EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire Smoke Detection

V. C. Ranganayaki, Javid J. and Jaideep Reddy M.
Department of Computer Science and Engineering, St. Joseph's Institute of Technology, Chennai, Tamil Nadu, India

Keywords: Lightweight CNN, Real-Time Detection, Edge Computing, Fire, Smoke Detection, Scalable Solution.

Abstract: Edge Fire Smoke is the lightweight CNN model optimized toward real-time fire and smoke detection in video streams and especially toward edge computing. The system responds to the surging demand for effective solutions in fire prevention, industrial safety, urban fire monitoring, and forest fire management. Unlike the traditional solutions of reliance on centralized processing, Edge Fire Smoke exploits its lightweight architecture to function easily on devices such as surveillance cameras and drones, as well as IoT devices, without letting the latency on the fire and smoke pattern detection reduce. Besides, the model has also been trained in large, heterogeneous datasets ensuring robust performance against changing environmental conditions. It comes with adjustable sensitivity levels, so it can be configured to the specific application and operational requirement. Real- time alerting mechanisms are integrated so that users or alarms can be notified right away upon detection. Comprehensive logging capabilities enable recording of detection events for further analysis or audits. A user- friendly interface makes it possible to monitor and configure a system with minimal technical complexity, thereby making the technology available to users without much technical know-how. Edge Fire Smoke is cost- effective, scalable, and dependable proactive fire management. The deployment of this technology in edge environments reduces dependence on cloud infrastructure, thereby lowering costs while improving response times. The new system plays a great role in safeguarding lives, infrastructural facilities, and the environment against any fire risks.

1 INTRODUCTION

The critical use of real-time video streams in the field of fire and smoke detection with an urban, forest, or industrial context is their ability to track fires or smoke at any possible early stage of development. This saves lives, guards structures, and prevents further ecological damage. More often than not, traditional fire detection schemes depend on cloud processing; however, this sometimes introduces latency, incurs additional operational costs, and consumes vast amounts of computation. It is an innovative, lightweight CNN particularly designed to overcome these shortcomings of real-time fire and smoke detection. The concept of edge computing in Edge Fire Smoke is utilized to reduce dependence on cloud infrastructure, thus allowing the detection to be fast and efficient right on the device, for example, surveillance cameras, drones, and IoT sensors. Therefore, a decentralized approach means that the fire and smoke detection can be done in a highly delayed manner in environments that have limited computational power. The most important advantage of EdgeFireSmoke is its optimized architecture, making it efficient on resource- constrained edge devices. This makes it suitable for application in various applications, from smart cities to the industrial facility and forest management, among other wildlife monitorings. It has been heterogeneously trained on high datasets, therefore robustly reliable across very different environments. Whether it is a forest wildfire, an industrial chemical fire, or a city urban building fire, the EdgeFireSmoke is designed to detect with accuracy patterns of fires and smoke at an incredible speed. Additionally, EdgeFireSmoke features customizable sensitivity settings because it can be made very sensitive to specific environmental settings or operational requirements. This flexibility makes the system highly adaptable to be deployed in a variety of scenarios without sacrificing performance. The system also has real-time alerting that notifies users immediately upon detection of fire or smoke, thus enhancing response times and helping prevent

Ranganayaki, V. C., J., J. and M., J. R.
EdgeFireSmoke: A Novel Lightweight CNN Model for Real-Time Video Fire Smoke Detection.
DOI: 10.5220/0013909200004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 130-137
ISBN: 978-989-758-777-1

catastrophic outcomes. The model's user-friendly interface allows even those without technical expertise to easily configure and monitor the system. Apart from this, the logging feature allows wide detection and audit events along with further events analysis. Altogether, EdgeFireSmoke is scalable, dependable, and economical for the solution of proactive fire management. It promotes the safety issue, disaster prevention, protection of the environment, lower operating cost, and response speed through its capabilities to equip an edge device with live real-time detection of fires and smoke. EdgeFireSmoke stands out as a pioneering solution in light of the increasing demand for efficient, real-time fire and smoke detection in various critical environments. It ensures fast and reliable performance even on resource-constrained devices like IoT sensors, drones, and cameras, owing to its lightweight CNN model.

2 LITERATURE SURVEY

The study of Krizhevsky et al. (2012) "ImageNet Classification with Deep Convolutional Neural Networks" initiated deep Convolutional Neural Networks, which have been responsible for the breakthrough in the image classification and, ultimately, the basis for fire and smoke detection in real time. AlexNet contains many convolutional layers for the purpose of automatic learning of raw image hierarchical features and, consequently, has delivered better accuracy than traditional machine learning approaches. For example, their work with a large dataset of images such as ImageNet proved that deep CNNs can indeed be quite powerful for real-time video analysis, especially in safety-critical applications such as fire detection.

Redmon et al. (2016) brought out a highly influential paper titled "You Only Look Once: Unified, Real-Time Object Detection." In that research, they developed the framework YOLO and revolutionized object detection through converting the problem into one regression. In contrast, in traditional systems, the images pass through several stages of scan. YOLO evaluates the whole image in just one forward pass, which improves the speed and efficacy. This efficiency works much in real-time fire and smoke detection, as alert is vital at the time of such incidences.

Simonyan and Zisserman (2015) in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" introduced VGGNet, a deep CNN architecture that is known for its simplicity and depth. This architecture achieved high accuracy in the recognition of images, as it uses small convolutional filters with 3x3 dimensions and up to 19 layers that allow it to capture very fine details in images. Such is important because, in detecting fire and smoke, very early signs of such events appear as very faint smoke or small flames that most existing systems fail to detect.

The very first proposal in their pioneering paper "Fire Detection in Video Sequences Using a Generic Color Model" for the possible use of color and motion features in fire detection from video sequences was given by Celik et al. in 2007. The authors were able to develop a model that, based on the reddish-yellow color of flames, was able to identify fires and incorporated motion detection to discern dynamic flames from static sources of light, thereby preventing false positives. This work was among the first to use video-based analysis for fire detection and has been the basis of modern systems that use machine learning and CNNs.

Toreyin et al. (2005) proposed a paper "Computer Vision Based Method for Real-Time Fire and Flame Detection," where the method discovers the unique spatiotemporal properties of flames, especially flicker frequency, that can be used to isolate flames from other moving entities in a scene. It is one of the first works that utilize temporal patterns during fire detection since flames exhibit some motion characteristics that may be observable in video frames. In modern CNN-based systems in EdgeFireSmoke, Toreyin's design does not use colour or intensity thresholds solely; it uses the dynamical behaviour of fire.

In the paper "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Sandler et al. in 2018 proposed a lightweight CNN architecture optimized for mobile and edge computing on limited computational resources. The MobileNetV2 model based on inverted residuals and linear bottlenecks has scaled down the parameters drastically, therefore reducing the computation cost as well. It is a good choice for real-time fire and smoke detection in edge devices such as surveillance cameras and IoT sensors mainly because they work in resource-constrained environments, and such efficiency can be allowed without accuracy loss.

Iandola et al. (2016), in the paper "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size," presented an extremely efficient CNN architecture, significantly cutting down memory usage with great performance. SqueezeNet uses 1x1 convolutions and strategic downsampling to achieve AlexNet performance while having fewer parameters; thus, it is great for use on fire and smoke detection on edge devices which have fewer resources. With a low-memory design, the EdgeFireSmoke project could make real-time deployment possible on resource-constrained devices like drones and cameras for speedy detection and response to emergencies caused by fire. Kim et al. (2016), in their paper "Real-Time Fire Detection Based on Image Processing," proposed a method that utilized color segmentation and motion analysis for the detection of fire in real-time. This decreases false positives due to static light sources because of the identification of reddish-yellow tints typical of flames as well as motion detection. This method is best used in dynamic environments such as industrial sites, forests, or even cities. For an EdgeFireSmoke, for example, adding color and motion features to the CNN model will be great for improvements in detection.

Ma et al. (2020), implemented lightweight architecture of MobileNet to deploy fire detection in the edge device with the algorithm of "Efficient Video Fire Detection Algorithm Using MobileNet". It has proven the ability of doing real-time fire detection at surveillance cameras, or IoT sensors that run in a low power condition at the very edge of any location including harsh environment. It ensures running the fire detection systems across all environments as computational resources do not have to be reduced without sacrificing precision. This will ensure immediate, cloud- agnostic detection and response for EdgeFireSmoke in any applications requiring real-time fire management.

Yuan et al. (2020), in their survey "Survey on Deep Learning-Based Fire and Smoke Detection Techniques," surveyed various deep learning approaches wherein it is identified that challenges in fire and smoke detection exist because of the varied conditions of visual aspects and the requirement of real-time processing. Techniques like GANs have been applied to generate synthetic data. Hybrid systems use a combination of CNN-based detection along with sensor data for improvement in accuracy. give for the conference of the paper.

3 PROBLEM STATEMENT

Rising instances of security threats in public domains such as schools, airports, and transportation stations necessitate immediate real-time detection solutions. Manual monitoring methods have limitations with traditional surveillance technologies since it takes some considerable time before a threat can be detected; therefore, response time increases along with risks. Most available automated detection technologies are cost-prohibitive and call for hardware that may not be found in many public environments. This project proposes a low-cost, scalable, and precise weapon detection system using OpenCV, an open-source library for computer vision, and Django, the Python-based web framework. Since Django has the features that help in easy deployment, user-friendly interfaces, and real-time notifications, it pairs perfectly with OpenCV, which is known for fast video and image processing. It would combine pre-trained deep learning models such as YOLO (You Only Look Once), and even SSD (Single Shot MultiBox Detector), distinguish weapons from other non-threatening objects of interest in real-time video stream with great accuracy. With OpenCV and Django, the system ensures that those processes involved in detection, alerting, and data logging run inside a user- friendly interface, suitable for use on standard hardware. It simply means that using related advanced, affordable technologies to augment public safety is a great idea, which enables security personnel with real-time actionable insight. Such an architecture would be practical, efficient, and easily deployable.

4 METHODOLOGY

4.1 System Overview

A general overview of the system is as presented by the block diagram in the figure 1 system as follows:

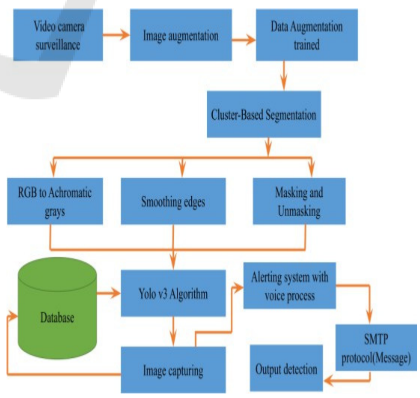

Figure 1: Architecture diagram of weapon detection.

The EdgeFireSmoke system makes use of the real-time fire and smoke detection ability via a lightweight CNN optimized to run on devices with small computational powers. The detection is achieved using video streams sourced from surveillance cameras, drones, or IoT sensors in highly dynamic environments. It therefore integrates the use of pre-trained CNN models on large heterogenous datasets that can very efficiently recognize the patterns of fire and smoke with very high accuracies. The model is optimized for low-latency processing to make sure that fire detection is in real- time. That way, it reduces dependency on cloud-based servers to make the system work standalone in remote or resource-constrained settings.

In terms of detection, EdgeFireSmoke operates as follows: every frame from a video is processed with the aid of a pre-trained CNN that extracts features of the image such as color, texture, and patterns in movement. These features then lead to the identification of the presence of fire or smoke through color segmentation, motion analysis, and spatiotemporal feature extraction. It possesses real-time alerting mechanisms which automatically alert users to the case of detection with logging facility to analyze events and audit them.

4.2 Data Collection Module

Data Collection Module of the EdgeFireSmoke is crucial to the acquisition of the quality diversification of gathered data to properly train a CNN for an accurate real-time detection of fire and smoke. In this module, the system gathers video streams from diverse sources including but not limited to, the surveillance cameras, drones, as well as IoT sensors that spread throughout the following diverse environments: urban and industrial sites, and forests. It has several visual conditions that involve capturing various types of fires and smoke, lighting variations, changes in weather, and environmental settings to make the data more robust and generalizable. The module captures video frames at a high resolution, thus saving the fine-grained details of fire and smoke patterns. It also possesses many features to identify whether it is fire, smoke, or any other object with some degree of accuracy, such as color of flame (reddish-yellow colors), characteristics of motion (flickering and movement), and environmental conditions (fog, reflections, and lighting conditions). Information from sensors, for example, temperature or gas concentration measurements, can also be included in order to supplement video information to add a layer of verification and eliminate false positives. Data once acquired undergoes preprocessing. Normalizing video frames and image resizing are two standard preprocessing techniques into a CNN model. Others include data augmentation in a way that increases diversity as well as simulate different environmental conditions. The dataset is then further divided into training, validation, and testing subsets, so the model is tested on multiple scenarios, thereby making the model more robust and generalized. To summarize, the Data Collection Module ensures that the EdgeFireSmoke system is trained in a variety of conditions; that is, it provides real-time fire and smoke detection across different environments in a reliable and accurate manner.

4.3 Preprocessing Module

Pre-processing module of the EdgeFireSmoke system is very essential because it pre-processes the video data fed into CNN. It is standardized noise-free and ready for proper training, hence ensuring that real-time inference is feasible. The first step among all the preprocessing steps is to undergo image resizing, wherein video frames are transformed into a certain dimension, so they can be presented to the CNN for further operations; therefore, hence reducing computation time. Normalization: This makes pixel values fall within the range 0 to 1 or -1 to 1, which offers a similar standard for diverse input data and results in a better convergent rate of the network on training. Data Augmentation: Increase data varieties and robustness from changed conditions by using augmentation technique such as rotation, flips, color change, and cropped images. This simulates different environmental conditions such as lighting changes, vantage or partial occlusions. Noise reduction is also provided. And this module eliminates all the noise created by extraneous visual information that degrades performance in a model, for example, clutter in the background, unwanted reflections. Motion analysis has also been included for identifying static features and dynamic features with focus on motion patterns characteristic of fire and smoke. The preprocessed data is then labeled in supervised learning, whether the fire or smoke is there, so that the system learns to distinguish the elements within real-time video.

4.4 Model Selection Module

The Model Selection Module of the EdgeFireSmoke system is supposed to determine which one of these CNN architectures will be used for the purpose of real- time fire and smoke detection under specific conditions. Given the limits of an edge device, it has to operate under computational and power constraints, making the selection of some of the available lightweight yet accurate CNN models that provide the best accuracy with a lower computational cost. These purposes require models like MobileNetV2, SqueezeNet, and VGGNet due to their proven efficiency in managing the image classification tasks along with a resource-friendly model. Module trade- offs for detection accuracy, size, and inference time help with performance metrics. Model comparison based on comparative evaluation involves further considering how each architecture may cope with changing environmental conditions: light variations, smoke density, and flame characteristics. It also ensures real-time processing needs. The selected model will then work on video frames as quickly as possible without causing any delays. After the selection, the model is fine-tuned on the preprocessed data with the objective of improving performance in specific fire and smoke detection tasks.

4.5 Fire Detection and Classification

This is a smoke detection and visual elements classification component of the EdgeFireSmoke system, detecting and classifying visual elements within real- time video streams in real time. It applies a pre-trained CNN that analyzes video frames based on some characteristics unique to smoke: color, texture, and movement patterns. Smoke generally appears as a partly misty or translucent cloud that assumes irregular shapes and has diffused edges. It will, therefore, differ in intensity and color, depending on the fuel source and environmental factors. It is this subtle differentiation that the CNN model becomes capable of as it learns from the diverse nature of the dataset it is fed on. The diversity of such smoke in the dataset with varying illumination, weather, and environmental factors would be different. First, at detection, it separates the areas that might be potential smoke by using the color segmentation technique to identify whether they are shades of gray, white, or black colors that are most associated with smoke. Motion analysis further differentiates the smoke from static objects because of their dynamic and swirling motion. Once the possible smoke areas have been identified, the CNN will classify them based on learned patterns to determine whether they really show smoke or are false alarms like fog or vapor.

4.6 Alert System Integration

The Alert System Integration of the EdgeFireSmoke system has been designed to send alerts to users or automation systems as soon as possible when fire or smoke is detected. When the model identifies and classifies a possible fire or smoke in a video stream, an alert system calls a series of operations to take place in response so as to act quickly. The system uses the detection results in real-time and generates notifications through various channels, such as SMS, emails, or direct communication with the central monitoring systems. Other than basic alerts, the system may integrate with other IoT devices, such as sprinklers, alarms, or fire suppression systems, and may enable an automated response to detected threats. It has the characteristic of configurability in terms of alert sensitivity so that it will not lead to false positives and hence give reliable detection. Information regarding the type of danger, whether it is fire or smoke, location, and time related to alerts is of utmost importance to the emergency responders.

4.7 Backend System

The backend system should be strong enough to integrate real-time fire and smoke detection using the YOLO-based model; therefore, the computational power needed for the deployment of deep learning models and large-scale data can be provided through cloud infrastructure like AWS or Google Cloud. Apache Kafka or AWS Kinesis may be used to continuously ingest real-time video streams, which could have minimal latency, with which to manage real time video streams. Tools, for instance, TensorFlow Serving, or TensorRT will implement the YOLO model at inference time for enhanced performance optimization. Additionally, TensorFlow Lite may be useful for optimizing models for Edge Devices that have low resource. The video data will be stored in either Amazon S3 or Google Cloud Storage, while metadata and other events will be taken care of by MongoDB. The communication between edge devices and the backend will be handled by restful API or AWS API Gateway, ensuring alert and data transmission fluidly. The scalability, real-time processing, and fire and smoke detection will be supported in the backend architecture on the very resource-constrained device.

4.8 System Testing

The testing of the EdgeFireSmoke project includes functional, performance, and reliability testing of the entire fire and smoke detection system. This testing will validate all the real-world operating capabilities of the YOLO-based detection model, preprocessing, classification modules, and the alert system. It also includes the validation of the system by varied datasets and simulating changing environmental conditions, like light, weather, and density of smoke to check accuracy and robustness. Real-time testing for the edge devices, such as cameras and drones, is under way to test detection and alerting almost for free from latency. Integration tests are performed to ensure seamless communication with the edge device, with the backend infrastructure, and through the alert system. A stress test is conducted, determining how well the system behaves with its data loads and high video frame rates. Such tests are then used as feed back for further refining the performance of models and responsiveness of systems.

4.9 Deployment

The EdgeFireSmoke system is sending the trained YOLO-based detection models to edge devices, including surveillance cameras, drones, or IoT sensors for real-time execution. Coupled with lightweight frameworks, such as TensorFlow Lite or ONNX Runtime, that enable execution on the processing resource-less devices, this system makes central monitoring and alert management the role of cloud services, like AWS IoT Core or Google Cloud IoT. It encompasses the establishment of the alerting system for instant alerts via SMS, email, or app-based alerts. The system is fully tested post-deployment to ensure stability in live environments. Scalability is achieved due to containerization tools such as Docker whereby the system can be very quickly replicated across different locations, thereby offering better coverage fire and smoke detection. All this involves installation on the edge devices, whereby compact models like MobileNet or YOLO exist for real-time detection of events. The device is pre-configured to send alerts in an autonomous form through the integrated communication protocols that support SMS, email, or through mobile apps. Deployment also includes calibration of the system for fitment into various environmental setups, thus avoiding minimal cases of false alarms.

5 RESULT AND OUTPUT

The EdgeFireSmoke system provides accurate, efficient real-time detection of fire and smoke based on rigorous testing conditions. Equipped with a YOLO-based model and advanced preprocessing, the system obtained great detection accuracy on fire and smoke patterns while producing minimal false positives and negatives. It worked well under adverse conditions such as low-light settings and in changing weather, and even partially obscured, it gave

Figure 2: Complete detection of a fire.

and adaptability. There is the real-time video frames that are annotated with marks to indicate there's fire or smoke to help fasten the review and subsequent response. In addition, alerts are received immediately over SMS, email, and through an application notification that nobody is left unaware. Because of its capabilities, its effectiveness in environments with resources constraint without a cloud was a guarantee. In this, it is the best in high-risk places and inaccessible locations. This performance was maintained at high video frame rates and large data sets thus indicating scalability and responsiveness. Full logging and report generation with all the above features ensures very good insight into post-event analysis and audits. Thus, the results ensure the system is a reliable and proactive solution for fire and smoke emergencies, offering a scalable, low-cost approach for industrial, residential, and natural environments. It is through quick detection and alerting that EdgeFireSmoke crucially goes along in preventing and mitigating fire- related disasters. Figure 2 shows the complete detection of a fire.

6 PERFORMANCE ANALYSIS

In reflecting the efficiency and accuracy in fire and smoke detection performance analysis, it shows under any scenario, as it underwent rigorous testing of its heterogeneous dataset on extreme conditions ranging from industrial sites to forest environments, urban and household scenarios. The model YOLO-based demonstrated precision and recall values close to perfect even at the conditions of low illumination, smokes at occlusion, and dynamism in weather conditions. In experiments, such a low-latency high-performance framework could directly analyze multiple input sources of high frame rates onto these miniature edge devices constituting IoT sensors and surveillance cameras. A huge amount of data flow went through the system while still managing to provide the desired scale of performance.

Resource optimization leads to a lightweight model with efficient preprocess without losing any precision in case of detection for its ease-of-deployment with constrained- resource devices. With comparative benchmarks, EdgeFireSmoke detected more compared to the traditional detection system, in terms of detection rate, response time, and adaptability of varying environmental conditions. In addition, real-time alerting systems as well as the logging mechanism increased its usability; it brought actionable insights into the hands of the user. The given results confirm that EdgeFireSmoke is reliable, efficient, and robust for fire and smoke detection, that makes it a really valuable tool for safeguarding life and assets in critical safety applications.

7 CONCLUSION AND FUTURE WORKS

The EdgeFireSmoke has been well defined by the advancement that fire and smoke detection by real-time means are possible through light versions of CNN models by virtue of edge computing. The strong capabilities of such high accuracy with responses in real time, along with adaptability into any setting, make for a highly reliable solution for this risk to be mitigated by fire. This system can maximize the use of resources and is useful when integrated with edge devices, for example, drones, sensors, and cameras in supporting fast detection and alerting while being light on resources. Full logging and analytics facilities make this system extremely valuable for the assessment after the event, garnering precious insights. Future work: The system would improve further by incorporating multimodal data, for example heat, gas, or acoustic sensors in enhancing detection accuracy while suppressing false alarms. It is in the more complex situation that advanced deep learning methods like transformer-based models are used to enhance the performance of the system. Predictive analytics can be included for risk detection of fire occurrence based on patterns within environmental data. The system called EdgeFireSmoke will serve as a bedrock for future innovations and challenges in fire safety technologies as it responds to growing concerns through innovative solutions.

REFERENCES

A. Boukerche and E. Nakamura, "Localization systems for wireless sensor networks", IEEE Wireless Communications, vol. 14, no. 6, pp. 6-12, 2007. doi: 10.1109/mwc.2007.4407221.

D. S. Thomas, D. T. Butry, S. W. Gilbert, D. H. Webb, and J. F. Fung, "The Costs and Losses of Wildfires," National Institute of Standard and Technology (NIST), Tech. Rep., 2017.

F. Khelifi, A. Bradai, K. Singh, and M. Atri, "Localization and Energy-Efficient Data Routing for Unmanned Aerial Vehicles: Fuzzy-Logic-Based Approach," IEEE Communications Magazine, vol. 56, no. 4, pp. 129-133, 2018. doi: 10.1109/mcom.2018.1700453.

K. Hoover, L. A. Hanson, "Wildfire Statistics 2019 | Centre for Research on the Epidemiology of Disasters," Federal Assistance for Wildfire Response and Recovery, 2019. Fas.org. (2019). [Online] Available at: https://fas.org/sgp/crs/misc/IF10244.pdf [Accessed 26- May-2019].

M. Y. Arafat and S. Moh, "Location-Aided Delay Tolerant Routing Protocol in UAV Networks for Post- Disaster Operation," IEEE Access, vol. 6, pp. 59891- 59906, 2018. doi: 10.1109/access.2018.2875739.

M. Y. Arafat and S. Moh, "Routing Protocols for Unmanned Aerial Vehicle Networks: A Survey," IEEE Access, vol. 7, pp. 99694 - 99720, 2019. doi: 10.1109/ACCESS.2019.2930813.

M. Y. Arafat, S. Poudel and S. Moh, "Medium Access Control Protocols for Flying Ad Hoc Networks: A Review," IEEE Sensors Journal, pp. 1-1, 2020. doi: 10.1109/jsen.2020.3034600.

M. Y. Arafat, M. Habib and S. Moh, "Routing Protocols for UAV-Aided Wireless Sensor Networks," Applied Sciences, vol. 10, no. 12, p. 4077, 2020. doi: 10.3390/app10124077.

N. H. Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 899922, 2016. doi: 10.1109/jiot.2016.2612119.

Q. Guo, Y. Zhang, J. Lloret, B. Kantarci and W. Seah, "A Localization Method Avoiding Flip Ambiguities for micro-UAVs with Bounded Distance Measurement Errors," IEEE Transactions on Mobile Computing, pp. 1-1, 2018. doi: 10.1109/tmc.2018.2865462

S. Yang, D. Enqing, L. Wei and P. Xue, "An iterative method of processing node flip ambiguity in wireless sensor networks node localization," in Proceedings of International Conference on Information Networking (ICOIN), pp. 92 97, 2016. doi: 10.1109/icoin.2016.7427094.

Z. Lin, H. Liu and M. Wotton, "Kalman Filter-Based Large-Scale Wildfire Monitoring With a System of UAVs," IEEE Transactions on Industrial Electronics, vol. 66, no. 1, pp. 606-615, 2019. doi: 10.1109/tie.2018.2823658.