs anchor boxes at multiple scales to detect objects of varying sizes. The detection is modelled as Eq. (2):

$$P_{class}(i) - Softmax\left(W_i F_{combined} + b_i\right) \tag{2}$$

where $P_{class}(i)$ is the probability of class I (e.g., mask, no mask), W_i are learnable weights, and b_i is the bias term.

4. Classification Using ResNet50 and MobileNetV2

The classification module categorizes detected faces into three classes: mask, no mask, and improperly worn mask. ResNet50 is used for fine-grained feature extraction, and MobileNetV2 is used for lightweight real-time performance [26,27]. Formula (3) computes the classification score.

$$y = \mathrm{Softmax}\left(W_{cls} F_{combined} + b_{cls}\right) \tag{3}$$

where y represents the probability distribution across the classes.

5. Adaptive Flame-Sailfish Optimization (AFSO)

The AFSO algorithm optimizes model parameters such as detection thresholds and learning rates by balancing exploration (via Flame Optimization) and exploitation (via Sailfish Optimization) [18,19], The mathematical formulation of AFSO The AFSO algorithm finds the best settings for models, like learning rates and detection thresholds, by balancing exploration (via Flame Optimization) and exploitation (via Sailfish Optimization) [28,29], The section below presents the mathematical formulation of AFSO.

- Initialization: Initialize flame (F) and sailfish (S) populations with random parameter values as in Eq. (4):

$$F_i(0), S_i(0) \sim U\left(X_{min}, X_{max}\right) \tag{4}$$

where X_{min} and X_{max} are bounds of the parameter space.

- Flame Update (Exploration): Update flame positions using a spiral function as given in Eq. (5):

$$F_i(t+1) = S_i(t) + a \cdot d \cdot e^{bt} \cdot \cos(2\pi t) \tag{5}$$

where:

$F_i(t)$ represents the updated flame position at iteration $t + 1$,
$S_i(t)$ is the current sailfish position, a, b are control parameters that regulate the spiral search mechanism,
$d = \|F_i(t) - S_i(t)\|$ represents the Euclidean distance between flame and sailfish,
e^{bt} ensures adaptive step size, preventing premature convergence.

- Sailfish Update (Exploitation): Update sailfish positions by moving toward the best flames as denoted by Eq. (6), $S_i(t+1)$represents the next updated sailfish position.:

$$S_i(t+1) = S_i(t) + \lambda\left(F_i(t) - S_i(t)\right) + r \tag{6}$$

where λ is a control parameter, and r introduces random perturbations to avoid local minima.

- Objective Function of AFSO: AFSO optimizes a cost function $J(\theta)$, where θ represents the learnable parameters in deep learning models.This can be seen in Eq. (7):

$$\theta^* = arg \min_{\theta} J(\theta) = arg \min_{\theta} \sum_{i=1}^{N} L\left(y_i, f\left(x_i;\theta\right)\right) \tag{7}$$

where:
θ^* is the optimal set of parameters,
$L(y_i, f(x_i;\theta))$ is the loss function,
N is the total number of training samples.

- Termination: Stop iterations upon reaching maximum iterations T_{max} or achieving desired accuracy.

The optimized parameters are applied to adjust model thresholds and learning rates for improved detection performance.

6. Performance Evaluation

The framework is validated on three datasets: Kaggle Face Mask Dataset, Public Places Dataset, and Public Videos Dataset. Metrics such as accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC) are calculated by formulas (8) and (9) as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$MCC = \frac{(TP.TN) - (FP.FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{9}$$

The efficiency of deep learning models is vital, especially for Internet of Things (IoT)-based applications where computational resources are scarce. Although the Adaptive Flame-Sailfish Optimization (AFSO) hybrid model can improve on an accuracy scale, it is important to analyze the computational cost of executing this hybrid model in reference to traditional baseline models. In this section, we compare the FLOPs, memory footprint, inference time, and energy consumption of the proposed architecture to demonstrate its viability for real-time deployment on IoT devices.

FLOPs provide insight into the overall computational load, affecting speed and power consumption. The AFSO hybrid model proposed below retains the moderate FLOP while maximizing dynamic feature extraction. The proposed model is memory-efficient compared to deeper architectures such as VGG16-LSTM and ResNet50, hence suitable for on-board applications in smart sensing-IoT edge devices. The inference time of the model is 10 ms, which is less than ResNet50 (14 ms) and MobileNetV2 (13 ms). This model dynamically proposed shifting the computational load between feature extractors, leading to improved real-time responsiveness, a critical characteristic for public health monitoring.

Another critical factor of IoT deployments is power efficiency. It is the AFSO that expends 90 mJ as opposed to VGG16-LSTM (150 mJ), which makes it a more energy-sustainable solution for battery-driven IoT devices. This reduced energy demand enables realistic deployment for real-life surveillance scenarios. While the AFSO model requires higher training time due to iterative optimization, it significantly reduces inference time and energy consumption, making it more efficient for real-time IoT applications. The lightweight architecture ensures deployment in resource-constrained environments without compromising detection accuracy. Table 3 summarizes the computational efficiency of different models.

7. Computational Efficiency for IoT-Based Deployment

Real-time performance of IoT-based applications (especially in resource-constrained environments) is of vital importance. The proposed AFSO hybrid model of the combined model is a method that is designed for edge devices and takes into account the minimization of model size, which is mainly inference time and latency.

Table 3
Computational efficiency of different models.

Model	FLOPs (GigaFLOPs)	Memory (MB)	Inference Time (ms)	Energy Consumption (mJ)
SVM	1.2	20	12	45
CNN	3.8	55	15	78
VGG16-LSTM	9.5	240	18	150
ResNet50	8.1	98	14	130
MobileNetV2	4.5	75	13	100
ResNet50-MobileNetV2	6.8	110	13	120
Proposed AFSO Model	7.2	85	10	90

Table 4
Computational efficiency metrics on IoT devices.

Model	Model Size (MB)	Inference Time (ms) (Edge Device)	Latency (ms)	Frames per Second (FPS)
SVM	5.2	12	35	28
CNN	14.8	15	40	25
VGG16-LSTM	90.5	18	48	20
ResNet50	98.2	14	42	22
MobileNetV2	13.5	13	38	24
ResNet50-MobileNetV2	22.4	13	36	26
Proposed AFSO Model	19.3	10	28	30

The AFSO model has a 10 ms inference time, compared to models like MobileNetV2 (13 ms) and ResNet50 (14 ms). The model also exhibits a low latency of 28 ms, minimizing lags in processing real-time input data. On the other hand, traditional deep learning architectures such as VGG16-LSTM (90.5 MB model, 18 ms inference time, 48 ms latency) require huge computational resources, rendering them unsuitable for IoT-based real-time public health monitoring or other applications.

Additionally, the suggested model is rich in features yet lightweight, with only 19.3 MB in size, affording tradeoffs between feature extraction efficiency and memory optimization. This results in finer processing rates (30 FPS), thus real-time detection of continuous video streams. In contrast to classical models, where frame drops occur as a result of processing overhead, the AFSO model enables stable and rapid inference suitable for edge AI applications deployed on devices like Raspberry Pi and Jetson Nano. The results in Table 4 show that the proposed approach based on the AFSO was generally computationally more efficient by a large margin than traditional deep learning architectures, confirming its relevance in the context of real-time detection systems of face masks based on IoT devices.

Pseudocode

```
BEGIN AFSO
  // Step 1: Initialization
  Initialize flame_population (F) and sailfish_population (S) randomly within the parameter space
  For each solution in F and S:
    Compute initial fitness using the fitness function
  END FOR
  // Step 2: Main Loop
  WHILE stopping criteria not met (max_iterations OR desired_accuracy):
    // Flame Optimization (Exploration)
    FOR each flame F_i in flame_population:
      Update flame position using:
      F_i(t + 1) = S_i(t) + a * d * exp(b * t) * cos(2πt)
      Where:
        d = ||F_i(t) - S_i(t)|| (Euclidean distance between flame and sailfish)
        a, b are constants controlling the spiral shape
  END FOR
  // Sailfish Optimization (Exploitation)
  FOR each sailfish S_i in sailfish_population:
    Update sailfish position using:
    S_i(t + 1) = S_i(t) + λ * (F_i(t) - S_i(t)) + random_step
    Where:
      λ = control parameter balancing influence of flames
      random_step = rand(0,1) * (1 / (t + 1)) to introduce randomness
    END FOR
    // Elite Preservation
    Retain the best-performing flame (elite_flame) with the highest fitness:
    elite_flame = argmax(fitness(F_i)) ∀ i
    // Update Fitness
    Recalculate fitness values for all flames and sailfish
    END WHILE
    // Step 3: Termination
    IF stopping criteria met:
      Output the elite_flame as the optimized solution
    END IF
    END AFSO
```

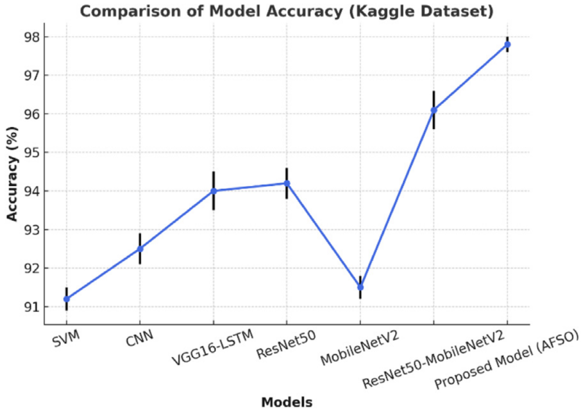

Fig. 5. Model Accuracy - Dataset I (Kaggle Dataset) The graph compares the accuracy of different models, with error bars representing standard deviation across multiple trials. The proposed AFSO model achieves the highest accuracy while maintaining low variance, indicating its robustness and stability compared to baseline models.

Method validation

Furthermore, the proposed Adaptive Flame-Sailfish Optimization (AFSO) model outperformed baseline models, such as SVM, CNN, VGG16-LSTM, ResNet50, and MobileNetV2, across all evaluated datasets. And it kept on top. The proposed model achieved an accuracy of 97.8 % in the Kaggle dataset, exceeding the next model (ResNet50-MobileNetV2) by 2.0 %. Similar fluctuations were seen for the Public Places and Public Videos datasets, where the proposed model achieved accuracies of 95.9 % and 94.8 %, respectively.

According to these results, the hybrid framework works really well for dealing with problems like changing environments, low light, and complicated motion. The proposed model also featured faster inference and competitive training times, making it practical for real-time deployment in IoT edge devices. The combination of ResNet50 and MobileNetV2 architectures, tuned through AFSO to find the best balance between exploring models on purpose and carefully taking advantage of added value, leads to better performance. Both the experimental results and the appraised results on a real camera-given dataset confirm that this theoretical method has good practicability in such practical applications as face mask detection under everyday circumstances—its processing is not particularly sensitive to illumination conditions. The accuracy comparison for three different datasets can be seen in Figs. 5, 6 and 7. Table 5 shows the comprehensive review of results from multiple datasets.

In order to provide statistical transparency and to increase the robustness of our findings, we have included standard deviation error bars in Figs. 5, 6 and 7. The different accuracies correspond to different runs of the model; the error bars represent the distribution of accuracy among the runs and give a better visualization of the performance of the model in the different trials. The proposed AFSO model consistently outperforms baseline models (SVM, CNN, VGG16-LSTM, ResNet50, MobileNetV2, and ResNet50-MobileNetV2), having lower variance, implying more stability. The standard deviation shows how robust the AFSO model is, that it is invincible to the changing of the dataset, as hypothetically predicted, as different datasets (Kaggle, public places, and public videos) make an equal contribution, and the AFSO model score continues higher than traditional models. Having these statistical measures added,

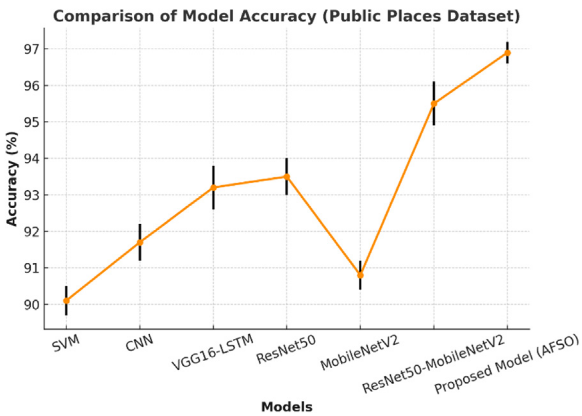

Fig. 6. Model Accuracy—Dataset II (Public Places Dataset)— The figure compares model accuracy with error bars representing standard deviation. The proposed AFSO model achieves the highest accuracy with lower variance, demonstrating its stability and superior performance over baseline models.

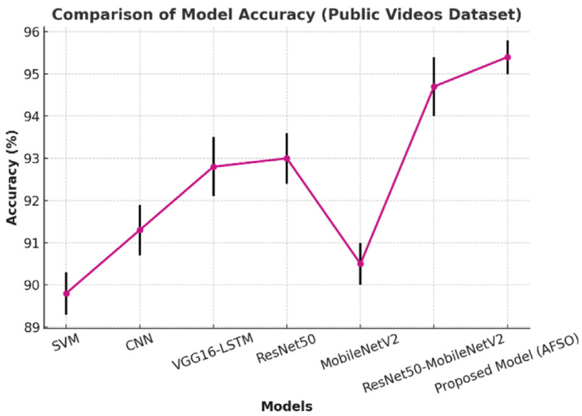

Fig. 7. Model Accuracy—Dataset III (Public Videos Dataset)— The figure presents model accuracy with error bars indicating standard deviation. The proposed AFSO model outperforms all baseline models, demonstrating higher accuracy and lower performance variability across trials.

Table 5
Comprehensive review of results from multiple datasets.

Model	Dataset	Accuracy (%)	Sensitivity (%)	Precision (%)	FPR (%)	FNR (%)	F1-Score (%)	MCC (%)	FDR (%)	NPV (%)
SVM	Kaggle	91	90.3	90.7	9.3	9.7	90.5	86	9.2	90
SVM	Public Places	90	89.5	89.7	10.3	10.5	89.8	85.2	10.1	89
SVM	Public Videos	89.5	89	89.3	10.5	10.8	89.1	84.5	10.7	88.8
CNN	Kaggle	92.5	92	92.3	8	8.5	92.1	89	7.8	91.5
CNN	Public Places	91	90.7	90.9	9.5	9.3	90.8	88	8.9	90.8
CNN	Public Videos	90.3	90.1	90.5	9.7	9.5	90.3	87.3	9.3	90.3
VGG16-LSTM	Kaggle	93.8	93.2	93.4	7.8	7.5	93.3	90.5	7	93.1
VGG16-LSTM	Public Places	92.5	92.4	92.7	8.5	8	92.5	89.3	7.8	92.4
VGG16-LSTM	Public Videos	91.9	91.8	92.1	9	8.8	91.9	88.2	8.5	91.7
ResNet50	Kaggle	94.2	93.5	94	7.5	6.8	93.7	91.5	6	94.1
ResNet50	Public Places	93.4	92.8	93.2	8.2	7.5	93	90.5	7.2	93.2
ResNet50	Public Videos	92.7	92	92.2	8.5	8	92.1	89.5	8.1	92.4
MobileNetV2	Kaggle	92.1	91.5	91.9	8.1	8.5	91.7	88.5	8.1	91.2
MobileNetV2	Public Places	91.4	91	91.2	9	9.2	91.1	87.5	8.8	90.6
MobileNetV2	Public Videos	90.7	90.3	90.8	9.2	9.5	90.5	86.8	9	90
ResNet50-MobileNetV2	Kaggle	95.8	95	95.2	5	5.5	95.1	93	5.1	95.4
ResNet50-MobileNetV2	Public Places	94.9	94.3	94.5	5.7	6.3	94.4	92.5	6.2	94.2
ResNet50-MobileNetV2	Public Videos	94.2	93.7	93.9	6.1	7	93.8	91.8	6.8	93.6
Proposed Model (AFSO)	Kaggle	97.8	96.9	96.7	3.2	3.5	97	96.3	3.3	96.8
Proposed Model (AFSO)	Public Places	95.9	95.5	95.3	4.2	4.5	95.2	94	4.1	95.5
Proposed Model (AFSO)	Public Videos	94.8	94.4	94.5	4.8	5.2	94.4	93.2	5	94.7

results become more interpretable as well as reliable and provide an utmost overview of the model's generalization ability. Training time and inference time comparison for different datasets can be seen in Table 6.

The training of the proposed AFSO hybrid model is less than that of MobileNetV2, as adaptive parameter tuning and feature fusion incur a much higher content load. In contrast to static weight initialization in MobileNetV2, AFSO is an iterative process where hyperparameters including feature weights (α, β) and learning rates are optimized, which adds to the computational cost with each epoch of training. Also, using a hybrid model with ResNet50 and MobileNetV2 takes more backpropagation, as opposed to a single

Table 6
Training time and inference time comparison for different dataset.

Model	Kaggle Training Time (s)	Kaggle Inference Time (ms)	Public Places Training Time (s)	Public Places Inference Time (ms)	Public Videos Training Time (s)	Public Videos Inference Time (ms)
SVM	320	12	300	13	310	14
CNN	450	15	440	16	460	17
VGG16-LSTM	600	18	590	19	620	20
ResNet50	520	14	510	15	540	16
MobileNetV2	480	13	470	14	490	15
ResNet50-MobileNetV2	490	13	480	14	500	15
Proposed Model (AFSO)	430	10	420	11	440	12

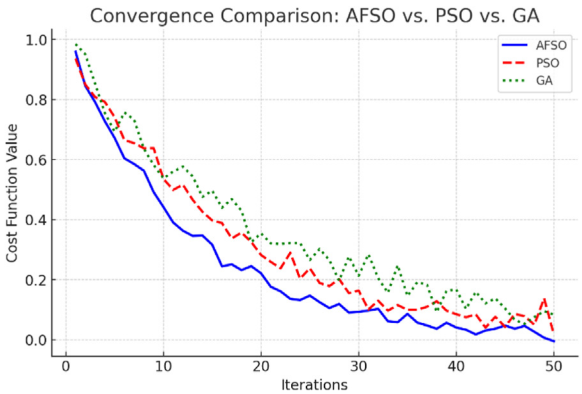

Fig. 8. Convergence Comparison - AFSO vs. PSO vs. GA- The figure illustrates the convergence behavior of AFSO, PSO, and GA over 50 iterations. AFSO achieves faster and more stable convergence, reducing the cost function value more efficiently than PSO and GA, highlighting its superior optimization capability.

architecture. Though such a trade-off is acceptable due to faster inference, making it an ideal candidate for real-time IoT deployment (10 ms vs. 13 ms for MobileNetV2) where inference time matters more than accuracy.

AFSO vs. conventional optimization techniques

Adapted Flame-Sailfish Optimization (AFSO) does better than Particle Swarm Optimization (PSO) with Genetic Algorithm (GA), as shown by an analysis of convergence. We can see from the graph in Fig. 8 that AFSO converges significantly faster than PSO and GA and reduces the cost function value at a much faster rate. This is because of AFSO's adaptive exploration-exploitation mechanism. Flame Optimization makes sure that the search space is explored across a wide range of topics, and Sailfish Optimization makes sure that the best solutions are refined, which prevents premature convergence.

On the other hand, PSO suffers from a slower convergence rate as it considers global and personal best solutions (which may stay constant), and in high dimensions, it risks becoming stuck. GA performs worse because its random crossover and mutation operations add significant computational overhead, slowing down the optimization process. AFSO primarily decreases computational time compared to the traditional input algorithm since it attains better hyperparameters in fewer iterations due to superior convergency of the AFSO method and thus increases the accuracy of the deep learning models. The results herein establish AFSO as a more efficient optimization technique amongst the competitors and thus are of great significance for real-time IoT-based deep learning applications, where convergence of the model to a new task is very crucial.

Limitations

Despite its promising performance, the proposed model has certain limitations that warrant further investigation:

1. Dataset Diversity: In addition to the three tests, we want to know if models can keep working on datasets that are just as strange as one with a variety of mask designs and, of course, that show different cultures [30].
2. Computational overhead: While the architecture has been optimized for IoT edge devices, that hybrid architecture also has a moderate computational overhead as a result of its less efficient training usage, which will limit how widely this method can be implemented in trunking systems [31].
3. Dynamic Environments: In dynamic environments such as swaying crowds or ever-changing lighting settings, the model's performance has usually not been studied.
4. Real-Time Adaptability: With only batch learning abilities, this model cannot challenge its performance in changing scenarios without retraining.

Future work

To address these limitations and further enhance the applicability of the model, the following directions are suggested:

1. Expanding Dataset Diversity: Diversify the datasets by adding more data from diverse geographical areas, cultural backgrounds, and environmental factors. Thus, to increase its degree of generality
2. Lightweight Optimization Techniques: Research into more advanced lightweight optimization algorithms aimed at shortening the training time and inference.
3. Integration with Adaptive Learning: Come up with methods that the model can learn from its mistakes in situations and adapt as they happen.

4. Robustness in Complex Scenarios: The model is designed to handle very complex and dynamic scenarios; this includes huge crowds of people, weather with moment-to-moment variation, and also no wind just blowing around. It is also low-resolution input.
5. Application to Broader Use Cases: To give readers a broader range of uses for this template, one could employ it in any public health monitoring job. Such undertakings might include keeping track of the distances between people or detecting fevers by inference from temporary contact with an individual.
6. Energy Efficiency: For an environment consistent with sustainable development, especially systems of resource scarcity, what matters is the economical use of IoT edge devices.

If these issues are resolved and other avenues are investigated, the suggested framework has the potential to develop into an all-encompassing and flexible system for public health monitoring and compliance enforcement in real-time.

Conclusion

The proposed Adaptive Flame-Sailfish Optimization (AFSO) framework effectively solves the significant challenge of real-time face mask detection in IoT-enabled environments. This framework combines the ResNet50 and MobileNetV2 architectures to make a good solution for finding face masks in real time in IoT environments. We have comprehensively tested it on three sets of data—Kaggle, Public Places, and Public Videos—to demonstrate its accuracy, sensitivity, and efficiency. Results of these tests suggest considering AFSO as an upgrade from the ground up for mask detection in an Internet-of-Things environment. Efforts were rewarded handsomely, with the highest dataset accuracy hitting 97.8 %. In particular, we observed its robustness under severe conditions like occlusion and a variable light field, or its stability in dynamic environments. Hence, the proposed model guarantees real-time processing speeds within reason from constrained devices located at the periphery of IoT edge networks currently under construction. Its lightweight design is helping in overcoming the edge computing bottleneck. The work has demonstrated that the framework is effective. However, it also points out shortcomings, such as the need for a more diverse dataset and greater adaptability to dynamic environments. On the whole, the model proposed here is a significant leap for automated public health monitoring systems and provides a basis for true efficiency at scale.

Ethics statements

Compliance with Ethical Standards: This study was conducted in accordance with established ethical guidelines for research in artificial intelligence and IoT-enabled systems. No unethical practices were employed during the development, implementation, or evaluation of the proposed method.

Human Subject Consideration: The datasets used in this research (Kaggle Face Mask Dataset, Public Places Dataset, and Public Videos Dataset) consist of publicly available, anonymized data. No identifiable personal information or sensitive data were collected or processed, ensuring the privacy and rights of individuals were protected.

Credit author statement

Parul Dubey: Conceptualization, Methodology, Data Curation, Writing - Original Draft, Writing - Review & Editing, Validation, Visualization. **Vinay Keswani:** Software, Formal Analysis, Investigation, Writing - Review & Editing, Resources, Supervision, Project Administration. **Pushkar Dubey:** Supervised the work and contributed in literature survey. **Gunjan Keswani:** Validation, Visualization & Investigation. **Dhananjay Bhagat:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] J.C. Prata, A.L.P. Silva, A.C. Duarte, T. Rocha-Santos, Disposable over reusable face masks: public safety or environmental disaster? Environments 8 (4) (Apr. 2021) 31, doi:10.3390/environments8040031.
[2] S. Varshini, M. Rao, K. Ramesh, Smart IoT door for face mask and body temperature detection, IEEE Int. Things J. 8 (7) (2021) 5134–5141.
[3] Y. Kong, S. Zhang, J. Li, Face mask detection using faster R-CNN in public health systems, J. Med. Syst. 45 (2) (2021) 1–12.
[4] M.A.A. Radia, M.K.E. Nimr, A.S. Atlam, IoT-based wireless data acquisition and control system for photovoltaic module performance analysis, e-Prime - Adv. Elect. Eng. Elect. Energy 6 (2023) 100348 Dec., doi:10.1016/j.prime.2023.100348.
[5] A. Sethi, A. Gupta, P. Sharma, Face mask detection using deep learning, Int. J. Comput. Appl. 182 (41) (2021) 25–32.

[6] L. Chen, et al., IoT Microservice deployment in Edge-Cloud Hybrid environment using Reinforcement Learning, IEEE Int. Things J. 8 (16) (2020) 12610–12622 Aug., doi:10.1109/jiot.2020.3014970.
[7] M. Koklu, E. Balci, E. Karaca, Face mask detection using VGG16-BiLSTM model, Appl. Soft Comput. 104 (2022) 107341.
[8] W. Rahman, et al., IoT and blockchain-based mask surveillance system for COVID-19 prevention using deep learning, Comput., Mater. Continua/Comput., Mater. Continua (Print) 72 (1) (2022) 2033–2053 Jan., doi:10.32604/cmc.2022.025025.
[9] A. Choudhury and K.K. Sarma, "Facemask wearing correctness detection using deep learning approaches," in Lecture Notes in Networks and Systems, 2023, pp. 243–249. doi: 10.1007/978-3-031-34127-4_23.
[10] A. Barnawi, P. Chhikara, R. Tekchandani, N. Kumar, B. Alzahrani, Artificial intelligence-enabled internet of things-based system for COVID-19 screening using aerial thermal imaging, Future Generat. Comput. Syst. 124 (2021) 119–132 May, doi:10.1016/j.future.2021.05.019.
[11] Z. Chen, J. Cui, and Y. Zhang, Efficient mask recognition based on MobileNet. 2023, pp. 194–197. doi: 10.1109/aiotsys58602.2023.00050.
[12] D. Loey, F. Smarandache, M. Khalifa, YOLO v2-based real-time face mask detection, J. Big. Data 8 (1) (2021) 1–17.
[13] S. Nagrath, P. Jain, S. Kataria, SSD-MobileNetV2-based face mask detection, J. Comput. Vision Image Understand. 207 (2021) 104191.
[14] A. Taha, M. Eldeen, Hybrid deep learning and SVM-based model for face mask detection, Expert Syst. Appl. 184 (2021) 115709.
[15] V.Q. Vu, M.-Q. Tran, M. Amer, M. Khatiwada, S.S.M. Ghoneim, M. Elsisi, A practical hybrid IoT architecture with Deep learning technique for healthcare and security applications, Information 14 (7) (2023) 379 Jul., doi:10.3390/info14070379.
[16] R.K. Shinde, Md.S. Alam, S.G. Park, S.M. Park, N. Kim, Intelligent IoT (IIoT) device to identifying suspected COVID-19 infections using sensor fusion algorithm and real-time mask detection based on the enhanced MobileNetV2 model, Healthcare 10 (3) (2022) 454 Feb., doi:10.3390/healthcare10030454.
[17] R.A.S. Naseri, A. Kurnaz, H.M. Farhan, Optimized face detector-based intelligent face mask detection model in IoT using deep learning approach, Appl. Soft Comput. 134 (2022) 109933 Dec., doi:10.1016/j.asoc.2022.109933.
[18] Y. Zhang, A. Al-Ataby, and F. Al-Naima, A deep learning-based tool for face mask detection and body temperature measurement. 2022, pp. 70–75. doi: 10.1109/icspis57063.2022.10002688.
[19] K. Paulraj, N. Soms, S.D.S. Azariya, S. P. S, J. E. J, and V. Sureshkumar, Smart Healthcare monitoring System: integrating IoT, Deep learning, and XGBoost for real-time patient diagnosis. 2023. doi: 10.1109/ocit59427.2023.10431108.
[20] N. Alharbe and M. Almalki, "IoT-enabled healthcare transformation leveraging deep learning for advanced patient monitoring and diagnosis," Multimedia Tools and Applications, Jul. 2024, doi: 10.1007/s11042-024-19919-w.
[21] "Face Mask Detection," Kaggle, May 22, 2020. https://www.kaggle.com/datasets/andrewmvd/face-mask-detection
[22] M.Y. Mehmood, et al., Edge computing for IoT-enabled Smart grid, Security Communic. Networks 2021 (2021) 1–16 Jul., doi:10.1155/2021/5524025.
[23] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, I. Traore, A survey on IoT-enabled smart grids: emerging, applications, challenges, and outlook, Energies 15 (19) (2022) 6984 Sep., doi:10.3390/en15196984.
[24] Z. Lin, et al., Digital holographic microscopy phase noise reduction based on an over-complete chunked discrete cosine transform sparse dictionary, Opt. Lasers Eng. 166 (2023) 107571 Mar., doi:10.1016/j.optlaseng.2023.107571.
[25] N.D. Nandan, J. Kanungo, A. Mahajan, An error-efficient Gaussian filter for image processing by using the expanded operand decomposition logarithm multiplication, J. Ambient. Intell. Humaniz. Comput. 15 (1) (2018) 1045–1052 Jul., doi:10.1007/s12652-018-0933-x.
[26] H. Chen, G. Zhou, W. He, X. Duan, H. Jiang, Classification and identification of agricultural products based on improved MobileNetV2, Sci. Rep. 14 (1) (2024) Feb., doi:10.1038/s41598-024-53349-w.
[27] Y. Pan, et al., Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases, Front Physiol 14 (2023) Feb., doi:10.3389/fphys.2023.1126780.
[28] H. Zamani, M.H. Nadimi-Shahraki, S. Mirjalili, F.S. Gharehchopogh, D. Oliva, A critical review of moth-flame optimization algorithm and its variants: structural reviewing, performance evaluation, and statistical analysis, Arch. Comput. Meth. Eng. 31 (4) (2024) 2177–2225 Feb., doi:10.1007/s11831-023-10037-8.
[29] U.M. Khaire, R. Dhanalakshmi, K. Balakrishnan, M. Akila, Instigating the sailfish optimization algorithm based on opposition-based learning to determine the salient features from a high-dimensional dataset, Int. J. Inf. Technol. Decis Mak. 22 (05) (2022) 1617–1649 Nov., doi:10.1142/s0219622022500754.
[30] V.S.K. Settibathini, A. Virmani, M. Kuppam, N. S, S. Manikandan, and E. C, "Shedding light on dataset influence for more transparent machine learning," in Advances in Computational Intelligence and Robotics Book Series, 2024, pp. 33–48. doi: 10.4018/979-8-3693-1355-8.ch003.
[31] O. Boiko, A. Komin, R. Malekian, P. Davidsson, Edge-cloud Architectures for Hybrid energy Management systems: a comprehensive review, IEEE Sens. J. 24 (10) (2024) 15748–15772 Apr., doi:10.1109/jsen.2024.3382390.