

Article

Swarm Intelligence with Deep Transfer Learning Driven Aerial Image Classification Model on UAV Networks

Saud S. Alotaibi ¹, Hanan Abdullah Mengash ², Noha Negm ^{3,4}, Radwa Marzouk ², Anwer Mustafa Hilal ^{5,*}, Mohamed A. Shamseldin ⁶, Abdelwahed Motwakel ⁵, Ishfaq Yaseen ⁵, Mohammed Rizwanullah ⁵ and Abu Sarwar Zamani ⁵

- ¹ Department of Information Systems, College of Computing and Information System, Umm Al-Qura University, Mecca 24382, Saudi Arabia; ssotaibi@uqu.edu.sa
 - ² Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; hamengash@pnu.edu.sa (H.A.M.); rmmarzouk@pnu.edu.sa (R.M.)
 - ³ Department of Computer Science, College of Science & Art at Mahayil, King Khalid University, Abha 62529, Saudi Arabia; nabeldelhamid@kku.edu.sa
 - ⁴ Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Menoufia 32511, Egypt
 - ⁵ Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia; asmaeil@psau.edu.sa (A.M.); iyasen@psau.edu.sa (I.Y.); mrizwan@psau.edu.sa (M.R.); azmani@psau.edu.sa (A.S.Z.)
 - ⁶ Department of Mechanical Engineering, Faculty of Engineering and Technology, Future University in Egypt, New Cairo 11835, Egypt; mohamed.abelbar@fue.edu.eg
- * Correspondence: a.hilal@psau.edu.sa



Citation: S. Alotaibi, S.; Abdullah Mengash, H.; Negm, N.; Marzouk, R.; Hilal, A.M.; Shamseldin, M.A.; Motwakel, A.; Yaseen, I.; Rizwanullah, M.; Zamani, A.S. Swarm Intelligence with Deep Transfer Learning Driven Aerial Image Classification Model on UAV Networks. *Appl. Sci.* **2022**, *12*, 6488. <https://doi.org/10.3390/app12136488>

Academic Editors: Sylvain Bertrand and Hyo-sang Shin

Received: 25 April 2022

Accepted: 22 June 2022

Published: 26 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Nowadays, unmanned aerial vehicles (UAVs) have gradually attracted the attention of many academicians and researchers. The UAV has been found to be useful in variety of applications, such as disaster management, intelligent transportation system, wildlife monitoring, and surveillance. In UAV aerial images, learning effectual image representation was central to scene classifier method. The previous approach to the scene classification method depends on feature coding models with lower-level handcrafted features or unsupervised feature learning. The emergence of convolutional neural network (CNN) is developing image classification techniques more effectively. Due to the limited resource in UAVs, it can be difficult to fine-tune the hyperparameter and the trade-offs amongst computation complexity and classifier results. This article focuses on the design of swarm intelligence with deep transfer learning driven aerial image classification (SIDTLD-AIC) model on UAV networks. The presented SIDTLD-AIC model involves the proper identification and classification of images into distinct kinds. For accomplishing this, the presented SIDTLD-AIC model follows a feature extraction module using RetinaNet model in which the hyperparameter optimization process is performed by the use of salp swarm algorithm (SSA). In addition, a cascaded long short term memory (CLSTM) model is executed for classifying the aerial images. At last, seeker optimization algorithm (SOA) is applied as a hyperparameter optimizer of the CLSTM model and thereby results in enhanced classification accuracy. To assure the better performance of the SIDTLD-AIC model, a wide range of simulations are implemented and the outcomes are investigated in many aspects. The comparative study reported the better performance of the SIDTLD-AIC model over recent approaches.

Keywords: computer vision; unmanned aerial vehicles; deep transfer learning; object detection; aerial image classification; parameter optimization

1. Introduction

Unmanned aerial vehicles (UAV) are utilized as a cost-efficient and prompt methodology for taking remote sensing (RS) images. The boon of UAV technology involves least cost, small size, security, natural function, and, especially, the fast and on-demand acquisition

of images [1]. The developments of UAV technologies have achieved the state that it can offer intense higher resolution RS images encircling lavish contextual and spatial information. This has allowed studies suggesting numerous original applications for UAV image examination, comprising disaster management, vegetation monitoring, object detection, detection and mapping of archaeological sites, oil and gas pipeline monitoring, and urban site analysis [2,3].

Aerial image classification methodologies grant distinct semantic categories that are usually established through exploiting changes in spatial deployments and structural forms for designing scenes [4]. In opposition with object or pixel related classifier methods, scene classification provides localization data from extensive aerial image which has apparent semantic data of the surfaces. Such methodologies are classified into three categories, which are: high level vision information, low level visual features, and mid-level visual representations [5,6]. Aerial scenes are differentiated by low level characteristics use, structural features, texture, spectral, and so on. Subsequently, low level feature vectors are pictorial ascriptions which can be derived globally or locally and are usually utilized for describing aerial scene images [7,8]. The typical low level feature methods are local binary patterns (LBP), Global Invariant Scale Transform (GIST), and color histogram Scale Invariant Feature Transform (SIFT). Mid-level analytical methods try to advance complete scene illustrations through conveying high order statistical outlines which are created by deriving local visual qualities [9]. The common processing pipeline derives local image patches, and they are programmed as local signals; therefore, creating a complete mid-level depiction of the aerial scenes. The familiar mid-level procedure is Bag of Visual Words (BoVW).

Deep learning (DL) procedures like Convolutional Neural Networks (CNNs) were broadly recognized as a notable approach for numerous computer vision applications (classification, image or video recognition, and detection), and have revealed amazing outcomes in various applications [10]. Therefore, there comes numerous advantages to stopping from utilizing DL methods in emergency response and calamity management applications to restore crucial data in a timely manner and permitting superior research and response in the course of time-critical circumstances, and supporting the decision-making processes [11]. Although CNNs were rising successfully at several classification roles via transfer learning (TL), their interpretation speed on implanted platforms, like those discovered on-board UAVs, is hampered by the high computational cost, which may acquire and the model size of these networks is prohibitive from a memory standpoint for these entrenched gadgets [12]. At the same time, most of the earlier works do not consider hyperparameter tuning process into account.

This article focuses on the design of swarm intelligence with deep transfer learning driven aerial image classification (SIDTLD-AIC) model on UAV networks. The presented SIDTLD-AIC model follows a feature extraction module using RetinaNet model, in which the hyperparameter optimization process is performed by the use of salp swarm algorithm (SSA). The SSA is chosen as it avoids the local optimal constraints, thus achieving a smooth balance between exploration and exploitation. In addition, a cascaded long short term memory (CLSTM) model is executed for classifying the aerial images. At last, seeker optimization algorithm (SOA) is applied as a hyperparameter optimizer of the CLSTM model and thereby results in enhanced classification accuracy. To assure the better performance of the SIDTLD-AIC model, a wide range of simulations are executed and the outcomes are investigated in various aspects.

2. Related Works

Haq et al. [13] applied DL based supervised image classification model and images gathered using UAV for the forest region classification. The DL technique based stacked Autoencoder (SAE) has shown remarkable potential with respect to the assessment of forest areas and image classification. The experiment result shows that DL technique provides improved performance than other machine learning approaches. The researchers in [14] address the shortcoming of multi-labeling UAV images, usually considered by a higher level

of dataset content, by presenting a novel technology based on CNN. They are employed as a means to produce an accurate representation of the query images that are analyzed afterward sub-dividing them into a grid of tiles. The multi-label classification process is implemented by combining a radial basis function neural network and a multi-labeling layer comprised of threshold operation. The researchers in [15] proposed a DL algorithm for classifying UAV images derived from the location and sensor of earth’s surface. Initially, the labelled and unlabelled UAV images are fed to a pre-trained CNN to generate deep feature representation. Next, we learned strong domain-invariant features with a further network comprised of two fully connected layers.

Rajagopal et al. [16] developed a new optimum DL-based scene classification algorithm captured by UAV. The suggested method includes a residual network-based features extraction (RNBFE) that extract feature from the convolutional layer of a DRN system. Furthermore, the various parameters result in configuration errors because of parameter tuning. Hence, self-adoptive global best harmony search (SGHS) approach is applied to tune the parameter of the presented model. The researchers in [17] present a multi-objective optimization algorithm to evolve deep CNN for scene classification that generates the non-dominant solution in an automatic manner at the Pareto front. Then, we used two sets of benchmark data sets for testing the effectiveness of the scene classification algorithm and making an extensive analysis. Pustokhina et al. [18,19] presented an energy-effective cluster-based UAV system using DL based scene classification model. The suggested method includes a clustering with parameter tuned residual network (C-PTRN) system that operates on two primary processes scene classification and cluster construction.

3. The Proposed Model

In this article, an automated SIDTLD-AIC method was established for the proper identification and classification of images into distinct kinds on UAV networks. The presented SIDTLD-AIC model follows a feature extraction module using RetinaNet model in which the hyperparameter optimization process is performed by the use of SSA. Next, the SOA-CLSTM model is applied to classify the aerial images. Figure 1 depicts the block diagram of SIDTLD-AIC technique.

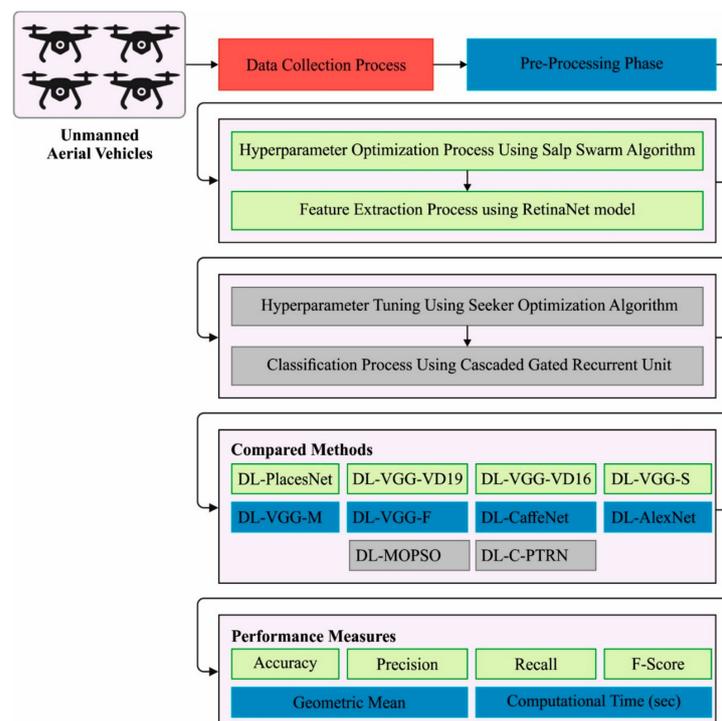


Figure 1. Block diagram of SIDTLD-AIC technique.

3.1. Feature Extraction Using RetinaNet Model

Transfer learning model is applied to enhance the efficiency of the DL model by the use of labeled data. It learns and employs many source processes for enhancing the learning process in relevant domains. It encompasses pre-training approaches which is trained on large scale dataset and is retrained at varying levels of the model on a small training set. The preliminary layer of the pre-training network can be modified upon requirement. The final layer of the model’s hyperparameters can be tuned for learning the abilities on new datasets. In this work, the RetinaNet based TL model is applied for deriving feature vectors. An input map was inspired by the individual layer still accomplishing the resulting map. The CNN is designed in an order of layers. Consider $\in R^{h' \times w' \times c'}$ ((h) : height, w : width, c : channel) are RGB images. Each layer gets X and the set of variables W as input as well as output images $y \in R^{h' \times w' \times c'}$, for example, $y = f(X, W)$. This makes an activation map to demonstrate the reaction of that filter at each spatial region. For calculating the input X with set of filters $W \in R^{h \times \bar{w} \times \bar{c} \times c'}$ and add a bias $b \in R^{c'}$ as follows.

$$y_{i'j'k'} = f \left(b_{k'} + \sum_{i=1}^{\bar{h}} \sum_{j=1}^{\bar{w}} \sum_{d=1}^c W_{ijdk} \times X_{i'+i,j'+j,d'} \right). \tag{1}$$

Next, the max-pooling layer is employed to decrease the computation and parameter with the decreased size of imputing shapes. It evaluates the maximum response of every image channel from $\bar{h} \times W$ sub-windows that implement as sub-sampling function. It can be expressed in the following:

$$y_{i'j'k'} = \max_{\substack{1 < j < \bar{w} \\ 1 < i < \bar{h}}} X_{i'+i,j'+j,k'}. \tag{2}$$

Finally, fully connected (FC) layer is a set of layers that integrate the data extracted by previous layer (feature). This layer gets an input X , processes them, and the last FC layer generates one dimensional vector of size. RetinaNet mainly consists of [20] two fully convolution network (FCNs), ResNet, and feature pyramid network (FPN). The ResNet employs network layer. The widely employed types of network layers are 50_, 101_, and 152_layers. The 101_layer with optimal trained efficacy is chosen. It could eliminate the structure of echocardiography with ResNet and, after, keep them to following sub-network. An FPN is an approach for efficiently eliminating the feature of each dimension from image with a conventional CNN architecture.

Focal loss: it can be improved version of the binary cross entropy (CE), the loss expression is given below:

$$CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1 - p) & \text{otherwise,} \end{cases} \tag{3}$$

In Equation (3), $y \in [-1, +1]$ indicates the ground truth type and $p \in [0, 1]$ represents the prediction probability to type $y = 1$.

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \tag{4}$$

The preceding formula can be abbreviated as follows:

$$CE(p, y) = CE(p_t) = -\log(p_t) \tag{5}$$

In order to resolve the problem of data imbalance among the negative and positive samples, the new process is changed into the succeeding process:

$$CE(p_t) = -\alpha_t \log(p_t). \tag{6}$$

Among them,

$$\alpha_t = \begin{cases} \alpha, & \text{if } y = 1, \\ 1 - \alpha & \text{otherwise,} \end{cases} \tag{7}$$

Here, $\alpha \in [0, 1]$ represents the weight factor. In order to resolve the problem, the concentrating variable C was determined to obtain the final process of focal loss:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t). \tag{8}$$

3.2. Hyperparameter Optimization: SSA

In this work, the hyperparameters of the RetinaNet model such as number of epochs, batch size, learning rate, and momentum are adjusted by the design of SSA. The SSA is simulated from the aggregation performance of salps that procedure a chain of salps and then hunt and move. The salp chain was developed from two kinds of salps, leader and follower [21]. The leader is the salp at the head of chains. Individual salps at the back of chains are followers. During the salp technique, food source F was determined as the individual with optimum fitness amongst every individual. The food source of t th order is $F(t)$. The steps of SSA technique are provided under.

1. Initialization of the population. In order for every individual, the places were arbitrary numbers amongst the upper as well as lower limits. They then compute fitness of every individual and sorted them. An individual with minimal fitness is the food source $F(t)$. $t = 1$, since one iteration was ended.
2. The population place was upgraded. The leader place was upgraded as:

$$x_j^i(t + 1) = \begin{cases} F_j(t) + c_1 [(ub_j - lb_j)c_2 + lb_j] & c_3 \geq 0.5 \\ F_j(t) - c_1 [(ub_j - lb_j)c_2 + lb_j] & c_3 < 0.5 \end{cases} \tag{9}$$

where $i = 1$, i.e., the count of leaders is 1. It ranks primary from the populations. $j = 1, 2 \dots D$. F_j , ub_j , and lb_j are $F(t)$, ub , and lb from the j th dimensional correspondingly. c_2 and c_3 implies the arbitrary numbers from zero and one. c_2 affects the step length of leader movement. c_3 defines if the leader moves forward/backward to food sources. T signifies the maximal number of iterations. c_1 refers the co-efficient of moving length.

$$c_1 = 2e^{-(4t/T)^2} \tag{10}$$

The place of follower is:

$$x_j^i(t + 1) = 0.5(x_j^{i-1}(t) + x_j^i(t)) \tag{11}$$

where $i \geq 2$, and is the sequence of followers from the population. $j = 1, 2 \dots D$.

3. Compute the fitness of every upgraded individual. The sort of individuals. Upgrade $F(t)$. Improve t by 1.
4. If the iteration accuracy condition was attained or $t = T$, the iteration terminates; or else, go to (2) to remain the iteration.

3.3. Image Classification Using Optimal CLSTM Model

In the final stage, the optimal CLSTM model is utilized to recognize different types of classes that exist in the aerial images [22]. A recurrent neural network (RNN) is a kind of DL method which is depending on existing input and the preceding input. In general, it is suitable for the scenario where the dataset has a consecutive correlation. But while handling a longer series of datasets, there exists an exploiting and vanishing gradient problem. In order to resolve this problem, a long short term memory (LSTM) is utilized that has an internal memory state which adds forget gate. The gate controls the time dependency and the effects of preceding inputs. Bidirectional long short term memory (BiLSTM) and bidirectional RNN (BiRNN) are other variations that reflect preceding input and assume

the upcoming input of a certain time frame. This work, can present the BiLSTM RNN and cascaded uni-directional LSTM models. The method comprises the initial layer of bi-directional RNN integrated with uni-directional RNN layer. The bi-directional LSTM comprises forward and backward tracks to learn patterns in two directions.

$$O_n^{f1}, h_n^{f1}, i_n^{f1} = L^{f1}(i_{n-1}^{f1}, h_{n-1}^{f1}, x_n : P^{f1}), \tag{12}$$

$$O_n^{b1}, h_n^{b1}, i_n^{b1} = L^{b1}(i_{n-1}^{b1}, h_{n-1}^{b1}, x_n : P^{b1}), \tag{13}$$

Equations (12) and (13) show the operation of forwarding and backward tracks. Figure 2 depicts the framework of LSTM.

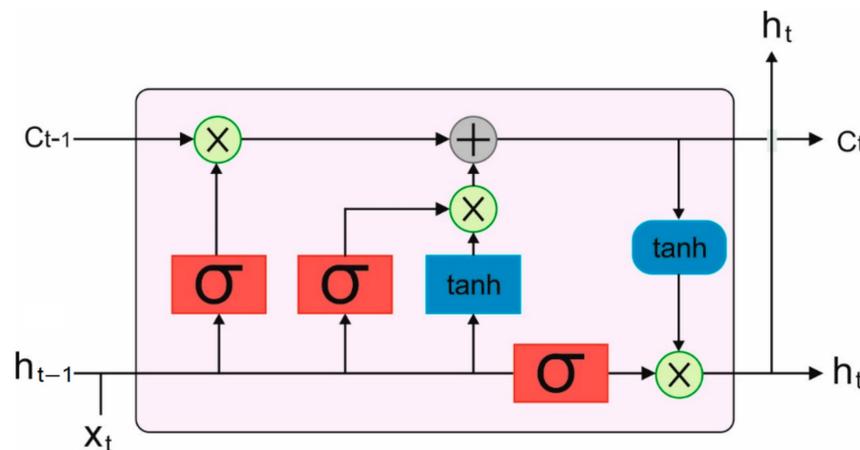


Figure 2. Infrastructure of LSTM.

From the equation, $O_n^{f1}, h_n^{f1}, i_n^{f1}$ and $O_n^{b1}, h_n^{b1}, i_n^{b1}$ indicate the output, the hidden state, and the internal state of the existing state for forwarding and backward LSTM tracks correspondingly. x_n denotes the sequential input, P indicates the LSTM cell variable. The output from these two tracks is integrated as in Equation (14) and forwarded into the next layers.

$$O_n^1 = O_n^{f1} + O_{N-n+1}^{b1}. \tag{14}$$

Bi-directional RNN and uni-directional RNN transform information into an abstract form and assist in learning spatial dependency. The output from the uni-directional layer can be attained by the following equation.

$$O_n^l, h_n^l, i_n^l = \text{LSTM}^l(i_{n-1}^l, h_{n-1}^l, O_n^{l-1}; P^l), \tag{15}$$

Now, the output from the lower layer O_n^{l-1} is integrated with preceding internal state i_{n-1}^l and hidden state h_{n-1}^l for obtaining output O_n^l of layer l , and P^l indicates a variable of the LSTM cell. The input dataset comprises a series of instances (x_1, x_2, \dots, x_N) , while every feature x_n is regarded at time n ($n = 1, 2, \dots, N$). The information is mainly classified into windows of time segment N and fed into the cascading LSTM. We attain predicted score vectors for every time step $(O_1^L, O_2^L, \dots, O_N^L)$ at the output. The entire prediction score can be attained by integrating the predictive score vector for the window N . The combination of scores can be implemented by using the sum rule as demonstrated in Equation (16) that implements well than other methodologies. Finally, the predictive score is transformed into probability using a softmax layer over Y .

$$Y = \frac{1}{N} \sum_{(n=1)}^N O_n^L. \tag{16}$$

We cascade LSTM to simulate incremental change of n time steps, and every LSTM is utilized for estimating the increment for one time step. In this work, θ -increment learning method learns increment of parameters using the cascaded LSTM network to gain higher frequency approximation, and θ represents the targeted parameter to be calculated.

In order to optimally elect the hyperparameter values of the CLSTM model, the SOA is exploited. In SOA, all the seekers have a central location vector \vec{c} , viz., the initial position for finding upcoming solutions, and it is regarded as estimated value Ex . Furthermore, all the seekers have a searching radius \vec{r} regarded as the En' , a trust level μ as membership degree, and a searching direction \vec{d} . Next, the seeker with some level of trust followed a potential direction and randomly moves towards the second point (novel candidate solution) in some searching radius from their existing location. In every time step t , the search decision-making is carried out for evaluating the four variables, and the seeker moves toward the novel location $\vec{x}(t + 1)$. The updating location from the central location can be defined as y -conditional cloud generator [23]:

$$\vec{x}_{ij}(t + 1) = \vec{c}_{ij}(t) + \vec{d}_{ij}(t) \times \vec{r}_{ij}(t) \times \sqrt{-\ln(\mu_i)} \tag{17}$$

Here “ i ” refers to the subscript index of seeker, and “ j ” indicates the subscript index of parameter dimension. The pseudo-code of the SOA is given in Algorithm 1.

Algorithm 1: Pseudocode of SOA

$t \leftarrow 0$
 Initialized generation of S position
 $\{x_i(t) | x_i(t) = (x_{i1}, x_{i2}, \dots, x_{iD}), i = 1, \dots, S, t = 0\}$ Uniformly and randomly in the parameters.
 Estimate all the seekers: Compute the fitness.
 Searching techniques provide search variables involving central location vector, searching direction, searching radius, and trust degree.
 Update new location of all the seekers is evaluated.
 $t \leftarrow t + 1$
 When $t < T_{max}$, then Go to 3; otherwise, End.

Instinctively, central location vector \vec{c} is fixed to existing location $\vec{x}(t)$. Similar to particle swarm optimization (PSO), all the seekers contain a memory stored in its optimal location \vec{p} and a global optimal location g accomplished by communicating with neighboring seekers. Every seeker is categorized into k class in the subscript index, and the seeker in a similar class belongs to virtual neighbors. Therefore, \vec{g} is established in the virtual neighbors.

$$\vec{c} = \vec{x}(t) + r_1 \varnothing_1 (\vec{p}(t) - \vec{x}(t)) + r_2 \varnothing_2 (\vec{g}(t) - \vec{x}(t)) \tag{18}$$

Now r_1, r_2 indicates the cognitive and social learning rates, correspondingly. \varnothing_1 and \varnothing_2 denotes the real number randomly and uniformly selected within the range of $[0, 1]$. In every experiment carried out in the study, $r_1 = 1, r_2 = 1$, and $k = 3$.

Generally, all the seekers have four significant directions, named local spacial direction \vec{d}_{ls} , local temporal direction \vec{d}_{lt} , global spacial direction \vec{d}_{gs} , and global temporal direction \vec{d}_{gt} , correspondingly.

$$\vec{d}_{lt} = \begin{cases} \text{sign} \left(\vec{x}(t) - \vec{x}(t - 1) \right) & \text{if } \text{fit} \left(\vec{x}(t) \right) \geq \text{fit} \left(\vec{x}(t - 1) \right) \\ \text{sign} \left(\vec{x}(t - 1) - \vec{x}(t) \right) & \text{if } \text{fit} \left(\vec{x}(t) \right) < \text{fit} \left(\vec{x}(t - 1) \right) \end{cases} \tag{19}$$

$$\vec{d}_{ls} = \text{sign} \left(\vec{x}(t) - \vec{x}(t) \right) \tag{20}$$

$$\vec{d}_{gt} = \text{sign}(\vec{p}(t) - \vec{x}(t)) \tag{21}$$

$$\vec{d}_{gs} = \text{sign}(\vec{g}(t) - \vec{x}(t)) \tag{22}$$

From the above equation, $\text{sign}(\cdot)$ indicates signum function, $\vec{x}'(t)$ represent the location of the seekers with the maximum fitness in a neighbor region, $\text{fit}(\vec{x}(t))$ denotes the fitness function (FF) of $\vec{x}(t)$. Next, searching direction is allocated based on the four directions.

$$\begin{aligned} \vec{d} = & \text{sign}(\omega(\text{sign}(\text{fit}(\vec{x}(t)) - \text{fit}(\vec{x}(t-1))))(\vec{x}(t) - \vec{x}(t-1)) \\ & + r_1\phi_1(\vec{p}(t) - \vec{x}(t)) + r_2\phi_2(\vec{g}(t) - \vec{x}(t)) \end{aligned} \tag{23}$$

In Equation (7), ω indicates the inertia weight that is fixed to $\omega = (T_{max}-t)/T_{max}$. Now, ϕ_1 and ϕ_2 indicates real numbers randomly and uniformly selected within $[0, 1]$.

Search Radius is essential, but challenging, to reasonably provide searching radius. For unimodal optimization problems, the performance is comparatively oblivious to searching radius to some extent. However, for multi-modal problems, various searching radii might lead to various performances of model particularly while handling variety of problems.

The μ variable is considered a quality assessment of location. It is equivalent to the fitness of $\vec{x}(t)$ or the index of ascensive sorting order of the fitness of $\vec{x}(t)$. Especially, the global optimal location has the maximal $\mu_{max} = 1.0$, when another location has a $\mu < 1.0$.

$$\mu = \mu_{max} - \frac{S - Sn}{S - 1}(\mu_{max} - \mu_{min}) \tag{24}$$

Here, Sn indicates the sequential value of $\vec{x}(t)$ afterward arranging the fitnesses of neighboring seekers in ascending sequence, μ_{max} and μ_{min} indicates the maximal and the minimal μ . We adapted $\mu_{max} = 1.0$, and $\mu_{min} = 0.2$.

The SOA method develops a fitness function (FF) to accomplish better classification accuracy. It describes a positive integer to characterize the improved performance of the candidate solution. In this work, the reduction of the classification error rate is regarded as the FF, as shown in Equation (25).

$$\text{fitness}(x_i) = \text{Classifier Error Rate}(x_i) = \frac{\text{number of misclassified samples}}{\text{Total number of samples}} * 100 \tag{25}$$

4. Experimental Validation

The performance validation of the proposed model is carried out using the UCM dataset [24]. The dataset contains a total of 2100 images and 21 classes (agricultural, airplane, baseballdiamond, beach, buildings, chaparral, denseresidential, forest, freeway, golfcourse, harbor, intersection, mediumresidential, mobilehomepark, overpass, parkinglot, river, runway, sparseresidential, storagetanks, and tenniscourt). It includes a total of 100 images under each class. The images were manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around the country. The pixel resolution of this public domain imagery is 1 foot. Each image measures 256×256 pixels. For experimental validation, the dataset is split into 70% of training set and 30% of testing set, i.e., 70 images from each class for training and remaining 30 images for testing purposes. Figure 3 showcases the sample images of UCM dataset.

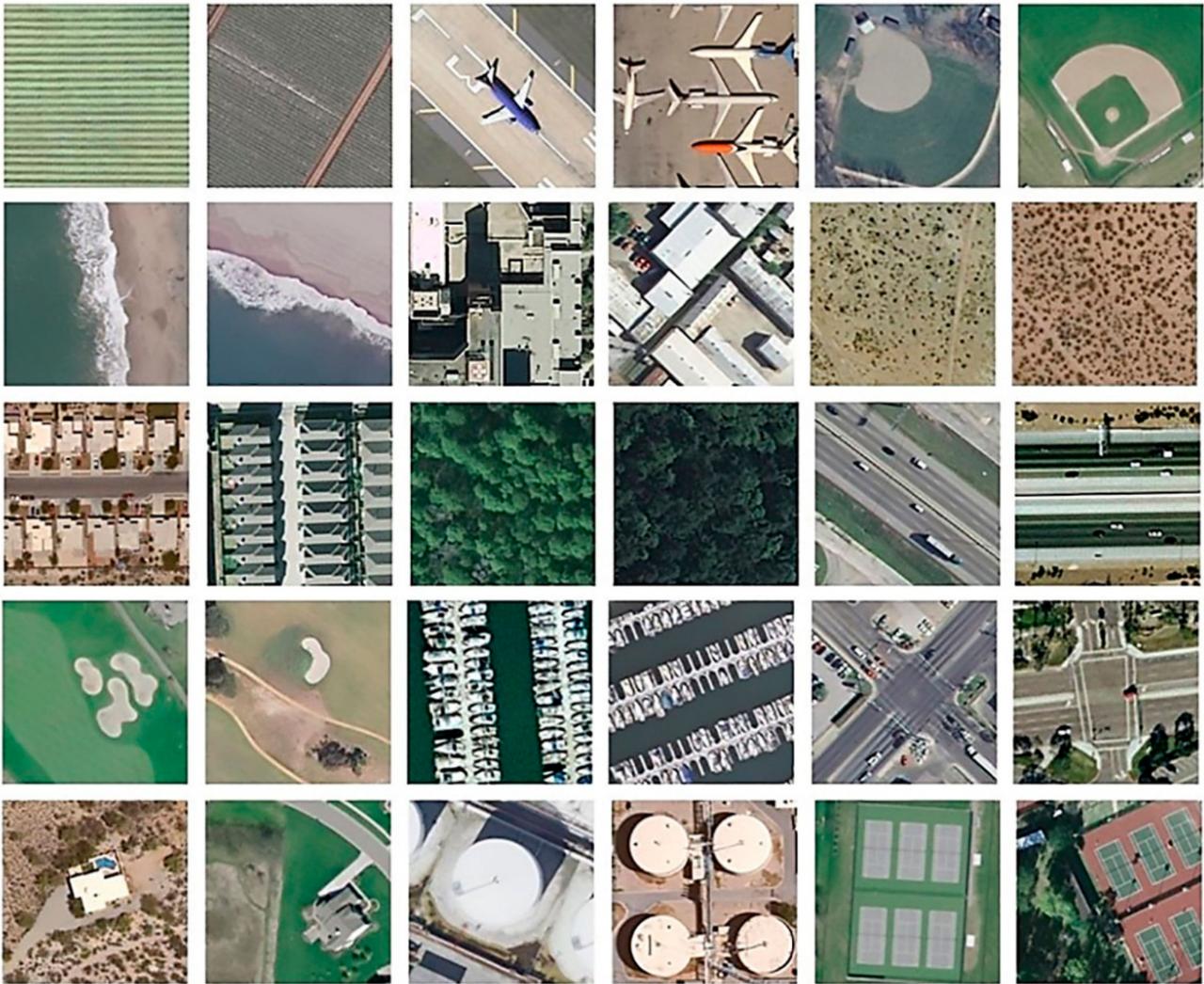


Figure 3. Samples-UCM Dataset.

Figure 4 illustrates the confusion matrices provided by the SIDTLD-AIC model on 70% of UCM datasets as training datasets. The results indicated that the SIDTLD-AIC model has effectually categorized all the 21 classes.

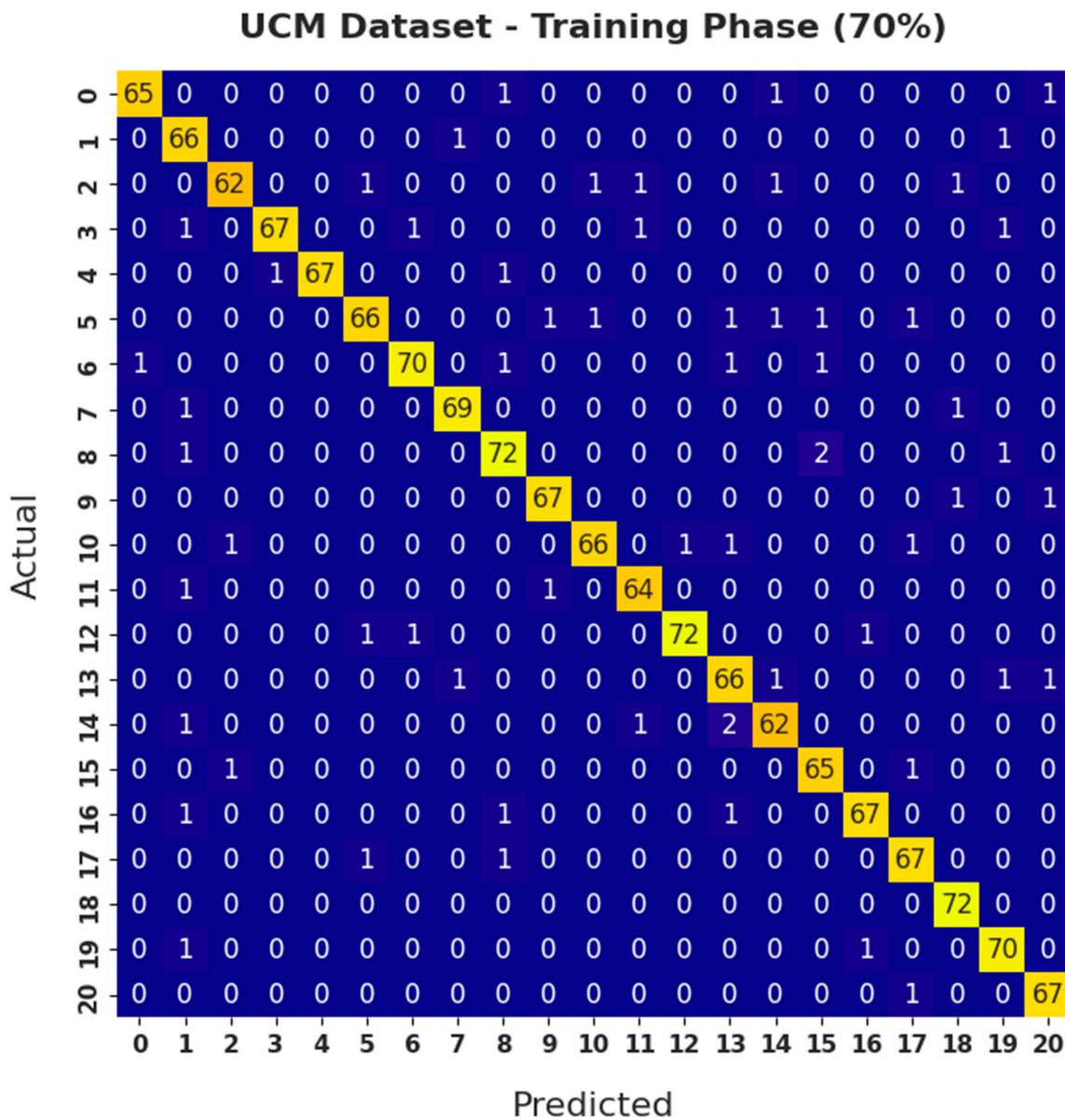


Figure 4. Confusion matrix of SIDTLD-AIC technique on 70% of UCM datasets as training datasets.

Table 1 reports the overall classification outcomes of the SIDTLD-AIC model 70% of UCM datasets as training datasets. The results inferred that the SIDTLD-AIC model has accomplished enhanced classifier outcomes on all class labels. For instance, the SIDTLD-AIC model has recognized class 1 samples with $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.39%, 90.41%, 97.06%, 93.62%, and 98.27%, respectively. Along with that, the SIDTLD-AIC method has recognized class 3 samples with $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.66%, 98.53%, 94.37%, 96.40%, and 97.11%, correspondingly. Moreover, the SIDTLD-AIC system has recognized class 13 samples with $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.32%, 91.67%, 94.29%, 92.96%, and 96.89%, correspondingly. Furthermore, the SIDTLD-AIC approach has recognized class 16 samples with $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.66%, 97.10%, 95.71%, 96.40%, and 97.76%, respectively. Lastly, the SIDTLD-AIC method has recognized class 20 samples with $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.73%, 95.71%, 98.53%, 97.10%, and 99.16%, correspondingly.

Table 1. Result analysis of SIDTLD-AIC technique with various measures on 70% of UCM datasets as training datasets.

Training Phase (70%)					
Class Labels	Accuracy	Precision	Recall	F-Score	Geometric Mean
0	99.73	98.48	95.59	97.01	97.73
1	99.39	90.41	97.06	93.62	98.27
2	99.52	96.88	92.54	94.66	96.13
3	99.66	98.53	94.37	96.40	97.11
4	99.86	100.00	97.10	98.53	98.54
5	99.39	95.65	91.67	93.62	95.64
6	99.59	97.22	94.59	95.89	97.19
7	99.73	97.18	97.18	97.18	98.51
8	99.39	93.51	94.74	94.12	97.16
9	99.73	97.10	97.10	97.10	98.47
10	99.59	97.06	94.29	95.65	97.03
11	99.66	95.52	96.97	96.24	98.37
12	99.73	98.63	96.00	97.30	97.94
13	99.32	91.67	94.29	92.96	96.89
14	99.46	93.94	93.94	93.94	96.78
15	99.59	94.20	97.01	95.59	98.36
16	99.66	97.10	95.71	96.40	97.76
17	99.59	94.37	97.10	95.71	98.40
18	99.80	96.00	100.00	97.96	99.89
19	99.59	94.59	97.22	95.89	98.46
20	99.73	95.71	98.53	97.10	99.16
Average	99.60	95.89	95.86	95.85	97.80

Figure 5 showcases the confusion matrices provided by the SIDTLD-AIC approach on 30% of UCM datasets as testing datasets. The results point out that the SIDTLD-AIC methodology has effectually categorized all the 21 classes.

Table 2 demonstrates the overall classification outcomes of the SIDTLD-AIC method on 30% of UCM datasets as testing datasets. The results exposed that the SIDTLD-AIC model has accomplished higher classifier outcomes on all class labels. For instance, the SIDTLD-AIC method has recognized class 1 samples with $accu_y$, $prec_n$, $reca_l$, F_{score} , and G_{mean} of 99.68%, 100%, 93.75%, 96.77%, and 96.82%, respectively. Next, the SIDTLD-AIC model has recognized class 3 samples with $accu_y$, $prec_n$, $reca_l$, F_{score} , and G_{mean} of 99.52%, 96.43%, 93.10%, 94.74%, and 96.41%, correspondingly. Furthermore, the SIDTLD-AIC system has recognized class 13 samples with $accu_y$, $prec_n$, $reca_l$, F_{score} , and G_{mean} of 99.21%, 93.10%, 90%, 91.53%, and 94.71%, respectively. Moreover, the SIDTLD-AIC methodology has recognized class 16 samples with $accu_y$, $prec_n$, $reca_l$, F_{score} , and G_{mean} of 99.52%, 100%, 90%, 94.74%, and 94.87%, respectively. Finally, the SIDTLD-AIC model has recognized class 20 samples with $accu_y$, $prec_n$, $reca_l$, F_{score} , and G_{mean} of 99.21%, 88.57%, 96.88%, 92.54%, and 98.10%, correspondingly.

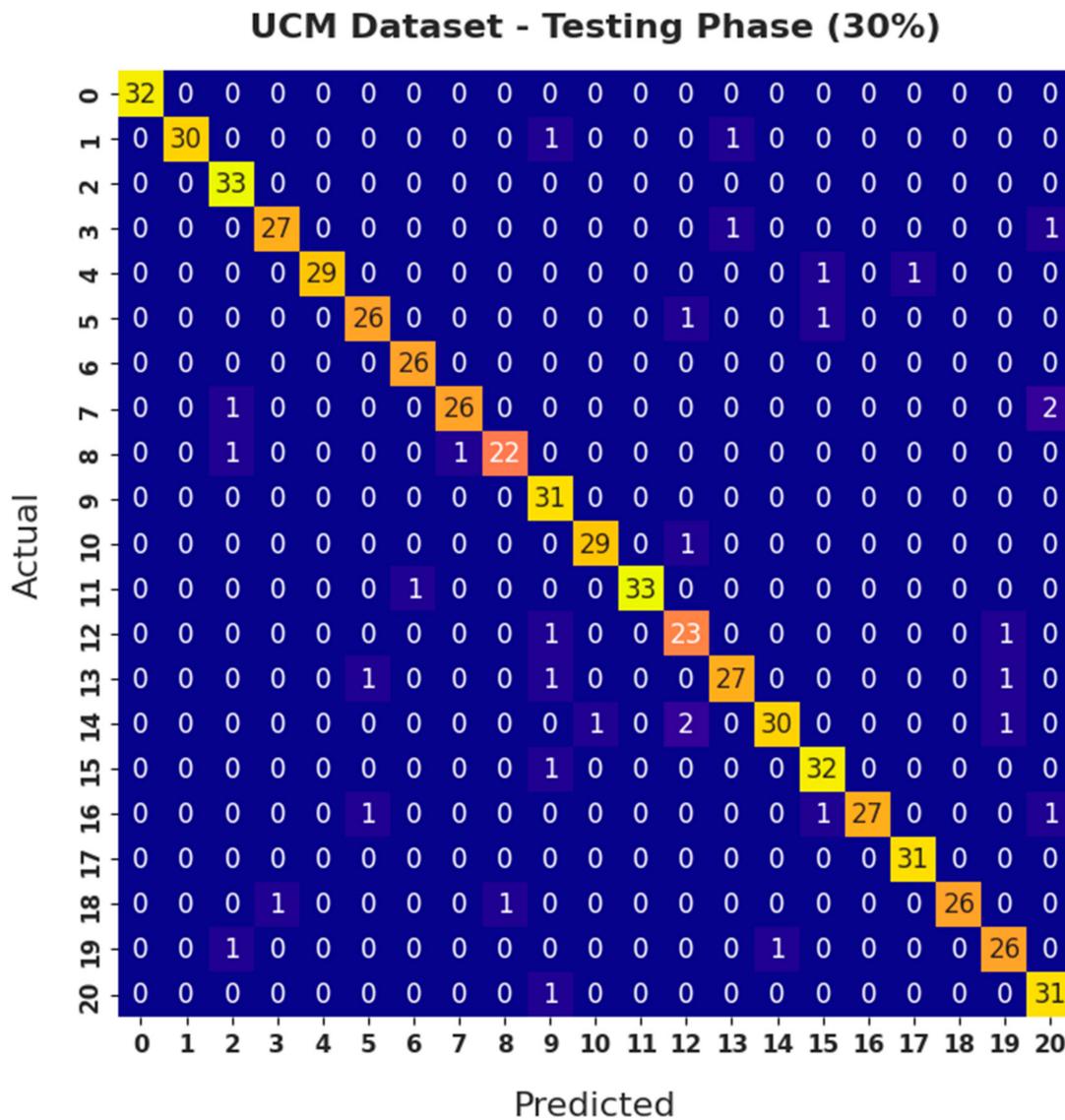


Figure 5. Confusion matrix of SIDTLD-AIC technique on 30% of UCM datasets as testing datasets.

Table 2. Result analysis of SIDTLD-AIC technique with various measures on 30% of UCM datasets as testing datasets.

Class Labels	Testing (30%)				
	Accuracy	Precision	Recall	F-Score	Geometric Mean
0	100.00	100.00	100.00	100.00	100.00
1	99.68	100.00	93.75	96.77	96.82
2	99.52	91.67	100.00	95.65	99.75
3	99.52	96.43	93.10	94.74	96.41
4	99.68	100.00	93.55	96.67	96.72
5	99.37	92.86	92.86	92.86	96.20
6	99.84	96.30	100.00	98.11	99.92
7	99.37	96.30	89.66	92.86	94.61

Table 2. *Cont.*

Class Labels	Testing (30%)				
	Accuracy	Precision	Recall	F-Score	Geometric Mean
8	99.52	95.65	91.67	93.62	95.66
9	99.21	86.11	100.00	92.54	99.58
10	99.68	96.67	96.67	96.67	98.24
11	99.84	100.00	97.06	98.51	98.52
12	99.05	85.19	92.00	88.46	95.60
13	99.21	93.10	90.00	91.53	94.71
14	99.21	96.77	88.24	92.31	93.85
15	99.37	91.43	96.97	94.12	98.23
16	99.52	100.00	90.00	94.74	94.87
17	99.84	96.88	100.00	98.41	99.92
18	99.68	100.00	92.86	96.30	96.36
19	99.21	89.66	92.86	91.23	96.12
20	99.21	88.57	96.88	92.54	98.10
Average	99.50	94.93	94.67	94.70	97.15

The training accuracy (TA) and validation accuracy (VA) attained by the SIDTLD-AIC model on UCM dataset is demonstrated in Figure 6. The experimental outcome implied that the SIDTLD-AIC model has gained maximum values of TA and VA. In specific, the VA seemed higher than TA.

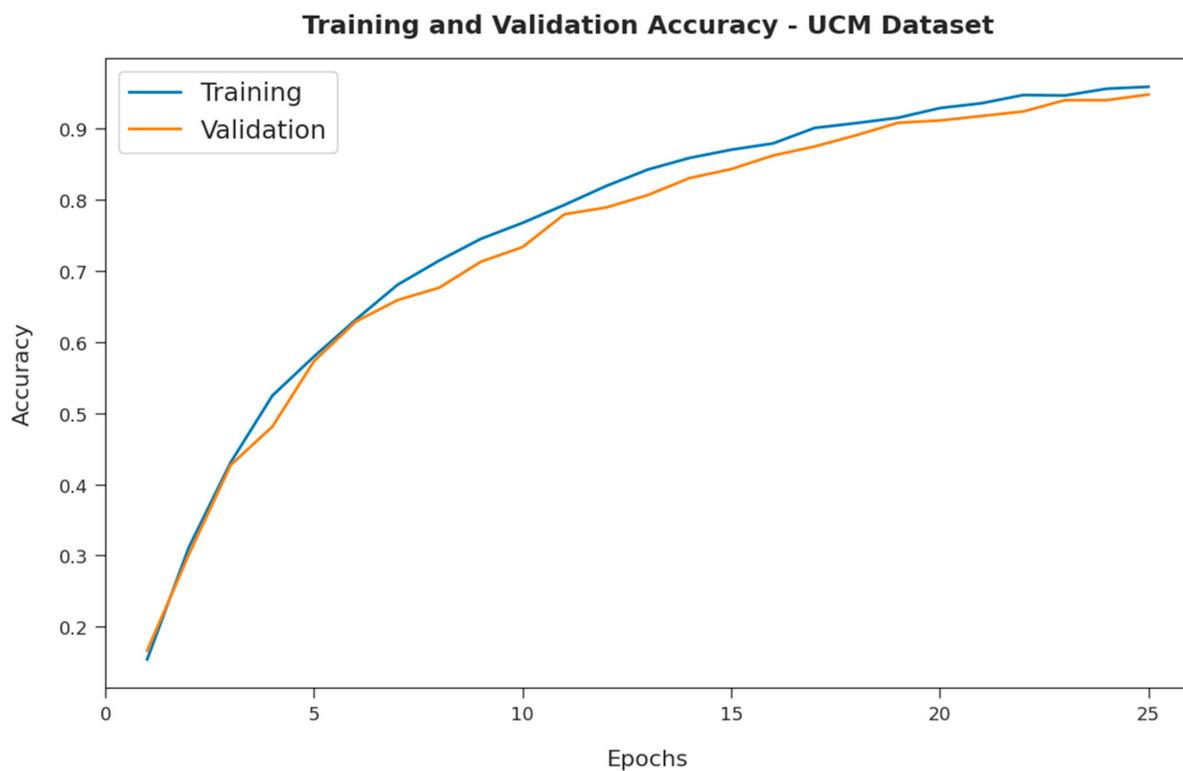


Figure 6. TA and VA analysis of SIDTLD-AIC technique on UCM dataset.

The training loss (TL) and validation loss (VL) achieved by the SIDTLD-AIC model on UCM dataset are established in Figure 7. The experimental outcome inferred that the SIDTLD-AIC model has been able least values of TL and VL. In specific, the VL seemed that lower than TL.

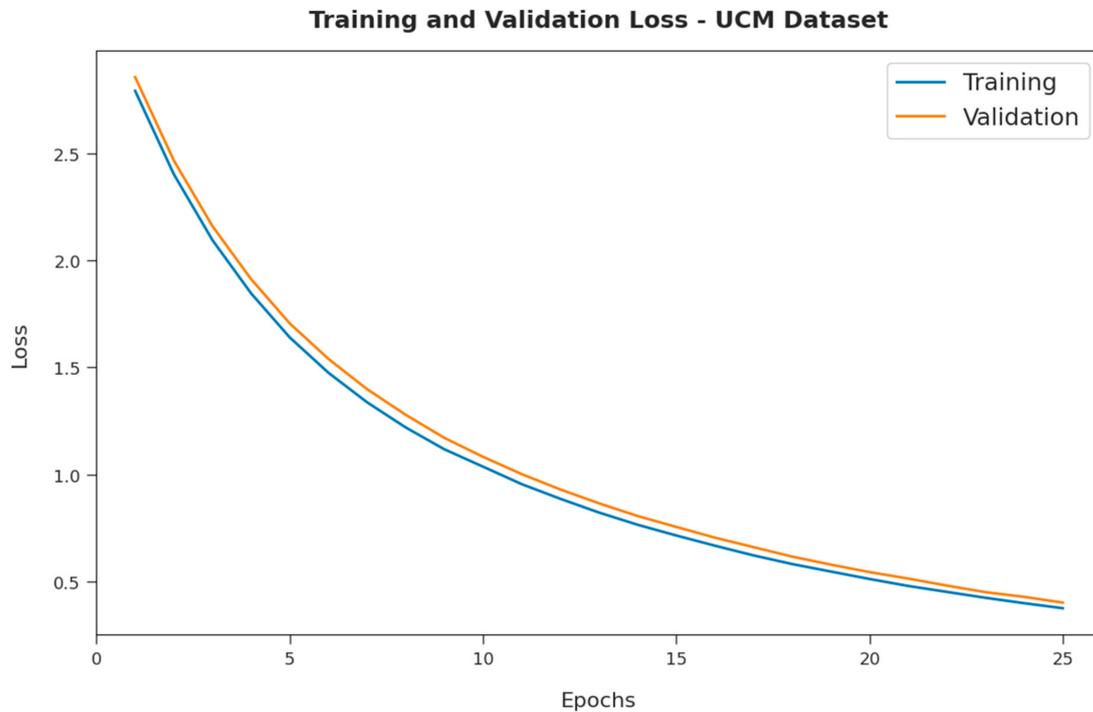


Figure 7. TL and VL analysis of SIDTLD-AIC technique on UCM dataset.

A brief precision-recall examination of the SIDTLD-AIC method on UCM dataset is portrayed in Figure 8. By observing the figure, it can be noticed that the SIDTLD-AIC method has been able maximal precision-recall performance under all classes.

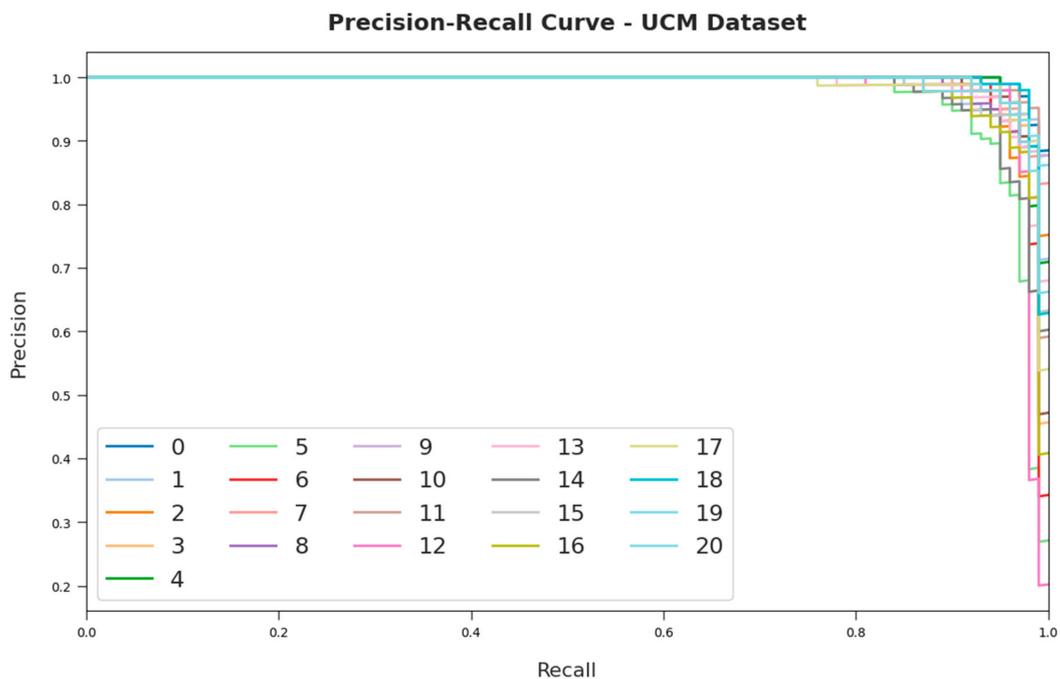


Figure 8. Precision-recall curve analysis of SIDTLD-AIC technique on UCM dataset.

A detailed ROC investigation of the SIDTLD-AIC approach to UCM dataset is represented in Figure 9. The results indicated that the SIDTLD-AIC model has exhibited its ability in categorizing different classes on the UCM dataset.

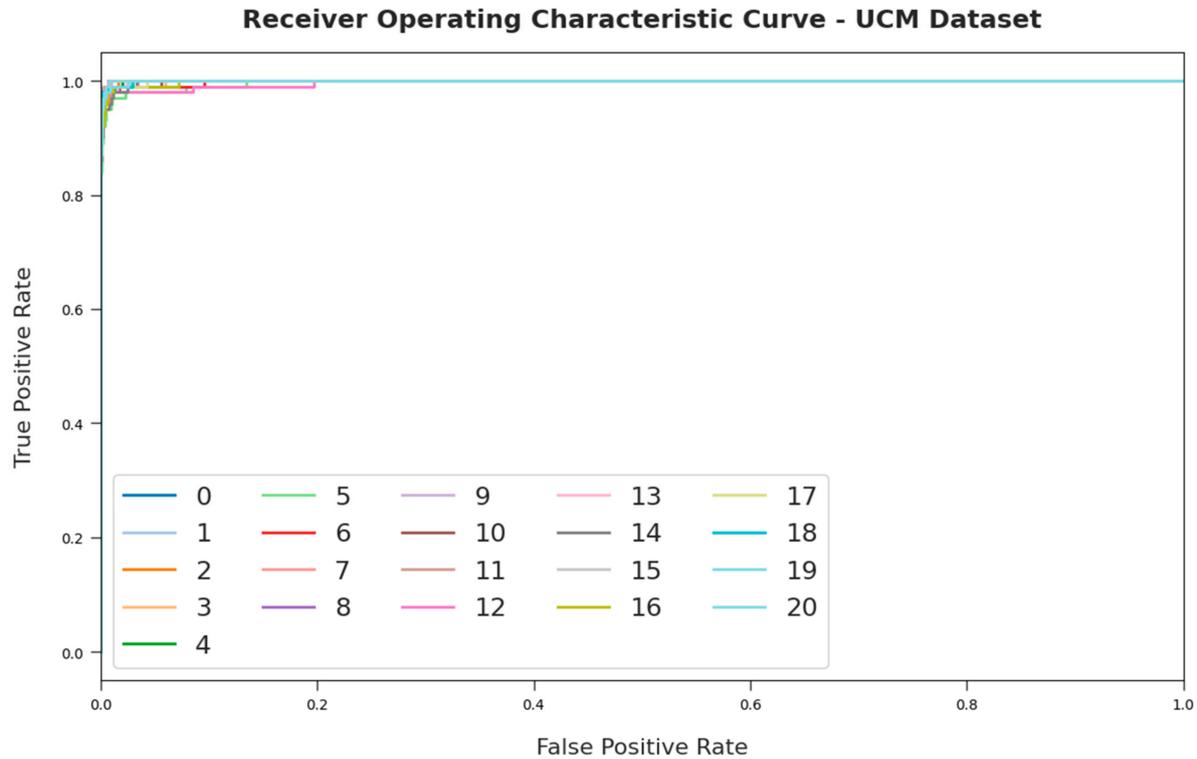


Figure 9. ROC curve analysis of SIDTLD-AIC technique on UCM dataset.

Figure 10 depicts the average image classification results of the SIDTLD-AIC model on 70% of UCM datasets as training datasets and 30% of UCM datasets as testing datasets. The figure shows that the SIDTLD-AIC model has resulted in better classification results under both aspects. On applied 70% of UCM datasets as training datasets, the SIDTLD-AIC model has resulted in average $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 96.60%, 95.89%, 95.86%, 95.85%, and 97.80%, respectively. Likewise, on applied 30% of UCM datasets as testing datasets, the SIDTLD-AIC model has resulted in average $accu_y$, $prec_n$, $reca_1$, F_{score} , and G_{mean} of 99.50%, 94.93%, 94.67%, 94.70%, and 97.15%, respectively.

Figure 11 illustrates a comparative $accu_y$ examination of the SIDTLD-AIC model with recent models. The experimental values implied that the DL-PlacesNet and DL-VGG-VD19 models have shown lower values of $accu_y$. Moreover, the DL-VGG-VD16, DL-VGG-M, DL-VGG-F, DL-CaffeNet, and DL-AlexNet models have resulted to closer $accu_y$ values. Then, the DL-VGG-S and DL based multiobjective PSO (DL-MOPSO) techniques have reached reasonable $accu_y$ values of 95.24% and 95.81%. Though the DL-C-PTRN model has resulted in considerable $accu_y$ of 98.96%, the SIDTLD and SIDTLD+SSA models have accomplished near optimal $accu_y$ of 98.98% and 99.01%. However, the SIDTLD-AIC model has accomplished superior outcome with maximum $accu_y$ of 99.50%.

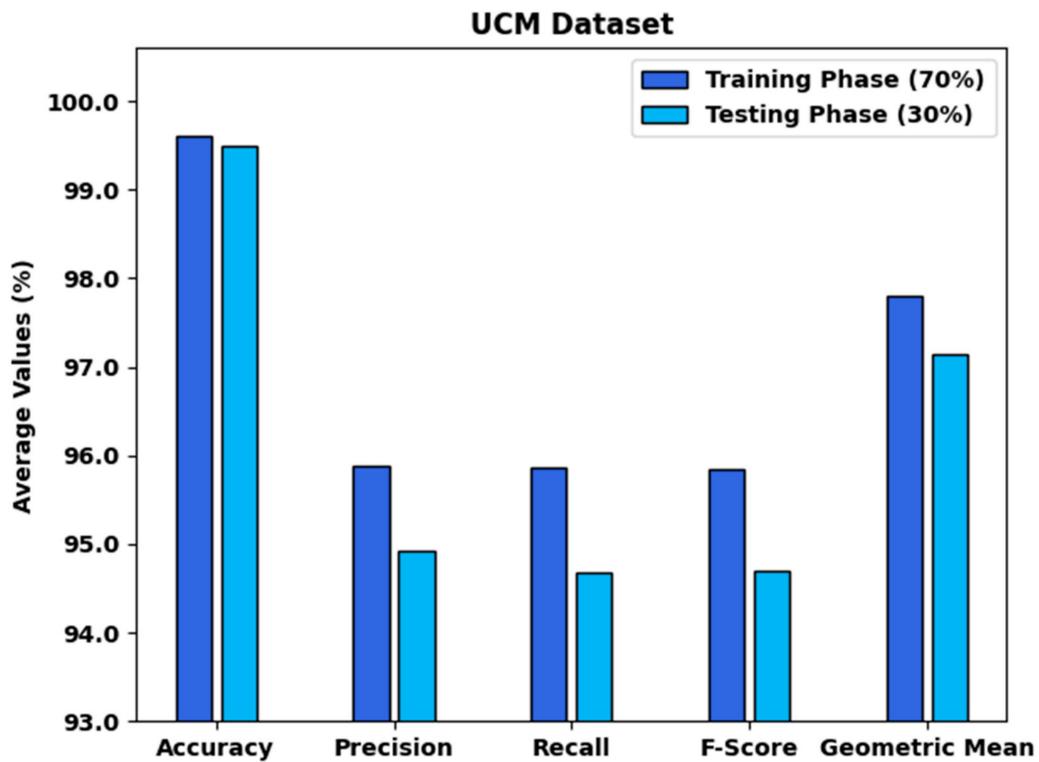


Figure 10. Average analysis of SIDTLD-AIC technique with various measures.

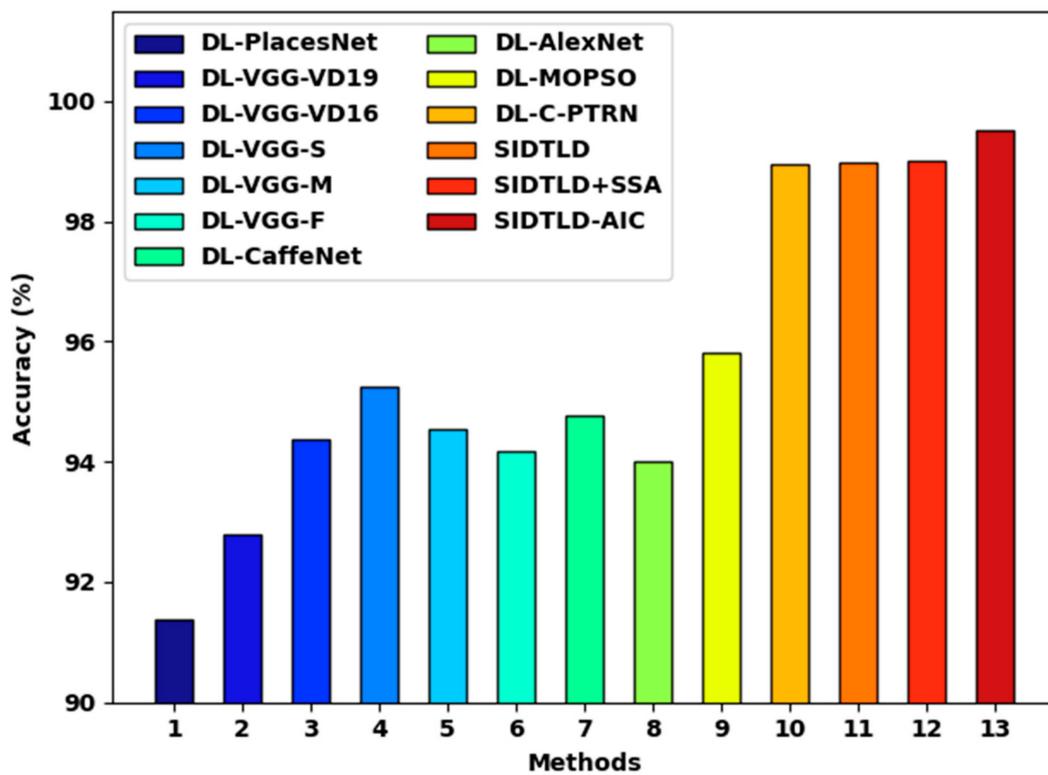


Figure 11. Accuracy analysis of SIDTLD-AIC technique with existing methods.

Finally, a computation time (CT) assessment of the SIDTLD-AIC model with recent models is carried out in Figure 12. The experimental values implied that the DL-PlacesNet, DL-VGG-VD-19, DL-VGG-VD-16, and DL-VGG-S approaches have obtained increased CT values. Followed by, the DL-VGG-F, DL-CaffeNet, and DL-AlexNet models have reached moderately reduced CT values. The DL-MOPSO and DL-C-PTRN models have accomplished reasonable CT of 135s and 95s, respectively. Meanwhile, the SIDTLD and SIDTLD+SSA models have attained CT of 67s and 54s, respectively. Finally, the SIDTLD-AIC model has outperformed other methods with minimal CT of 40s. The results implied that the SIDTLD-AIC model has gained enhanced classification performance due to the inclusion of SSA and SOA based hyperparameter optimizers. From the above results and discussion, it can be stated that the SIDTLD-AIC model has accomplished enhanced image classification results on the UAV networks.

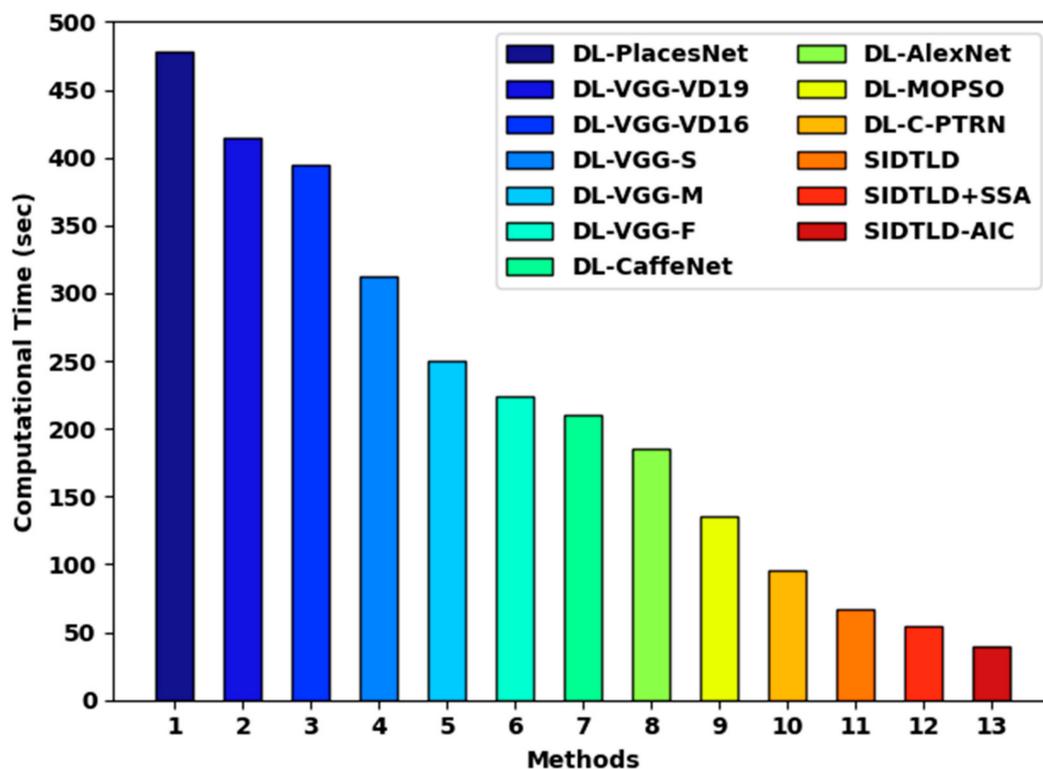


Figure 12. CT analysis of SIDTLD-AIC technique with existing approaches.

5. Conclusions

In this article, an automated SIDTLD-AIC technique was established for the proper identification and classification of images into distinct kinds on UAV networks. The presented SIDTLD-AIC model follows a feature extraction module using RetinaNet model in which the hyperparameter optimization process is performed by the use of SSA. Next, the SOA-CLSTM model is applied to classify the aerial images. For assuring the better performance of the SIDTLD-AIC method, a wide range of simulations are executed and the outcomes are investigated in various aspects. The comparative study reported the better performance of the SIDTLD-AIC model over recent approaches with maximum accuracy of 99.50%. Thus, the presented SIDTLD-AIC model can be exploited for aerial image classification in real time environment such as vegetation mapping, crop classification, disaster management, weather prediction, etc. In future, hybrid metaheuristics should be utilized for improving the overall classification performance. Furthermore, the proposed model can be extended to real-time large-scale databases in future. Moreover, the investigation of the performance using statistical analysis can be done in our future work.

Author Contributions: Conceptualization, S.S.A. and N.N.; methodology, H.A.M.; software, R.M.; validation, A.M.H., R.M. and M.R.; investigation, A.M.H., M.A.S.; resources, M.R.; data curation, R.M.; writing—original draft preparation, S.S.A., H.A.M., N.N., I.Y.; writing—review and editing, A.M., M.A.S., A.S.Z.; visualization, A.S.Z.; supervision, S.S.A.; project administration, R.M.; funding acquisition, H.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (42/43). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R114), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4210118DSR21).

Data Availability Statement: Data sharing not applicable to this article as no datasets were generated during the current study.

Conflicts of Interest: The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

References

- Choi, S.K.; Lee, S.K.; Kang, Y.B.; Seong, S.K.; Choi, D.Y.; Kim, G.H. Applicability of image classification using deep learning in small area: Case of agricultural lands using UAV image. *J. Korean Soc. Surv. Geod. Photogramm. Cartogr.* **2020**, *38*, 23–33.
- Tetila, E.C.; Machado, B.B.; Astolfi, G.; de Souza Belete, N.A.; Amorim, W.P.; Roel, A.R.; Pistori, H. Detection and classification of soybean pests using deep learning with UAV images. *Comput. Electron. Agric.* **2020**, *179*, 105836. [[CrossRef](#)]
- Öztürk, A.E.; Erçelebi, E. Real UAV-bird image classification using CNN with a synthetic dataset. *Appl. Sci.* **2021**, *11*, 3863. [[CrossRef](#)]
- Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep learning approach for car detection in UAV imagery. *Remote Sens.* **2017**, *9*, 312. [[CrossRef](#)]
- Bashmal, L.; Bazi, Y.; Al Rahhal, M.M.; Alhichri, H.; Al Ajlan, N. UAV image multi-labeling with data-efficient transformers. *Appl. Sci.* **2021**, *11*, 3974. [[CrossRef](#)]
- Anwer, M.H.; Hadeel, A.; Fahd, N.A.-W.; Mohamed, K.N.; Abdelwahed, M.; Anil, K.; Ishfaq, Y.; Abu Sarwar, Z. Fuzzy cognitive maps with bird swarm intelligence optimization-based remote sensing image classification. *Comput. Intell. Neurosci.* **2022**, *2022*, 4063354.
- Abunadi, I.; Althobaiti, M.M.; Al-Wesabi, F.N.; Hilal, A.M.; Medani, M.; Hamza, M.A.; Rizwanullah, M.; Zamani, A.S. Ederated learning with blockchain assisted image classification for clustered UAV networks. *Comput. Mater. Contin.* **2022**, *72*, 1195–1212.
- Li, J.; Yan, D.; Luan, K.; Li, Z.; Liang, H. Deep learning-based bird's nest detection on transmission lines using UAV imagery. *Appl. Sci.* **2020**, *10*, 6147. [[CrossRef](#)]
- Youme, O.; Bayet, T.; Dembele, J.M.; Cambier, C. Deep Learning and Remote Sensing: Detection of Dumping Waste Using UAV. *Procedia Comput. Sci.* **2021**, *185*, 361–369. [[CrossRef](#)]
- Mittal, P.; Singh, R.; Sharma, A. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image Vis. Comput.* **2020**, *104*, 104046. [[CrossRef](#)]
- Bouguettaya, A.; Zarzour, H.; Kechida, A.; Taberkit, A.M. Vehicle detection from UAV imagery with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–21, *in press*. [[CrossRef](#)] [[PubMed](#)]
- Huang, H.; Deng, J.; Lan, Y.; Yang, A.; Deng, X.; Zhang, L.; Wen, S.; Jiang, Y.; Suo, G.; Chen, P. A two-stage classification approach for the detection of spider mite-infested cotton using UAV multispectral imagery. *Remote Sens. Lett.* **2018**, *9*, 933–941. [[CrossRef](#)]
- Haq, M.A.; Rahaman, G.; Baral, P.; Ghosh, A. Deep learning based supervised image classification using UAV images for forest areas classification. *J. Indian Soc. Remote Sens.* **2021**, *49*, 601–606. [[CrossRef](#)]
- Zeggada, A.; Melgani, F.; Bazi, Y. A deep learning approach to UAV image multilabeling. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 694–698. [[CrossRef](#)]
- Bashmal, L.; Bazi, Y. Learning robust deep features for efficient classification of UAV imagery. In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–4.
- Rajagopal, A.; Ramachandran, A.; Shankar, K.; Khari, M.; Jha, S.; Lee, Y.; Joshi, G.P. Fine-tuned residual network-based features with latent variable support vector machine-based optimal scene classification model for unmanned aerial vehicles. *IEEE Access* **2020**, *8*, 118396–118404. [[CrossRef](#)]
- Rajagopal, A.; Joshi, G.P.; Ramachandran, A.; Subhalakshmi, R.T.; Khari, M.; Jha, S.; Shankar, K.; You, J. A deep learning model based on multi-objective particle swarm optimization for scene classification in unmanned aerial vehicles. *IEEE Access* **2020**, *8*, 135383–135393. [[CrossRef](#)]

18. Pustokhina, I.V.; Pustokhin, D.A.; Kumar Pareek, P.; Gupta, D.; Khanna, A.; Shankar, K. Energy-efficient cluster-based unmanned aerial vehicle networks with deep learning-based scene classification model. *Int. J. Commun. Syst.* **2021**, *34*, e4786. [[CrossRef](#)]
19. Outay, F.; Mengash, H.A.; Adnan, M. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transp. Res. Part A Policy Pract.* **2020**, *141*, 116–129. [[CrossRef](#)]
20. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery. *Remote Sens.* **2019**, *11*, 531. [[CrossRef](#)]
21. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
22. Yadav, R.K.; Bhattarai, B.; Jiao, L.; Goodwin, M.; Granmo, O.C. Indoor Space Classification Using Cascaded LSTM. In Proceedings of the 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 9–13 November 2020; pp. 1110–1114.
23. Shafik, M.B.; Chen, H.; Rashed, G.I.; El-Sehiemy, R.A. Adaptive multi objective parallel seeker optimization algorithm for incorporating TCSC devices into optimal power flow framework. *IEEE Access* **2019**, *7*, 36934–36947. [[CrossRef](#)]
24. Yang, Y.; Newsam, S. Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification. In Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), San Jose, CA, USA, 2–5 November 2010.