Journal of Intelligent Systems and Internet of Things
Vol. 18, No. 01, PP. 250-259, 2026
DOI: https://doi.org/10.54216/JISIoT.180119

Climate Change Prediction in Urban Environment Using UAV Imaging Based on Cloud IoT and Deep Learning Techniques

M. Prema Kumar[1,*], P. Chinnasamy[2], B. Bala Abirami[3], Juvvala Sailaja[4], S. Bhuvana[5], Sai Krishna Vunnam[6]

[1]Professor, Dept. of ECE, Shri Vishnu Engineering College for Women (A), Bhimavaram, Andhra Pradesh, India

[2]Associate Professor, Department of Computer Science and Engineering, School of Computing, Kalasalingam Academy of Research and Education, Srivilliputtur, India

[3]Assistant Professor, Department of Artificial intelligence and Data Science, Panimalar Engineering College, Chennai, India

[4]Assistant Professor, Department of Information Technology, Aditya University, surampalem, Andhra Andhra Pradesh, India

[5]Research Scholar, Department of Computer Science and Engineering, Dr.MGR Educational and Research, India

[6]Associate Professor, Department of Computer Science and Engineering, MLR Institute of Technology, Hyderabad, India

Emails: medapatipremakumar@gmail.com; chinnasamyponnusamy@gmail.com; bala.bami@gmail.com; sailu.sailaja130@gmail.com; institute,bhupreethi@gmail.com; vskrishna@mlrit.ac.in

Abstract

Advancements in Unmanned Aerial Vehicles (UAVs), popularly identified as drones, offer unprecedented opportunities to improve various applications of Extensive Internet of Things (IoT). In this framework, Deep Learning (DL) techniques are considered a practical alternative for improving the real-time obstacle detection and avoidance performance of fully autonomous UAVs. This research propose novel technique in urban environment climate change detection utilizing UAV image based on cloud IoT with deep learning model. Here the UAV images has been collected through cloud IoT module and prepared for dataset. This dataset with UAV images has been processed for filtering and contour reduction by normalization. Then processed image features are extracted utilizing graph cut fuzzy convolutional ResNet attention neural network with moath firefly sparrow colony optimization model. The simulation results has been analyzed for various UAV dataset in terms of training accuracy, average precision, recall, QoS, scalability. Proposed technique Average precision of 97%, QOS of 92%, SCALABILITY of 96%, training accuracy of 98%, RECALL of 95%.

Received: February 18, 2025 Revised: June 02, 2025 Accepted: July 05, 2025

Keywords: Urban environment; Climate change detection; UAV image; Cloud IoT; Deep learning model

1. Introduction

Natural disasters account for an average of 60 thousand deaths every year, which comprises 0.1 percent of all deaths in the world. Such natural disasters comprise landslides, storms, earthquakes, floods, etc. Among these calamities, floods are most common and account for 40% of natural calamities across the globe. The major reasons for the rapid increase in exposure to flood hazards are global warming, cyclones, heavy rains, ice cap melting, and winter blizzards. Likewise, floods account for thousands of lives lost and average damages amounting to billions of US dollars. Floods

are not only deadly but also destructive to property, infrastructure, crops, livestock, and agricultural lands, which results in enormous economic losses that need to be mitigated in an era of sustainability and smart city strategies [1]. The precipitations' flood-damages rose remarkably within these years, from USD 6 billion to USD 10 billion in excess of extreme rainfall events registered in entire first decade of new millennium. Because of this, huge amounts of money have been allocated in order to design effective flood management systems. The economic growth of the country that has been affected is also encumbered by the costs of related emergency response, reconstruction, and their provision. A report published in 2016 on the economic losses incurred due to floods suffered in various regions in the year 2012 estimates that the total cost of floods globally amounts to USD 19 billion. There are people who die because the interventions and recovery systems are not put in place in good time as there is late warning about floods or there are no effective technologies capable of flood sensing in advance. It brings out the importance of the modern day technologies, which can enable the mapping of the flood-affected regions within the shortest period possible in order to initiate the intervention measures. [2] It is important to do proper flood management and to carry out the rescue of the stranded people as quickly as possible so as not to incur heavy losses in terms of both money and human life. A key element that offers crucial decision support and analytical capabilities for disaster management is the Geographic Information System (GIS). It makes it possible for the authorities to gather, store, handle, and evaluate geographic or spatial data in order to respond appropriately to disasters. Better flood detection is made possible by GIS's ability to automatically identify areas affected by flooding and combine the findings with existing geographic data [3]. It has been used to identify possible transportation options and rescue routes in places affected by flooding. However, this technology is very dependent on the disaster's information being available. Because such data is only accessible after a few days, if not weeks, emergencies like floods will be handled slowly. A GPS receiver may receive geolocation and time data from Global Positioning System (GPS), a global navigation satellite system (GNSS), anywhere on or close to Earth where four or more GPS satellites have an unhindered line of sight [4]. Disaster management and relief efforts following floods have made extensive use of this technology. The GPS sensor nodes that are placed on a building's roof and provide relative position data from both before and after flood disasters serve as one example. By mining deep semantic features from images, authors have demonstrated deep learning (DL) technology in detecting forest fires, increasing the accuracy of wildfire detection. DL technology has already been applied in the fields of intelligent agriculture, logistics, and indoor target positioning. Disaster management and relief efforts following floods have made extensive use of this technology. The GPS sensor nodes that are placed on a building's roof and provide relative position data from both before and after flood disasters serve as one example. By mining deep semantic features from images, authors have demonstrated deep learning (DL) technology in detecting forest fires, increasing the accuracy of wildfire detection. DL technology has already been applied in the fields of intelligent agriculture, logistics, and indoor target positioning. Building a deep-level network method is necessary to extract deep abstract characteristics from images, and deep neural network (FCNN) training is a difficult and time-consuming process. Furthermore, a large number of labeled cases were required for the deep model training. Two-stage networks and one-stage networks are the two types of classical deep neural networks used for object detection. Proposal regions must first be generated, and then they must be classified and located by two-stage networks like RCNN and Faster RCNN. Two-stage networks are suitable for applications requiring improved detection accuracy, as numerous studies have shown. One-stage networks, like Yolo and SSD, are quicker than two-stage networks and instantly coordinate positions and calculate class probabilities. Thus, in real-world UAV applications where high-speed needs are necessary, one-stage networks offer significant benefits. In a similar vein, various speedier lightweight networks exist, including ESPNetv2, Yolov3, MobileNet SSD, and others [5].

2. Research contribution

To propose an enhanced approach with UAV images in determining urban climate change which employs cloud IoT and deep learning networks. In this case, cloud IoT module was used to collect the UAV images that were subsequently processed to make a dataset. This entire UAV image bank has been collected, processed by normalization, and demising. The features of the processed image are then optimized using moath firefly sparrow colony optimization after the images have been obtained through a graph cut fuzzy convolutional ResNet attention neural network.

3. Related works

High-resolution spatial photographs from target sites can be obtained with the help of Unmanned Aerial Vehicles (UAVs) [6]. As smart technologies in industry 4.0 age, these UAVs are now frequently utilized in place of more conventional imaging instruments like satellite imaging as well as GPS-based monitoring. For faster, more intelligent, and better-informed replies, UAVs may swiftly gather exact picture data and send it to the appropriate off-site computers. In a similar vein, flood prediction and detection are increasingly being done using Artificial Neural Network (ANN) models. In order to create regional flood inundation maps during storm occurrences, study [7]

suggested a hybrid ANN-based model that combines a self-organizing map (SOM) as well as recurrent nonlinear autoregressive network with exogenous inputs (RNARX). According to authors, RNARX network can predict inundation depths, but 4 × 4 SOM network can cluster target area's inundation depths. In a similar vein, [8] created an early flood warning system that detects typhoon flood episodes and reliably predicts the depth and breadth of inundation by combining a hydrodynamic method, k-means clustering method, SVM. Systems based on fuzzy logic are also widely utilized and are utilized to predict river water levels and sound an early warning in the event of flooding [9]. With an accuracy of 80%, flood detection has been achieved using harmonic analysis as well as change detection on multi-temporal data. Similarly, Chobe floodplain in Namibia's Caprivi region was flooded to a certain amount using SAR pictures and a novel change detection and thresholding technique (CDAT) [10]. A Bayesian network has been suggested to combine remotely sensed information, including interferometric-SAR coherence data and multi-temporal SAR intensity images, with geomorphic and other terrestrial data like roads and structures [11]. Moreover, the Multilayer Perceptron (MLP), a technique of artificial neural networks (ANN) based on back-propagation, has been utilized to predict floods by analyzing rainfall time series data and water levels in a weir that may reach into the urban area [12]. Likewise, flood modeling has been done using a Wavelength Neural Network (WNN). The relevant literature so demonstrates that while deep learning is uncommon and has not been thoroughly tested or documented for such applications, image processing and ML methods are routinely employed for flood detection. [13] have reported numerous effective applications of autonomous navigation with UAVs guided by monocular views. However, you won't be able to sense depth as well if you only have one eye. Only a few recommendations were made regarding how to enhance UAV autonomous guidance in this circumstance. The aim of Structure from Motion (SfM) methods is to reconstruct scenes from the motion data of the UAV, as per the research design of [14]. From a small set of consecutive images, an established depth map was produced, which was subsequently used to generate waypoints [15]. Direct depth estimate was suggested since it made it possible to create intricate depth visualizations in real-time and made navigating in a crowded outdoor environment easier. The SfM-based obstacle-avoiding technique, however, is unable to shield UAV from moving targets that move during mapping cycles or while mapping itself. According to [16], the mapping cycles must save and compare the location's photos sequentially in order to obtain depth information.

4. Proposed cloud IoT model

Fig. 1 describes how UAVs interact with the different parts of cloud-IoT-based distributed computing method that provides video surveillance. There are three types of video surveillance systems that use UAV images: distributed, centralized, layered methods. Each type of system has pros as well as cons. For UAV placement, the placements and number of stopping arguments are instantly optimized. The problem presents a significant barrier for conventional gradient as there is no explicit description of slope vector and number of stopping points is unknown and not predetermined, necessitating variation during the optimization process. This issue can be resolved by evolutionary methods. In this study, a drone has been used to study the data-collection method of IoT. Here, drone placement is adjusted by simultaneously optimizing the placements and number of halting arguments, which improves IoT device services and results in more energy-efficient data collecting.

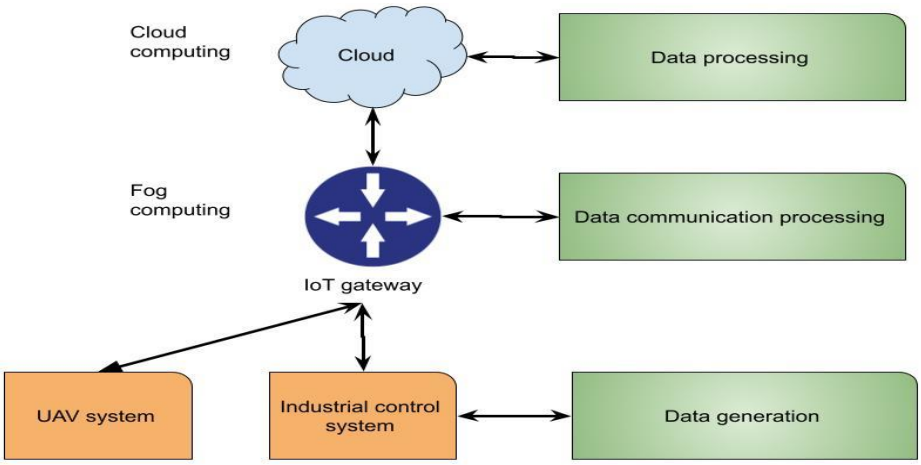


Figure 1. Interaction of UAV with cloud-IoT integrated distributed computing system

A rotary-wing UAV and a number of ground-based IoT devices with N=[1,2,…,n]N=1,2,…,n are part of conceptual method, which is an IoT data-collection method with aid of a drone. A UAV serves as a framework for gathering flight data from these IoT devices. Because of its mobility and flexibility, the UAV may relocate the stations' placements multiple times, increasing coverage while consuming less energy. In this case, it is assumed that there are k stopping places and that no previous knowledge exists. It is possible to set the breakpoints as K=[1,2,…,k].K=1,2,…,k.

Therefore, every IoT parameter chooses one breakpoint to which its data is sent. Because of the system's bandwidth constraint, furthermore, the UAV may identify IoT devices to transmit signals using eqn (3) at every stopping location.

$$C_3 = \sum_{t=1}^{m} a_{ij} \leq M \qquad (3)$$

The following formula is used in the suggested model to determine channel amplification between drone at jth stopping point and IoT device by eqn (4)

$$h_{tf} = h_0 a_{tj}^{-2} = \frac{h_0}{(X_J - X_t)^2 + (Y_f - Y_t)^2 + H^2} \qquad (4)$$

5. UAV image feature extraction using graph cut fuzzy convolutional ResNet attention neural network (GFCResAtNN)

the image can be considered as a graph Z = {V, E}, where V is the set of nodes and E is the set of edges of connected nodes. A cut on a graph refers to the separation of V into two subsets S and T as follows

$$s \in S, t \in T,\ S \cup T = V,\ S \cap T = \phi \qquad (5)$$

The two subsets stand for the foreground and background for image segmentation. Reducing the total of weights between the edges, the mincut idea is the most basic and often used graphcut strategy. The definition of the weighted edges is the most important part of the graph structure since it determines the actual shape of the energy function. The data item and smooth item, which represent the edge weights in graph cut cutting, make up the general energy function of graph cuts. In general, the graph cuts energy function definition is described in equation (4), where f is the label the node has allocated, meaning that the cost of a node from the source or sink is recorded as f.

$$E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) \qquad (6)$$

Among (6), A data item referred to as edata represents the difference between a node's value and the value of label f. To assess the smoothness of a segment, we use a term called "esmooth." Calculating smoothing items is a bit more complex than working with data items, which usually rely on a distance measurement to determine the degree of difference. The smoothing item needs to identify borders or changes while also being resilient to gradual transitions and noise. This characteristic is known as discontinuity preserving.

The second phase of the fuzzy logic inference system takes the crisp values from the first stage, which range between 30 and 70 percent, as its input. It can be quite tricky to tell apart low faults from medium noise, especially in the areas around the center of the tank. If the shade is positioned at the bottom, top, left, or right of the tank, the second level of fuzzy logic is set up to detect the intermediate noise it generates. Each of the four components goes through this process separately. The outputs from the first stage, falling between 30% and 70%, serve as the input for the second stage. The pixel density in areas that the first stage of fuzzy logic could not classify plays a crucial role in the second stage of the classification process. To tackle the unpredictable and often vague nature of decision-making issues, we turned to fuzzy membership functions. We utilized triangular fuzzy membership functions to convert the expert decision matrix data into fuzzy numbers (TFMFs). A triangular fuzzy number, defined by three parameters, is derived from a crisp number using TFMFs. Depending on the selected criteria; we applied a defuzzification method to yield a clear value that reflects the overall score for each alternative. The centroid approach, which identifies the center of gravity of the fuzzy number, was chosen as the most widely used defuzzification technique. We utilized Equation (7) along with earlier equations, applying TFMF to calculate the ratio of the fuzzification data.

$$\frac{\text{Imp}(\widehat{E1/C1})}{\sum_{j=1}^{n} \text{Imp}(\widehat{E1/CC_{1j}})} \qquad (7)$$

The mean values were computed to determine the final fuzzy values of the weight coefficients for the evaluation criteria (wf1,wf2,...,wnf) T. The final weight value of each criterion was calculated using Equation (8) and the fuzzy EDM (EDM)).

$$\widetilde{w_j} = \left(\sum_{i=1}^{m} \frac{\mathrm{Imp}(\widetilde{E_{ij}}/C_{ij})}{\sum_{j=1}^{n} \mathrm{Imp}(\widetilde{E_{ij}}/C_{ij})}\right)/m \quad (8)$$

This was further clarified by using the fuzzy membership function on the EDM data, followed by defuzzifying the fuzzy numbers for each criterion to obtain clear values that are suitable for further analysis. The centroid method, a widely used defuzzification technique, helped identify the center of gravity of the fuzzy number. You can find the centroid for TFMF using Equation 10. Once we had the centroid values for each criterion, we assigned a weight to each based on its relative importance. To ensure everything is balanced, the weights of all criteria should total 1. Finally, we multiplied the centroid value of each criterion by its weight to arrive at the final weight.

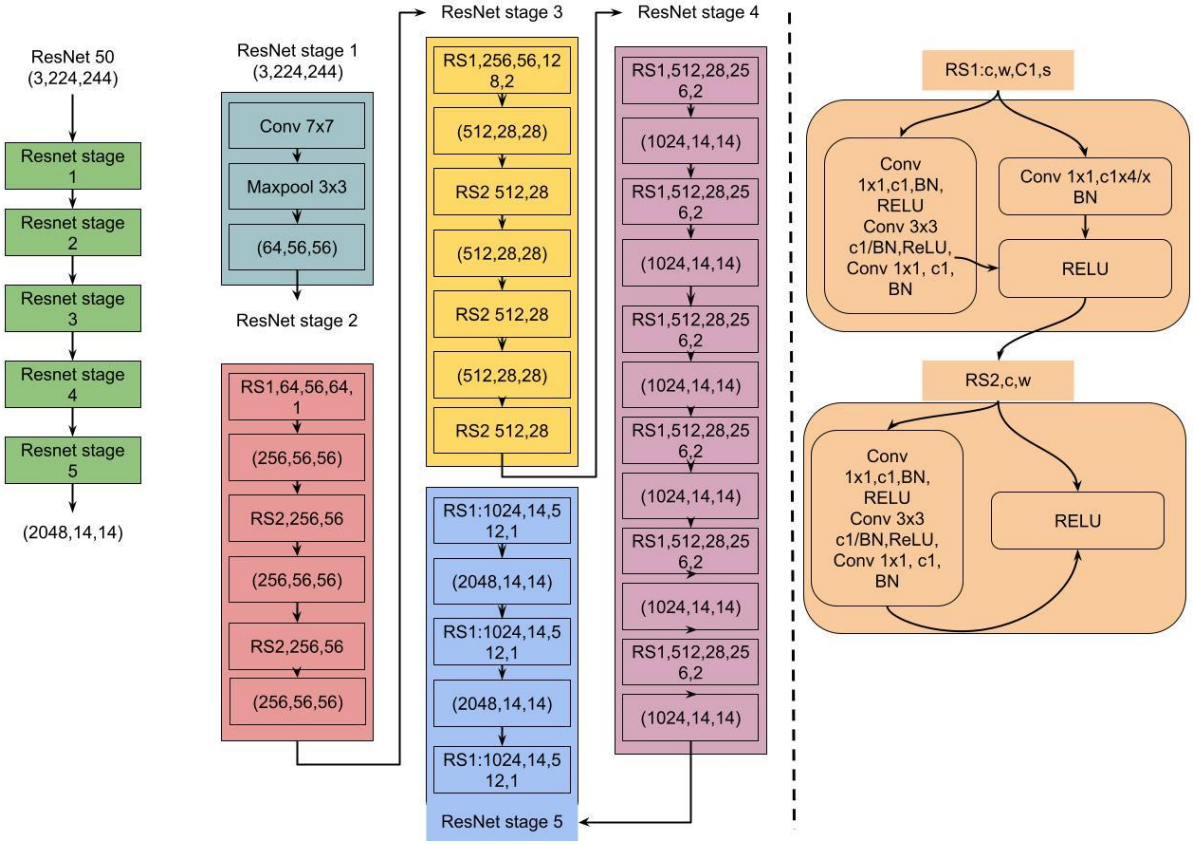


Figure 2. Convolutional ResNet attention model

Five separate stages make up the ResNet50 architecture used in this study for feature extraction: four residual blocks come after the initial convolutional processing block. Preliminary feature extraction and input dimension modification are accomplished by the convolutional processing block using four successive components: a 3 × 3 max pooling layer, a 7 × 7 convolutional layer, a batch normalisation (BN) layer, and a ReLU activation layer. The subsequent four residual blocks contain varying numbers of residual units: three, four, six, and three, respectively. These residual units are categorized into two types—the dimension-altering residual structure (RS1) and the dimension-preserving residual structure (RS2). This strategic implementation of these two residual structures enables ResNet50 to effectively address the vanishing gradient problem while achieving superior feature extraction performance. The detailed architectural configuration of the ResNet50 network is illustrated in Figure 2. To figure out the attention scores for each dimension feature, you need to multiply the Query and Key values. This helps the model understand how much attention to give to the different dimensions of the sample features. You can find the formula in Equation (9).

$$\text{score} = \text{Query} \times \text{Key} \quad (9)$$

Normalize the calculated attention score and convert the score into probability form via SoftMax, the formula is given by eqn (10)

$$\text{value} = \text{softmax}\left(\frac{\text{Query} \times \text{Key}^{T}}{\sqrt{d_{key}}}\right) \quad (10)$$

The final weighting matrix A is obtained by multiplying the value by Value. The number of trainings each iteration was 1000, while the model iteration was set at 200. 0.0001 was chosen as the learning rate. As the optimal network parameters, we used the conventional stochastic gradient descent technique.

6. Moath firefly sparrow colony optimization (MFSCO)

The parameters consist of dropout rate (γ), number of layers (L), kernel sizes (k), weight values (w), and learning rate (α). 1) The initialization procedure: Initially, the preprocessed output data is initialised as vectors for the coefficients a, b, and c. Fitness assessment: The fitness utility is assessed and the outcome is predicted by Equation (5).

$$Fit_i = \max accuracy$$

Let's break down the solution based on the filters: First, we have roaming as the primary fitness factor, followed by d_δ as the second finest fitness, and then a as the third finest fitness. During this stage, every male lion makes its way through the area, maintaining a specific location-to-number ratio of %S. While roaming, the lion adjusts its territory after finding the perfect spot. Equation (6) shows when a lion decides to move to a more favorable area. Along with these three candidates, we also consider α, β, δ, and ω. Additionally, the group encircles the prey's location to track them down. To calculate the encircling or trapping behavior of grey wolves while hunting, we use the following equation (11).

$$d(t+1) = d_{ps}(t) = \vec{B} \cdot \vec{K}$$

$$\vec{K} = \left|\vec{C}.d_{ps}(t+1) - d_{ps}(t)\right| \quad (11)$$

- Lion Optimization inspires the hunting technique. To feed their pride, the female lions focus on a specific group of prey. They follow certain strategies to surround their target and support their pride. Each lion has adapted to its environment, taking into account the surroundings of their fellow pride members. This teamwork allows them to encircle the prey and use their training to create local pressure and attacks. For this strategy, the hunters are split into three groups. The center of the seekers is where the prey is located. When we talk about swarms and particles, a swarm is made up of a collection of particles, which can be likened to groups of fish or birds.
- Let us break down the concept of position in the context of particles in a search space. Each particle, which we'll call i, has two key traits: its position and its velocity. The solution to our optimization problem is essentially the position of particle i, denoted as XiXi. The limits of the search space are defined by the symbols [$Xmin,max$]..
- • Velocity: A particle's velocity the particle can update its position since ViVi specifies its mobility in the search space. [$Vmin,Vmax$][Vmin,Vmax] represents the velocity's upper and lower bounds.
  • The objective function is sometimes referred to as the cost function or fitness function. An element from the decision space is mapped to the objective space via the goal function. The position Xi is used to assess the objective function, and the result is a real integer called the fitness value or cost value. The results of every objective function in the MOPSO scenario create a vector.
- • Local best: A particle's local best value is the position value that yields the highest fitness value throughout the course of its whole movement history. The symbol for it is $pBesti$pBesti.
  • Viable solution set: A viable solution is one that complies with every MOP constraint. The feasible solution set is a collection of all viable solutions.
  • Non-dominated solution: If there is not any viable solution that improves upon the existing one in some objective function without making another objective function worse, then the feasible solution is non-dominated.

  • External repository: All of the best particles (non-dominated solutions) are kept there [10]. This repository is represented by the letter A and is frequently referred to as an external archive. The maximum size of an external repository is Amax. Its size is constrained to prevent the excessive computational expense of updating and querying the external repository.
  • Leader: One solution (L) is chosen from the external repository to serve as the swarm's leader, and its rank is determined by the $pBetL$pBestL value.

For a specific species of firefly, the flash pattern is frequently distinct. The landscape of the fitness value to be optimized influences or determines a firefly's brightness.

When it comes to maximization, brightness is directly linked to the fitness value. The core concept behind intelligence optimization is that an agent, like a firefly, shines brighter based on its quality in a given situation, which can be seen as a solution to an optimization problem. Brighter fireflies attract their counterparts, regardless of gender, which boosts the efficiency of searching the area. The firefly algorithm is built on three idealized rules that reflect some of the basic flashing behaviors of real fireflies. Here are a few of those rules: • All fireflies, no matter their gender, are unisex and will be drawn to those that are more attractive and luminous. • A firefly's brightness directly correlates with its level of attraction. Additionally, because the air absorbs light, the brightness may diminish as the fireflies get farther away. A firefly will migrate at random if there is not another that is more appealing or brighter than it is. For vigilance detection, a specific percentage of sparrows are chosen at random from the entire population. The sparrow population instantly engages in anti-predator behavior upon identifying the threat. The following is the position update equation: Equation (12) indicates that the discoverers usually comprise 10–20% of the population.)

$$\text{xbest}_{d}^{t+1} + \frac{1}{D}\sum_{d=1}^{D} \left(\text{rand}\{-1,1\} \mid x_{i,d}^{t}\right. \qquad (12)$$

Typically, these equations use exponential and additive terms to adjust the sparrows' locations. In addition to the position update equation (13), the other sparrows, who are not the discoverer, act as followers.

$$x_{i,d}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{\text{xworst}_{d}^{t} - x_{i,d}^{t}}{i^2}\right), & i > \frac{n}{2} \\ -\text{xbest}_{d}^{t+1} \mid, & i \le \frac{n}{2}, \end{cases} \qquad (13)$$

In a d-dimensional space, the best posture for a sparrow during the population's t + 1 iteration is shown as xbestt+1 d, while its least favorable posture from the t iteration is noted as xwor stt d. If the i-th follower has not been fed and is feeling low on fitness, it means that i is greater than or equal to n/2. To boost its fitness, it needs to venture out and hunt in distant areas. On the other hand, if i is less than n/2, the ideal foraging spot will be randomly selected around the i-th follower. Typically, about 10–20% of sparrows are tasked with detection and early warning. Their positions can be updated using equation (14) as a reference.

$$x_{i,d}^{l+1} = \begin{cases} x\ \text{best}_{d}^{t+1} + \beta \cdot \mid x_{i,d}^{t} - x\ \text{best}_{d}^{t} \mid, & f_i \neq f_g \\ x_{i,d}^{t} + K \cdot \left(\frac{\mid x_{i,d}^{t} - x\ \text{worst}_{d}^{t} \mid}{\mid f_i - f_w \mid + e}\right), & f_i = f_g, \end{cases} \qquad (14)$$

where e is the smallest constant, fg and fw represent the current sparrow's best and worst adaptabilities, respectively, and β is the step length adjustment coefficient based on the normative standard distribution. This indicates that the sparrow is an outlier in the population and is consequently vulnerable to abuse when fi = fg. Given that fi = fg, the sparrow's quick grouping of other sparrows suggests that it recognises the threat posed by the predator. When there are no risks from predators, sparrows have more freedom to move about the search area. In this situation, sparrows may be more likely to be guided towards local or global optimal placements by the exponential part in the position update equation. The main goal of sparrows is to swiftly leave the area of danger when they sense the presence of predators. In this case, sparrows may be able to swiftly leave the danger zone thanks to the exponential term in the position update equation. In order to guarantee that the sparrows' escape route has enough unpredictability and lower the chance of being preyed upon, the additive term may simultaneously lessen its influence.

7. Experimental analysis

Simulation setup- For testing purposes, we employed a Core i7-7700 processor and a graphics-processing unit (GPU) in our experiment. Additionally, the proposed module has been trained using Keras and Python V3.9.

Dataset description- Twenty thousand image pairings with five modalities and accurate annotations make up UEMM-Air. Additionally, we run many tests on our dataset and produce new benchmark results. When compared to other comparable datasets, we discovered that models pre-trained on UEMM-Air perform better on downstream tasks. The dataset is openly accessible to facilitate the study of multi-modal UAV object identification models (https://github.com/1e12Leon/UEMM-Air).

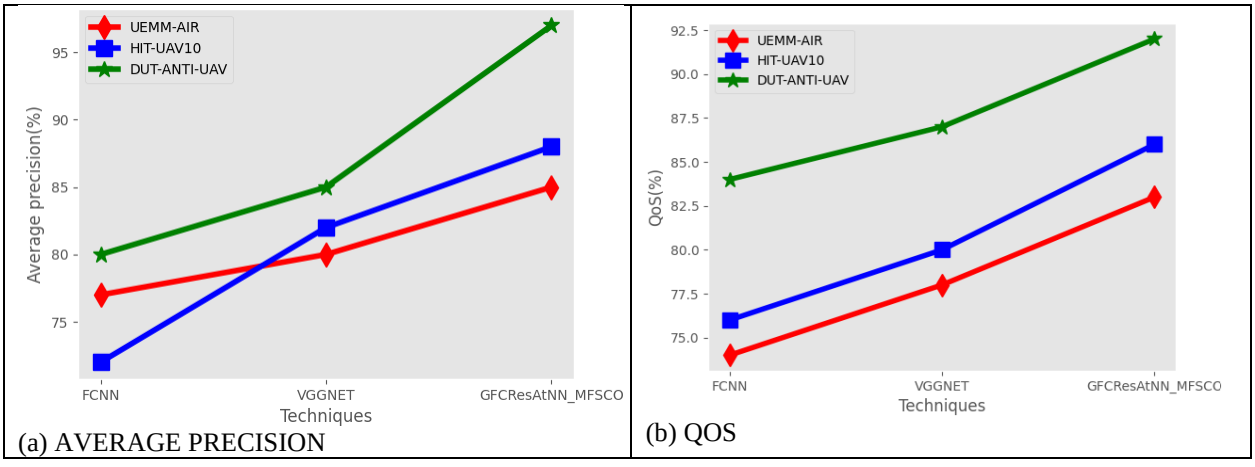
The 2,898 infrared thermal images in the collection were taken from 43,470 frames in hundreds of videos; all of the pictures were desensitized and gathered in public. The HIT-UAV10 offers both oriented and conventional annotated bounding boxes for each object in the photos to encourage efficient usage of the dataset on various applications. HIT-UAV10 has 24,899 annotated items across five object categories: Person, Car, Bicycle, OtherVehicle, DontCare. Items that the annotators were unable to properly classify are included in the DontCare category. There are 290 validation photos, 579 test images, and 2,029 training images in dataset.

There are around 35 different kinds of UAVs with different backgrounds in the DUT-Anti-UAV. The dataset's background encompasses a vast range of landscapes, including the sky, dark clouds, rainforest, high-rise, residential, rural, playground. Furthermore, consideration is given to diverse weather situations and lighting circumstances.

Table 1: Parametric analysis for various UAV dataset between proposed and existing techniques

Dataset	Techniques	Average precision	QoS	Scalability	Training accuracy	Recall
UEMM-AIR	**FCNN**	77	74	72	79	75
	VGGNET	80	78	79	83	81
	GFCResAtNN_MFSCO	85	83	87	89	84
HIT-UAV10	**FCNN**	72	76	77	81	78
	VGGNET	82	80	83	85	84
	GFCResAtNN_MFSCO	88	86	87	89	90
DUT-ANTI-UAV	**FCNN**	80	84	83	85	82
	VGGNET	85	87	90	93	88
	GFCResAtNN_MFSCO	97	92	96	98	95

The table-1 shows comparative analysis based on various UAV image dataset. Dataset analysed are UEMM-AIR, HIT-UAV10, and DUT-ANTI-UAV dataset. Parameters analysed are Average precision, QOS, SCALABILITY, training accuracy, RECALL.



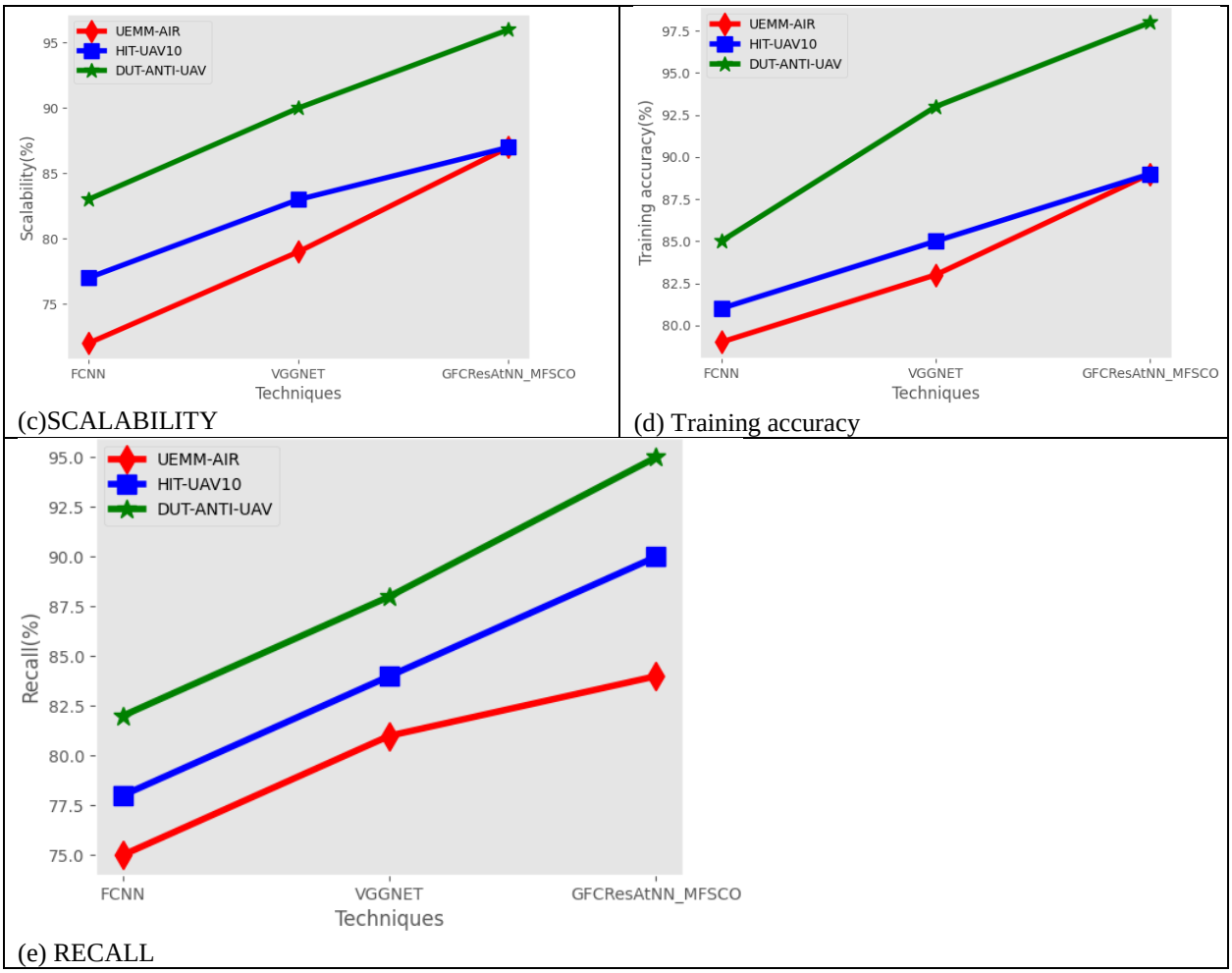


Figure 3. Parametric comparison for UAV image dataset in terms of (a) Average precision, (b) QOS, (c) SCALABILITY, (d) training accuracy, (e) RECALL

Figure-3 (a)- (e) shows comparative analysis for UAV image dataset. Proposed technique Average precision 85%, QOS 83%, SCALABILITY 87%, training accuracy of 89%, RECALL of 84%; FCNN attained Average precision 77%, QOS of 74%, SCALABILITY of 72%, training accuracy of 79%, RECALL of 75%, VGGNET attained Average precision of 80%, QOS of 78%, SCALABILITY of 79%, training accuracy 83%, RECALL 81%. for UEMM-Air dataset. Proposed technique attained Average precision 88%, QOS 86%, SCALABILITY of 87%, training accuracy of 89%, RECALL of 90%; FCNN attained Average precision of 72%, QOS of 76%, SCALABILITY of 77%, training accuracy of 81%, RECALL of 78%, VGGNET attained Average precision of 82%, QOS 80%, SCALABILITY of 83%, training accuracy of 85%, RECALL of 84% for HIT-UAV10 dataset. Proposed technique Average precision of 97%, QOS of 92%, SCALABILITY of 96%, training accuracy of 98%, RECALL of 95%; FCNN attained Average precision of 80%, QOS of 84%, SCALABILITY of 83%, training accuracy of 85%, RECALL of 82%, VGGNET attained Average precision of 85%, QOS of 87%, SCALABILITY of 90%, training accuracy of 93%, RECALL of 88% for DUT-ANTI-UAV dataset. Compared to alternative approaches that take weeks or even months to build a suitable reaction, this method is rapid and accurate. Following training, 300 aerial photographs of buildings and roads—taken at various scales, altitudes, and lighting conditions—were used for testing. UAV-based picture capture is a viable method for capturing high-resolution spatial photos of catastrophe sites during events such as floods. Machine learning models perform better when these photos contain extensive information about ground objects. The related image processing methods help initiate an appropriate emergency reaction and improve the precision and accuracy of landmark identification. Compared to man-driven aerial vehicles, unmanned aerial vehicles (UAVs) are more versatile and safer to operate since they are not constrained by their takeoff and landing conditions. Additionally, they fly at a low altitude, which allows them to explore normally inaccessible regions and get beyond satellite imaging limitations like cloud cover or other obstructions that obscure target view.

8. Conclusion

Using UAV images based on cloud IoT and deep learning models, this study proposes a unique method for detecting climate change in urban environments. Here, the UAV photos were gathered using an IoT cloud module and packaged for the dataset. Noise reduction as well as normalization is applied to this UAV picture dataset. The features of the processed image are then extracted using a graph cut fuzzy convolutional ResNet attention neural network with moath firefly sparrow colony optimization model is used for optimization. To evaluate the strength of the proposed methodology, it was also applied to artificially misregistered images. This problem is very relevant in diminishing the performance of change detection, however, it is few time analyzed via literature for supervised change detection methods. Finally, the new strategy of change detection analysis was tested by identifying the land cover changes in a natural disaster prone region using sub meter resolution VHR satellite images for binary land cover change detection. The usefulness of the UAV based imaging for carrying out flood inundation mapping in current study proves that UAVs are very much able to facilitate the monitoring in real time of already submerged areas.

References

[1] S. N. Mohanty *et al.*, "UAV databased temperature patterns analysis with carbon emission detection using deep neural network," *Remote Sens. Earth Syst. Sci.*, pp. 1–12, 2024.

[2] D. Hernández, J. C. Cano, F. Silla, C. T. Calafate, and J. M. Cecilia, "AI-enabled autonomous drones for fast climate change crisis assessment," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7286–7297, 2021.

[3] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Deep learning techniques to classify agricultural crops through UAV imagery: A review," *Neural Comput. Appl.*, vol. 34, no. 12, pp. 9511–9536, 2022.

[4] F. A. Almalki and M. C. Angelides, "Autonomous flying IoT: A synergy of machine learning, digital elevation, and 3D structure change detection," *Comput. Commun*, vol. 190, pp. 154–165, 2022.

[5] T. Mollick, M. G. Azam, and S. Karim, "Geospatial-based machine learning techniques for land use and land cover mapping using a high-resolution unmanned aerial vehicle image," *Remote Sens. Appl.: Soc. Environ.*, vol. 29, p. 100859, 2023.

[6] M. Sailaja, M. P. Kumar, B. S. Jyothi, G. L. N. Vanguri, S. Manjula, and D. D. Priya, "Remote Sensing–Based UAV Imaging in Heat Pattern Analysis Impact on Climate Change Detection Using Fuzzy Stacked Lasso Elastic Net Model," *Remote Sens. Earth Syst. Sci.*, vol. 7, pp. 699–708, Oct. 2024.

[7] M. Durairaj *et al.*, "Sustainable Agriculture-Based Climate Change Training Models using Remote Hyperspectral Image with Machine Learning Model," *Remote Sens. Earth Syst. Sci.*, pp. 1–10, 2024.

[8] F. Zennaro *et al.*, "Exploring machine learning potential for climate change risk assessment," *Earth-Sci. Rev.*, vol. 220, p. 103752, 2021.

[9] I. Stamatopoulos, T. C. Le, and F. Daver, "UAV-assisted seeding and monitoring of reforestation sites: a review," *Aust. For.*, vol. 87, no. 2, pp. 90–98, 2024.

[10] A. D. Mate *et al.*, "Marine Life Analysis Based on Ocean Water Level Rise and Climate Change Using Underwater Imaging Techniques," *Remote Sens. Earth Syst. Sci.*, pp. 1–13, 2024.

[11] V. Sharma, E. Honkavaara, M. Hayden, and S. Kant, "UAV remote sensing phenotyping of wheat collection for response to water stress and yield prediction using machine learning," *Plant Stress*, vol. 12, p. 100464, 2024.

[12] W. Zhu *et al.*, "UAV flight height impacts on wheat biomass estimation via machine and deep learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 2023.

[13] S. Arunmozhiselvi, T. A. Kumar, P. M. Bala, S. Usharani, and G. Glorindal, "A systematic approach to agricultural drones using a machine learning model," in *Machine Learning Approaches and Applications in Applied Intelligence for Healthcare Data Analytics*, CRC Press, 2022, pp. 41–60.

[14] D. Rolnick *et al.*, "Tackling climate change with machine learning," *ACM Comput. Surv*, vol. 55, no. 2, pp. 1–96, 2022.

[15] X. Deng, Y. Zhang, and H. Qi, "Towards optimal HVAC control in non-stationary building environments combining active change detection and deep reinforcement learning," *Build. Environ*, vol. 211, p. 108680, 2022.

[16] C. B. Pande, "Land use/land cover and change detection mapping in Rahuri watershed area (MS), India using the google earth engine and machine learning approach," *Geocarto Int.*, vol. 37, no. 26, pp. 13860–13880, 2022.