


Please cite the Published Version

Hamza, Ameer, Khan, Muhammad Attique, Rehman, Shams ur, Al-Khalidi, Mohammed , Alzahrani, Ahmed Ibrahim, Alalwan, Nasser and Masood, Anum (2024) A novel bottleneck residual and self-attention fusion-assisted architecture for land use recognition in remote sensing images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. pp. 1-16. ISSN 1939-1404

DOI: <https://doi.org/10.1109/jstars.2023.3348874>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Accepted Version

Downloaded from: <https://e-space.mmu.ac.uk/633604/>

Usage rights:  [Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Additional Information: This is an open access article which originally appeared in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

A Novel Bottleneck Residual and Self-Attention Fusion-Assisted Architecture for Land Use Recognition in Remote Sensing Images

Ameer Hamza, Muhammad Attique Khan, *Member, IEEE*, Shams ur Rehman, Mohammed Al-Khalidi, Ahmed Ibrahim Alzahrani, Nasser Alalwan, Anum Masood

Abstract— The massive yearly population growth is causing hazards to spread swiftly around the world and have a detrimental impact on both human life and the world economy. By ensuring early prediction accuracy, remote sensing enters the scene to safeguard the globe against weather-related threats and natural disasters. Convolutional neural networks, which are a reflection of deep learning, have been used more recently to reliably identify land use in remote sensing images. This work proposes a novel bottleneck residual and self-attention fusion-assisted architecture for land use recognition from remote sensing images. First, we proposed using the fast neural approach to generate cloud-effect satellite images. In neural style, we proposed a 5-layered residual block CNN to estimate the loss of neural-style images. After that, we proposed two novel architectures, named 3-layered bottleneck CNN architecture and 3-layered bottleneck self-attention CNN architecture, for the classification of land use images. Training has been conducted on both proposed and original neural-style generated datasets for both architectures. Subsequently, features are extracted from the deep layers and merged employing an innovative serial approach based on weighted entropy. By removing redundant and superfluous data, a novel Chimp Optimization technique is applied to the fused features in order to further refine them. In conclusion, selected features are classified using the help of neural network classifiers. The experimental procedure yielded respective accuracy rates of 99.0% and 99.4% when applied to both datasets. When evaluated in comparison to state-of-the-art (SOTA) methods, the outcomes generated by the proposed framework demonstrated enhanced precision and accuracy.

Index Terms— Remote Sensing; Land Use; Neural-Style; Deep Learning; Self-Attention Network; Fusion; Optimization

I. INTRODUCTION

Satellite imaging plays a crucial role in a wide range of applications, including environmental monitoring, law

enforcement, and disaster response [1]. In order to do these duties, it is crucial to use a human procedure for recognizing the facilities and objects shown in the photographs. This strategy is crucial for organizations or governments aiming to produce precise representations of the Earth [2]. Remote sensing (RS) refers to the process of collecting and analyzing data about an object, location, or event from a distance. The technologies used to capture these pictures are categorized under RS [3]. Satellite imaging has been used to generate paddy maps with the implementation of advanced remote sensing technology [4]. Satellite imaging sources are often freely accessible, offering a wide coverage area and regular updates with excellent time resolution over a large geographical expanse [5]. Due to the growing need for enhanced clarity, satellite pictures are progressively rising in size [6]. Furthermore, with the increasing abundance of remote sensing (RS) data, it is now possible to explore a diverse array of intricate scientific problems [7].

The scope of data collecting, processing methods, and applications within remote sensing is extensive [8]. Furthermore, technologies from several disciplines, such as pattern recognition and machine learning, are typically used to build approaches to interpret remotely sensed data [9]. Before classification, the classification techniques primarily use extracted features for training. These attributes can generally be categorized into three main groups: low, mid, and high levels [10]. While possessing unique advantages, most current classification techniques have solely relied on attributes derived from one of the three categories, resulting in limited precision [11]. On the other hand, we think that using several features at different levels, a process known as feature fusion will improve classification resilience and accuracy by reducing all drawbacks associated with using a single method [12]. The primary driving force behind feature fusion is providing the classifier with the

Corresponding Author: Muhammad Attique Khan (attique.khan@ieee.org), Anum Masood (anum.masood@ntnu.no)

Ameer Hamza and Shams ur Rehman are with Department of CS, HITEC University, Taxila, Pakistan.

Muhammad Attique Khan is with Department of CS and Mathematics, Lebanese American University, Beirut, Lebanon, and Department of CS, HITEC University, Taxila, Pakistan

Mohammed Al-Khalidi is with Department of Computing and Mathematics, Manchester Metropolitan University, UK.

Ahmed and Nasser are with Computer Science Department, Community College, King Saud University, P.O. Box 28095 Riyadh 11437, Saudi Arabia.

Anum is with Department of Physics, Norwegian University of Science and Technology, Trondheim, NO-7491, Norway. (Corresponding author e-mail: anum.masood@ntnu.no).

most complete and pertinent data possible, which could not be accomplished with a single technique [13].

Many scene classification methods have been presented to help comprehend and identify the scene information in aerial photographs; these methods can be broadly classified into two categories: low-level scene characteristics and mid-level scene features[14]. A few of the often employed low-level techniques are GIST, Local Binary Pattern (LBP) [15], Color Histogram (CH) [16], and Scale Invariant Feature Transform (SIFT) [17]. Using low-level feature descriptors as a code, the midlevel methods depict a scene. Improved Fisher Kernel (IFK), Locality-Constrained Linear Coding (LLC), Probabilistic Latent Semantic Analysis (PLSA) [18], Bags of Visual Words (BoVW) [19], Spatial Pyramid Matching (SPM), Latent Dirichlet Allocation (LDA) [20], and Vector of local aggregated Descriptors (VLAD) are some of the midlevel coding techniques [21]. In recent years, deep learning techniques have made significant progress in computer vision applications like face, object, and picture recognition. Convolutional neural networks (CNNs) are among the best-performing algorithms for deep learning [22]. CNN models, including CafeNet and GoogLeNet, have recently outperformed low-level and mid-level techniques in classifying aerial scenes [23]. Deep models based on convolution neural networks (CNNs) have been employed in this context for feature extraction.

The deep features of the images that are nonlinear, discriminant, and invariant are extracted by means of multiple convolutional and pooling layers. [24]. Various techniques are used, including transfer learning through fine-tuning and feature descriptors. This allows for learning high-level features in deeper layers and determining low-level characteristics in shallower layers [25]. However, to train CNNs effectively and achieve decent performance, a substantial amount of labeled training data is needed [26]. The design of an intelligent system heavily relies on computer vision. Features extraction, fusion, and selection are the three primary parts of computer vision. The deep features are extracted by the deep learning model [27].

Consequently, feature fusion approaches were proposed by computer vision researchers. The fusion process has increased the system's precision and predictor count. Serial-based fusion and parallel fusion are two common fusion methods [28]. Lastly, feature selection is crucial in determining the most discriminating features when considering feature fusion approaches for better classification [29]. Feature selection should proceed concurrently with feature fusion to maximize the classification rate; otherwise, the performance will suffer in terms of computational time, memory usage, and overall accuracy (OA) [30]. In this article, we proposed an automatic deep learning-based framework for classifying land use scenes from satellite images, as shown in Figure 1.

The primary contributions are as follows:

- We proposed a technique to generate cloud-effect satellite images using the fast neural approach. In neural style, we proposed a 5-layered residual block CNN to estimate the loss of neural-style images.

- We proposed two novel architectures, named 3-layered bottleneck CNN architecture and 3-layered bottleneck self-attention CNN architecture, for the classification of land use images.
- We proposed a newly weighted entropy serial feature fusion technique, and the fused information was further refined by employing a new Chimp optimization for the best feature selection.
- A detailed ablation study has been performed to validate the performance of the proposed method. Also, a brief comparison has been conducted with a few recent techniques.



Figure 1: Samples of each class from UC-Merced land use dataset

The remaining article has been organized as follows. Section 2 presents this paper's related work, including a summary of a few existing techniques. Section 3 describes the proposed methodology, including the proposed CNN model, weighted entropy serial fusion process, and chimp optimization for feature selection with mathematical justification. Section 4 discusses the results of the proposed methodology on UC-Merced and cloudy UC-Merced datasets, and Section 5 concludes the results of the proposed methods.

II. RELATED WORK

Several deep learning-based computer vision techniques have recently been presented for classifying satellite images [31]. While some researchers used satellite data to study agricultural and urban areas, others used airborne to concentrate on buildings [32]. For example, Nguyen et al.[33] presented a method for the classification of satellite images using deep learning. They utilized CNN-based deep Neural Network architecture. As a result, they achieved 93% accuracy. The limitation of this presented method is that mapping the agriculture framework is a challenging task for processing satellite images. Unnikrisnan et al. [34] developed a technique for classifying imagery from satellites with deep learning. This study employed the CNN-based architecture of Alex-Net and

VGG Neural Network to train and assess various outcomes. The drawback of this strategy is the decrease in the number of trainable parameters, which will be examined in future analysis. Chen et al. [35] presented an approach for classifying remote sensing using deep learning. They created CNN-based network architecture using Attention-Guided sparse filters. As a result, they achieved 94% accuracy. The limitation of this method was the lack of a large-scale dataset. Insufficient annotated images have been demonstrated to be a barrier to improving CNNs' accuracy in classifying land-use scenes. Datta et al. [36] presented a method presented a method for the classification of imbalanced hyperspectral images using ADASYN. The authors extracted the global discriminative features and mulit-grained scanning features for the classification and they achieved 94.19% average accuracy. The limitation of work was the authors did not include the ablation study and comparison with recent techniques. Khalid et al. [37] presented an automatic deep learning framework for the classification of plant deceases. The authors implemented the CNN and mobilenetV2 architecture with explainable AI for using GradCam algorithm and they achieved the 89% accuracy. Abdulkareem et al. presented a survey on deep mapping analysis framework for the analysis of dehazing images. The authors used three online databases namely y, IEEE Xplore, Web of Science (WOS), and ScienceDirect (SD). The data was collected from 2008 to 2021. They collected 152 articulated in the final set and they picked 55 out of 152 articles based on dehazing methods.

Zhang et al. [38] presented a method for classifying satellite images using deep learning. They utilized CNN-based architecture to combine the TC best track data with various infrared satellite image intensities. The limitation of this method is that the aforementioned physical attributes will be incorporated into our model to enhance its performance in estimating TC intensity, particularly to offer novel concepts for examining abrupt variations in TC intensity. Kadhim et al. [39] presented a deep-learning approach for the classification of satellite images. In this work, they utilized the CNN-based Convolutional Neural Network of Alex-Net and VGG19 for training purposes. As a result, they achieved 98% accuracy. Li et al. [40] presented a model based on SDVI as, compared to other models, they offered the best accuracy. In this paper, ship detection was done using SDSOI images. The enhanced YOLOV3 real-time network is employed to identify minute particles. Compared to SOAT, it is 9.6% more efficient. The primary drawbacks of this investigation are that (i) changes in input size have an impact on performance, and (ii) it shows a lackluster level of noise tolerance.

Wu et al. [41] presented a model for detecting airplanes in satellite images that relies on a novel aircraft detection

framework based on the objectiveness methodologies of CNN and BING. CNN is useful in object detection tasks and can automatically extract features from the provided raw data. BING provides a candidate object zone that increases the detection rate and saves time. A dataset from Google Earth's airplane detection was used in this paper. This work's limitations stem from the fact that differences in the type of aircraft, including size and position, reduce efficiency. Liu et al. [42] presented a technique for classifying imagery from satellites with deep learning. They employed a convolutional neural network (CNN) architecture for the purpose of classification. Consequently, they attained a level of accuracy of 99%.

An approach was developed by Olsen et al. [43] that targets Synthetic Aperture Radar images characterized by an exceptional degree of detail. Vessels were identified in high-resolution synthetic aperture radar images by utilizing information gathered from both terrestrial and satellite sources. The results are presented using high-resolution images obtained from RADARSAT-2. A primary limitation of this methodology is its propensity to induce delays in the process of decision-making, particularly in comparison to alternative models. Chaudhary et al. [44] presented a YOLOv2-based model. This model demonstrated that YOLOv3's average accuracy (AP) score, which was about 90.25, was significantly higher than YOLOv2's, which generated an AP score of approximately 90.05. The fact that this model's inference time is 22ms, far faster than YOLOv2's 25ms, is another important finding. The Sentinel-1 images and Gaofen-3 are included in the dataset. This model has two shortcomings: (i) it is unsuitable for video; (ii) it is negatively impacted by weather.

III. PROPOSED METHODOLOGY

In this section, the proposed framework for the classification of land use scenes using remote sensing images is presented. Figure 2 illustrates that, initially, the UC-Merced dataset is utilized to generate the cloudy effect in the images by using a novel fast-style transfer method. After that, both datasets are utilized to train the two novel customized deep-learning models. Deep features are extracted from both trained models. The extracted features are fused using a newly proposed weighted entropy serial fusion approach. Fused features are further utilized for the feature selection process. The best features are selected by employing a chimp optimization technique. In addition, the selected features are passed to the neural network classifier for the final classification.

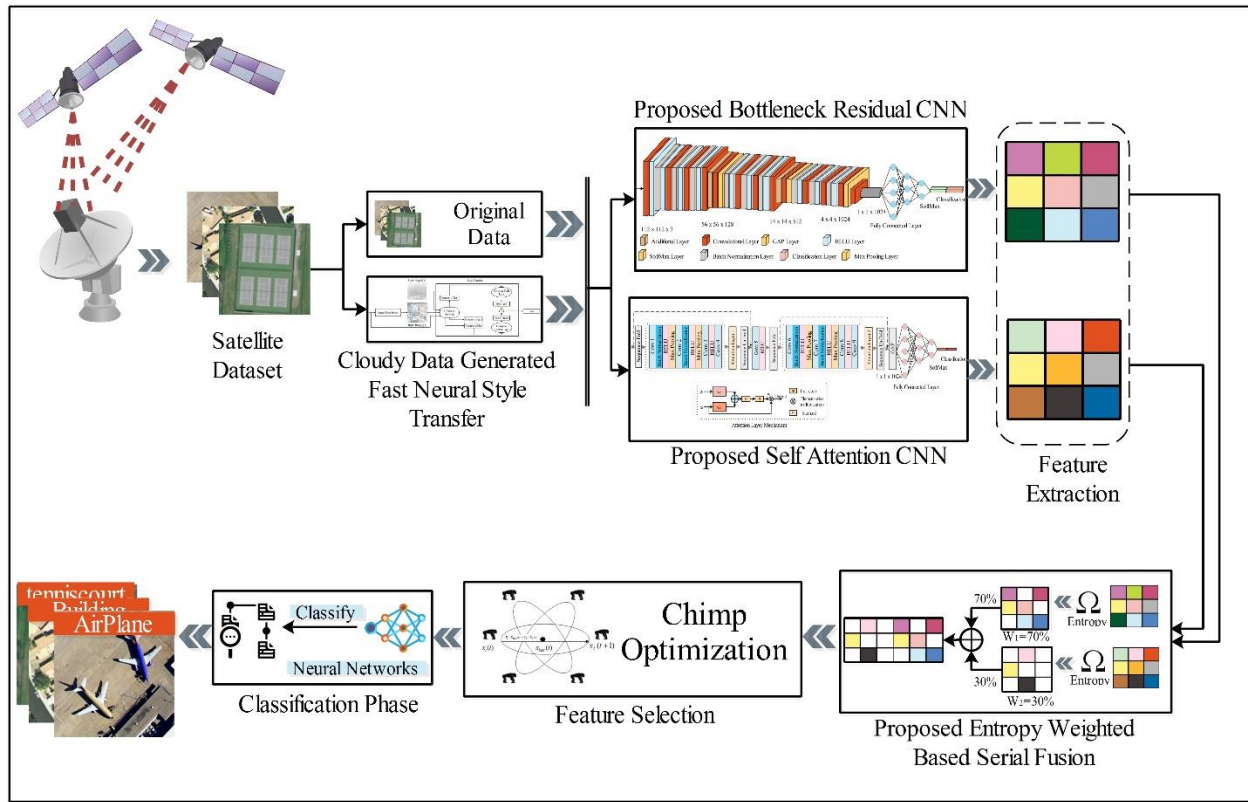


Figure 2: Proposed framework for the classification of land use scenes from satellite images

A. Proposed Neural Style Cloudy Dataset Generation

i) Original Dataset

In this research, we selected the publically available land use dataset for the experimental purpose. The selected dataset is UC-Merced land use (<https://captain-whu.github.io/BED4RS/>). The nature of dataset was RGB. The UC-Merced dataset consists of 21 land use classes: agriculture, airplane, baseball, diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home park, and others. The total number of instances in the UC-Merced dataset is 2,100 as shown in Table 1. Every class consists of 100 samples, and the dimension of each sample is 256×256 pixels. A few sample images have been shown in Figure 3.

ii) Proposed Cloudy Dataset

The second dataset is generated using the UC-Merced land use dataset by using a fast neural style transfer approach [45] to create a cloudy effect in the dataset because the UC-Merced dataset has clear scenes in the dataset. Both datasets are separately used for the experimental process. Figure 3 presents a few samples of cloudy UC-Merced land use dataset.

The fast neural style transfer technique is the extended form of neural style transfer [46], which creates a stylish image by giving the reference style image. We used a fast neural style transfer approach to generate the cloudy effect in the UC-Merced dataset. In fast neural style transfer, an image transformer, also called an image-to-image network, is employed.

This network is primarily divided into three main phases; the initial phase takes an input an RGB image size of $256 \times 256 \times$

3 denoted with $f(x)$ and samples it to feature map dimension of $64 \times 64 \times 3$.

Table 1: Description of selected and generated satellite dataset

DATASETS	TOTAL IMAGES	TRAINING/T ESTING
UC-MERCED	21,00	1,050
CLOUDY UC-MERCED	21,00	1,050

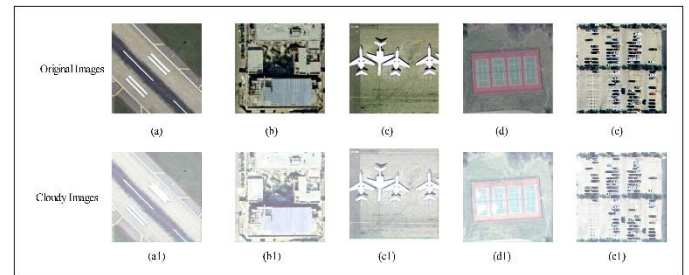


Figure 3: Samples of original and cloud-generated UC-Merced dataset. Samples (a), (b), (c), (d), and (e) presented the original images, and (a1), (b1), (c1), (d1), and (e1) presented the generated cloudy UC-Merced land use images.

The second phase consists of five identical residual blocks. The final phase of the network performs upsampling on the feature map to restore it to the image's original dimensions, ultimately producing an altered image, which is denoted with $g(x)$ in Figure 4. In addition, a two-residual block-based CNN is created to extract the features of the content and style images at multiple levels. These multilayer features are utilized to

determine content and style loss, respectively. The estimation of the style transfer loss includes the utilization of a 2-residual block-based (CNN). This process involves providing the input images, denoted as $f(x)$, the altered images, denoted as $g(x)$, and the style image, denoted as $S(x)$, to the proposed CNN. The network is capable of extracting several features from the given images. The content loss is computed by the algorithm via the utilization of the spatial features of both the input image $f(x)$ and the output image $g(x)$. In addition, the calculation of style loss is performed by combining the stylistic features of the generated output image $g(x)$ and the style image $S(x)$. Ultimately, the overall loss is computed by combining the losses acquired from content and style. The content and style loss are mathematically formulated as:

$$\mathcal{L}_{con} = \frac{1}{\alpha} \sum_{i=1}^{\alpha} \mu([\varphi(f(x)_i) - \varphi(g(x)_i)]^2) \quad (1)$$

$$\mathcal{L}_{sty} = \frac{1}{\alpha} \sum_{i=1}^{\alpha} \sum_{j=1}^4 [G(\varphi_j(f(x)_i)) - G(\varphi_j(S(x)))]^2 \quad (2)$$

Where $f(x)$ presents the input image $g(x)$ denotes the transformed image, α is denoted the mini-batch size, and φ presents the extracted activation layer, and G is denoted the gram matrix. Gram matrix is employed to calculate the style of image by considering the correlation among the multiple features in the hidden layers of proposed model. The gram matrix is measured using equation (3).

$$G(\alpha, \beta, \gamma) = \frac{1}{H \times W \times C} \sum_{h=1}^H \sum_{w=1}^W \varphi_{\tau}(h, w, c_i) \varphi_{\tau}(h, w, c_j) \quad (3)$$

Where φ_{τ} are denoted the activations for the τ^{th} image in the mini-batch. The high-level architecture of a style transfer network is presented in Figure 4.

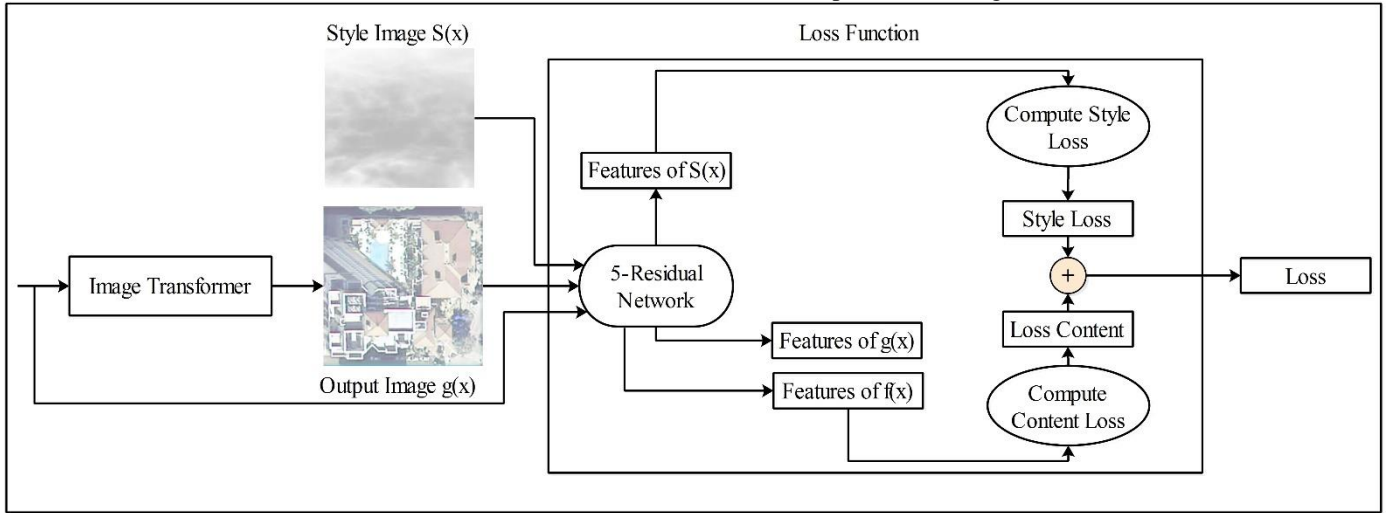


Figure 4: High-level architecture of fast style transfer network for the generating of cloudy effect in satellite images

B. Proposed Bottleneck Residual Architecture

A bottleneck residual block is a building block utilized in deep convolutional neural networks, specifically in ResNet (Residual Network) architectures. ResNet is a well-known deep-learning architecture with excellent results in several computer vision tasks[47]. In this work, we designed a CNN that contains three bottleneck residual blocks. The proposed CNN accepts the input size of $224 \times 224 \times 3$. After input, the convolutional layer and RELU activation are attached with a 3×3 kernel size, 32 depths, and a 2×2 stride. The first bottleneck block has a convolutional layer with a 1×1 kernel size; the depth is 64, and the stride is 1. The second convolutional has a 3×3 kernel size, depth is 96, and stride is one, and the third convolutional has a 1×1 kernel size, depth is 32, and stride is 1. Each convolutional is attached to batch normalization and RELU activation in the block. After that, one convolutional and one max pooling layer is attached with 3×3 , 3×3 kernel size and pool size, 128 depth, 2×2 , 2×2 stride, and padding are the same. The second bottleneck block starts with batch normalization, RELU activation, and a convolutional layer with a 1×1 kernel size, 256 depths, and a 1×1 stride. In addition, batch

normalization and RELU are attached, followed by the second convolutional layer with a 3×3 kernel, size 256, depth, and a 1×1 stride. After two bottleneck blocks, a convolutional layer having a 3×3 filter size, 256 depths, and stride 2×2 , and max pooling with 3×3 pool size and 2×2 stride, respectively, and the RELU activation layer is attached. After that, the third bottleneck block is created, the first and third convolutional layers of this block having 1×1 filter size, 1024, 512 depth, and 1×1 stride, respectively. The middle convolutional has a 3×3 filter size, 1024 depths, and a 1×1 stride. Each convolutional layer of this block is attached with one batch normalization and one RELU activation. After the third block, one convolutional layer is attached with a 3×3 filter size, 1024 depths, and 2×2 strides. The closing layers are global average pooling, fully connected, softmax, and classification. The total parameters of this model is 16.3million with 11 layers are convolutional out of 43 layers. The designed model is trained on both datasets and deep features are extracted from the global average pool layer. The dimension of extracted features is $N \times 1024$. The architecture of the proposed bottleneck residual-based CNN is presented in Figure 5.

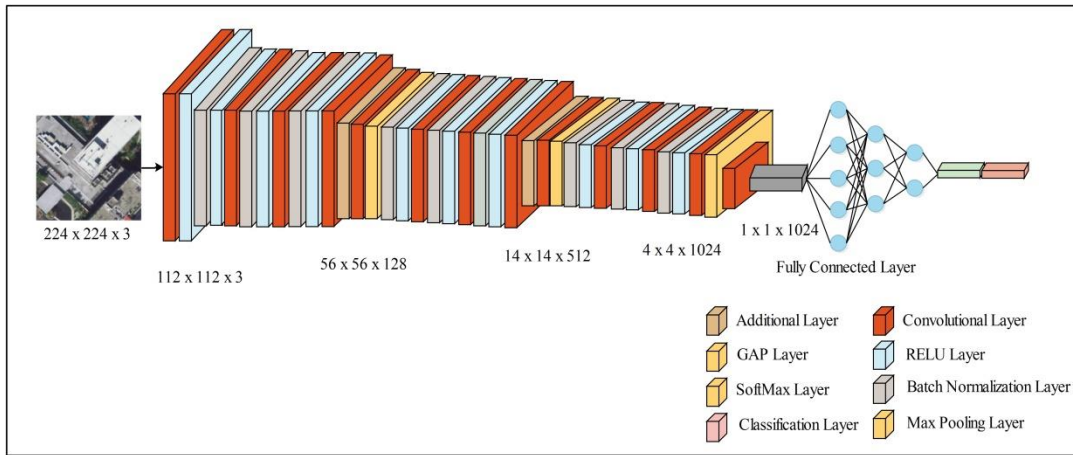


Figure 5: Architecture of the proposed bottleneck residual-based CNN for the classification of land use images

C. Proposed Self-Attention Architecture

A deep learning strategy with a 2D structure was designed for the classification problem as the images from the satellite have tiny objects to see. For this challenge, an efficient model was designed. The proposed model strategy consists of two sequence folds with CNN and two attention mechanism blocks, as shown in Figure 6. The attention mechanism was utilized to improve the capturing of deep features and concentration on small instances of feature maps. The proposed attention-based CNN accepts the input size of $227 \times 227 \times 3$. The sequence folded has been added to transform the data into a sequence. Four convolutional layers are attached with filter size 3×3 , depth 32, 64, 96, and stride 1×1 , 2×2 , 1×1 , and 1×1 , respectively. The first two convolutional layers are attached with batch normalization, RELU activation, and max pooling, having a 3×3 pooling size with a 2×2 stride.

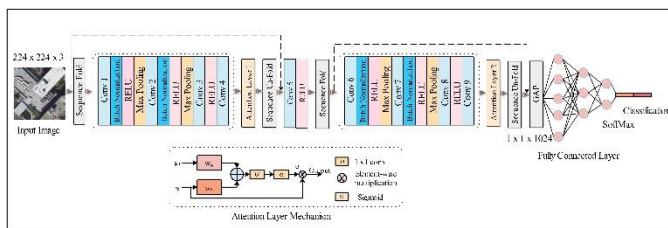


Figure 6: Proposed attention mechanism-based CNN for the classification of satellite images

After that, one attention layer was added, and the sequence was unfolded and placed. In addition, the convolutional layer is placed with a 3x3 filter size, 128 depth, and 2x2 strides. RELU activation is attached to introduce the non-linearity, and the sequence folded layer is attached to transform the feature map into sequences. After that, two convolutional layers are attached with 1x1 kernel size, 256, 512, 1024 depth, and 1x1 stride. After both convolutional layers, batch normalization, RELU activation, and max pooling layer with 3x3 pooling size and 2x2 stride are attached. Moreover, the attention layer and sequence unfolds are attached, and the feature map is passed to the global average pooling layer. At the end of the proposed network, the fully connected layer is placed with an output size of 21 classes, and the softmax and classification layers are attached to the

final output. The total parameters of the attention model is 32.5 million with 45 layers and 13 layers are convolutional layers. The proposed was trained on the selected datasets and global average pool activation is utilized for the attention feature extraction. The size of extracted information is $N \times 1024$. The complete proposed self-attention architecture is presented in Figure 6.

D. Proposed Weighted Entropy Serial Fusion

Feature fusion is an essential process in which heterogeneous information from multiple sources is integrated to yield an enhanced outcome [48]. We proposed a feature fusion approach named weighted entropy serial fusion in this work. This approach calculates the entropy for deep features extracted from the bottleneck CNN and self-attention CNN. Following that, the weight is measured for both entropy-calculated features. The 70% and 30% of the weights are considered for entropy bottleneck CNN and self-attention CNN features. After that, the weighted vectors are fused using a serial approach, as shown in Figure 7.

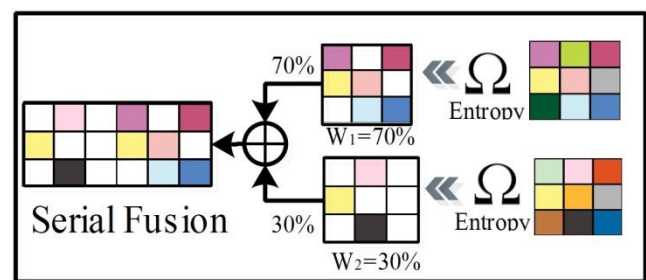


Figure 7: Proposed weighted entropy serial fusion

The mathematical description of this proposed approach is defined as suppose the proposed bottleneck CNN and self-attention CNN feature vectors $\mathbb{V}_I^{bf}, \mathbb{V}_I^{sf}$ having dimensions of $I \times 1024$. Where I is the number of samples and bf, sf are the features. The entropy of both vectors is computed as follows:

$$\omega_{entropy} = \sum_M \sum_N p(M, N) \log p(M, N) \quad (4)$$

The entropy of \mathbb{V}_I^{bf} , and \mathbb{V}_I^{sf} is defined as:

$$\Omega(\mathbb{V}_I^{bf})_{ent} = \omega_{entropy}(\mathbb{V}_I^{bf}) \quad (5)$$

$$\Omega(\mathbb{V}_I^{sf})_{ent} = \omega_{entropy}(\mathbb{V}_I^{sf}) \quad (6)$$

Where $\Omega(\mathbb{V}_I^{bf})_{ent}$ and $\Omega(\mathbb{V}_I^{sf})_{ent}$ presented the calculated entropy of both the proposed bottleneck CNN and self-attention CNN features vectors and $\omega_{entropy}$ denoted the formula that is utilized to measure the entropy. After performing the entropy process, the 70 and 30% weights are measured by using equations 7 and 8.

$$w_{\Omega(\mathbb{V}_I^{bf})_{ent}} = \text{int}64(bf \times 0.70) \quad (7)$$

$$w_{\Omega(\mathbb{V}_I^{sf})_{ent}} = \text{int}64(sf \times 0.30) \quad (8)$$

Where $w_{\Omega(\mathbb{V}_I^{bf})_{ent}}$ and $w_{\Omega(\mathbb{V}_I^{sf})_{ent}}$ presented the output of weighted vectors, and bf and sf denoted the number of features. In the last phase, the weighted entropy features are fused using a serial approach mathematically presented in Equation 9.

$$\mathbb{V}_{serialFusion} = \begin{pmatrix} w_{\Omega(\mathbb{V}_I^{bf})_{ent}} \\ w_{\Omega(\mathbb{V}_I^{sf})_{ent}} \end{pmatrix}_{I \times F} \quad (9)$$

Where $\mathbb{V}_{serialFusion}$ is denoted the final output, we obtained the final output dimension of $N \times 1024$ for both datasets. The resultant features are passed to the proposed Chimp optimization for feature selection.

D. Proposed Chimp optimization

In computer vision, feature selection (FS) is the systematic process of finding a subset of features from an initial set to eliminate redundant, irrelevant, or noisy features. Adopting FS improves accuracy and a shortcoming in computational time [49]. In this research, we employed chimp optimization for feature selection. The Chimp optimization was introduced by Khishe and Mosavi in 2020. The basic idea underlying chimp optimization originates from the sexual drive and intellect of chimpanzees, which distinguishes them from other social hunters by their collective haunting behavior [50]. This technique's simplicity, capacity to avoid local optima, quick convergence, and low computational cost have made it widely used to identify the optimal answers for complex optimization issues [51]. The mathematical models of driving, chasing, blocking, attacking, and independent groups. The Chimp optimization is given as:

i) Driving and Chasing the Prey

Hunting prey occurs between the processes of exploration and exploitation. The mathematical model of driving and chasing the prey is formulated in equations 10 and 11.

$$z = |k \cdot s_{prey}(t) - j \cdot s_{chimp}(t)| \quad (10)$$

$$s_{chimp}(t+1) = s_{prey}(t) - a \cdot z \quad (11)$$

Where t represents the number of iterations and a, j, k indicates the coefficients of the vector. s_{prey} And s_{chimp} denoted the position vector of prey and chimps. So, a, j, k are measured as follows:

$$a = 2 \cdot u \cdot x_1 - u \quad (12)$$

$$k = 2 \cdot x_2 \quad (13)$$

$$j = \text{chaotic} - \text{value} \quad (14)$$

Where u indicates reduced non-linearity x_1 and x_2 is represented as the range of random vectors and j is based on the chaotic value.

ii) Attacking Method (exploitation phase)

Two approaches are devised to model the aggressive behavior exhibited by chimpanzees mathematically. Chimpanzees can engage in various behaviors, such as driving, blocking, and chasing, to investigate and ultimately surround the location of their prey. The hunting process is typically carried out by chimpanzees that assume the role of aggressors. Drivers, barrier, and chaser chimpanzees occasionally engage in the hunting process. Regrettably, there is a lack of available information about the prey's optimal location within an abstract search space. To mathematically model the behavior of chimpanzees, it is postulated that the initial attacker (optimal solution), driver, barrier, and chasers possess superior knowledge regarding the whereabouts of potential prey. Four of the most optimal solutions acquired thus far are retained, and the remaining chimpanzees are compelled to adjust their positions based on the location of the most superior chimpanzees. The relationship is denoted by equations (15) to (17).

$$z_{Attacker} = |k_1 s_{Attacker} - j_1 s|, z_{Barrier} = |k_2 s_{Barrier} - j_2 s|, \quad (15)$$

$$z_{Chaser} = |k_3 s_{Chaser} - j_3 s|, z_{Driver} = |k_4 s_{Driver} - j_4 s|. \\ s_1 = s_{Attacker} - a_1(z_{Attacker}), s_2 = s_{Barrier} - a_2(z_{Barrier}), \quad (16)$$

$$s_3 = z_{Chaser} - a_3(z_{Chaser}), s_4 = s_{Driver} - a_4(z_{Driver}). \\ s(t+1) = \frac{s_1 + s_2 + s_3 + s_4}{4} \quad (17)$$

iii) Social Incentive (sexual motivation)

Acquiring a meal and consequent social motivations, such as sexual and grooming behaviors, during the last stage leads to chimpanzees relinquishing their hunting responsibilities. Consequently, they endeavor to acquire meat through coercive and disorderly means. The observed disorderly conduct exhibited by chimpanzees in the concluding phase effectively addresses the issues of being trapped in local optima and the sluggish rate of convergence encountered when attempting to solve complex, high-dimensional problems. The mathematical model is expressed in equation 18 to update the chimpanzees' position during optimization.

$$s_{chimp(t+1)} = \begin{cases} s_{prey(t)} - a \cdot z, & \text{if } \mu < 0.5 \\ \text{Chaotic}_{value} & \text{if } \mu \geq 0.5 \end{cases} \quad (18)$$

After every iteration of the chimp algorithm, fitness value is measured by utilizing the KNN. The KNN classifier returned the cost value. The cost function of KNN is mathematically defined in equation (19)

$$\text{cost} = \alpha \times \text{error} + \beta \times \left(\frac{\text{No of selected features}}{\text{Max of features}} \right) \quad (19)$$

Where α is 0.99 and β is 0.01, and error is calculated by using equation (20).

$$E_{err} = 1 - A_{cc} \quad (20)$$

The optimized features have a size of $N \times 192$ for the original UC-Merced dataset and $N \times 207$ cloud effect UC-Merced dataset. The hold-out method is employed for validation purposes the value of the hold-out is 0.2 which indicates that

the 20% of features are used as validation. The optimization is stopped when it approaches 100 iterations.

The final features are further passed to neural network classifiers for the final classification

IV. RESULTS AND ANALYSIS

In this section, the experimental results of the proposed framework have been presented. The UC-Merced land use and generated cloudy effect in UC-Merced datasets (as discussed under the section Data collection and preprocessing) are used for the experimental process. The datasets were divided into 50:50 ratios. The 50% of the images were used for training, and the remaining data was used for testing. We selected $k = 10$ due to the widespread use of k -fold cross-validation and its capacity to achieve an optimal ratio between computational costs and variance, which is associated with the generalizability of the efficiency estimate. The experiment utilized feature dimensions of $N \times 1024$; a lesser value of k was found to be ineffective, whereas a performance consisting of 10 was observed when k was set to that value. So, for all experiments were performed using ten k -fold cross-validation. The hyperparameters like an optimizer, mini-batch size, learning rate, and epochs with values SGDM 16, 0.0001, and 10 were used to train the proposed models. Neural Network classifiers were chosen based on varying hidden layers to evaluate the classification outcomes. The narrow neural network consists of 10 hidden layers and one fully connected layer, whereas the medium neural network has 25 hidden layers and one fully connected layer. The neural network is defined by its extensive architecture, consisting of 100 hidden layers and one fully connected layer. Tri-layered neural network architecture consisting of three layers was used, including 10 hidden layers and 2 fully connected layers. The bi-layered neural network model consisted of 10 hidden layers and 3 fully interconnected layers. The classification results are evaluated using precision, recall, f1-score, accuracy, false discovery rate (FDR), false negative rate (FNR), Fowlkes Mallows index (FM), and computation time in seconds. The FDR, FNR, and FM were calculated using equations 21, 22, and 23.

$$FDR = 1 - PPV \quad (21)$$

$$FNR = 1 - TRP \quad (22)$$

$$FM = \sqrt{TRP \times PPV} \quad (23)$$

The PPV is a positive predictive value known as precision, and TPR is a true positive rate known as recall. All simulations were

conducted using a core i7 13gen configured with a 12GB 3060RTX NVIDIA graphics card, 500 SSD, and 128GB of RAM. The results are divided into two major sections. In the first section, we discuss the results of the UC-Merced dataset, and in the second section, the results of the cloudy UC-Merced dataset are explained.

A. Experimental Results of UC-Merced Dataset

In the first section, the results of the UC-Merced dataset have been presented. The features are extracted from the proposed bottleneck residual CNN and self-attention CNN. Table 2 shows the results of the bottleneck residual CNN model. In this table, the WNN classifier achieved a higher accuracy of 95.7%. The precision rate is 95.9%, the recall rate is 95.71%, the F1-score is 95.80%, FDR is 4.23, FNR is 4.29, and FM is 95.73%.

In Contrast, Table 3 illustrates the results of the self-attention CNN model with a maximum of 95.6% accuracy of the WNN classifier. The precision rate is 95.6%, the recall rate is 95.1%, the F1-score is 95.3%, FDR is 4.4, FNR is 4.9, and FM is 95.3%. The computation is noted for all listed classifiers in both cases. The MNN classifier has the lowest computation time, 10.52 (s). In contrast, the TNN classifier has the highest computation among all listed classifiers, 33.657 (s) using bottleneck residual CNN weights. At the same time, the MNN classifier takes 33.988 (s) for execution.

In the next phase, the extracted features are fused using the proposed weighted entropy serial fusion, and the results of the proposed fusion are presented in Table 4. In this table, the WNN classifier attained the highest accuracy of 98.9%. The precision rate is 98.83%, the recall rate is 98.94%, the f1-score is 98.8%, the FDR is 1.2, the FNR is 1.1 and FM is 98.8%. These parameters are also measured for the listed classifiers. The accuracy is significantly improved from the previous experiments. The computation time is recorded for all listed classifiers, and it is observed that after the fusion process, the computation time also increases. The maximum time recorded for the TNN classifier is 53.23 (s) among all classifiers, while the lowest computation is 19.22 (s) for the MNN classifier. Figures 8, 9, and 10 shows the confusion matrices of both architectures that can utilize to verify the performance of WNN.

Table 2: Results of bottleneck residual CNN for the classification of UC-Merced land use dataset

CLASSIFIER	PRECISION	RECALL	F1- SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	88.71	88.00	88.35	88.0	11.27	12.00	88.34	21.79
MNN	89.30	93.61	91.40	93.6	10.7	6.39	91.42	10.52
WNN	95.90	95.71	95.80	95.7	4.23	4.29	95.73	13.79
BNN	84.83	89.04	86.88	84.7	15.17	10.9	86.90	23.02
TNN	82.65	89.35	85.86	83.4	17.35	10.6	85.93	33.65

Table 3: Results of Self-attention CNN model for the classification of UC-Merced land use dataset

CLASSIFIER	PRECISION	RECALL	F1-SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	88.8	88.5	88.6	88.6	11.2	11.5	88.6	35.869
MNN	93.8	93.7	93.7	93.8	6.2	6.3	93.7	33.988
WNN	95.6	95.1	95.3	95.6	4.4	4.9	95.3	41.501
BNN	86.3	86.1	86.2	86.2	13.7	13.9	86.1	67.002
TNN	74.0	81.4	77.5	81.4	26.0	18.6	77.6	112.88

Table 4: Results of the proposed weighted entropy serial fusion for the classification of UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1-SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	94.98	94.72	94.8	94.8	5.1	5.3	94.7	22.60
MNN	98.24	98.15	98.1	98.2	1.8	1.9	98.1	19.22
WNN	98.83	98.94	98.8	98.9	1.2	1.1	98.8	24.86
BNN	92.92	92.91	92.9	93.0	7.1	7.1	92.9	40.01
TNN	92.61	92.42	92.5	92.5	7.4	7.6	92.4	53.23

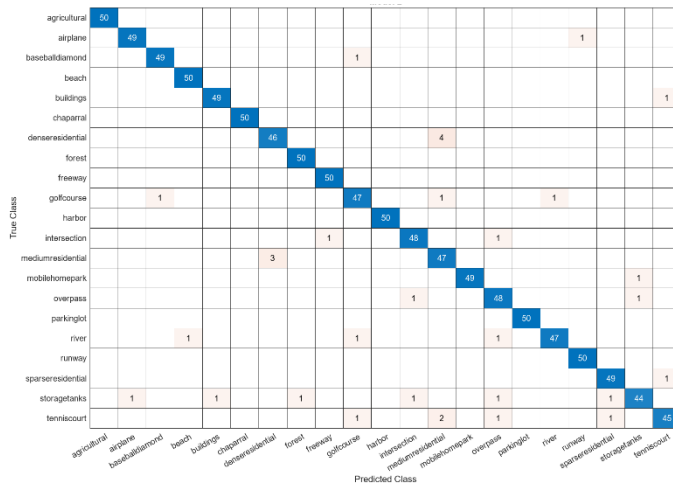


Figure 8: Confusion matrix of WNN classifier with bottleneck residual CNN weights for the classification of UC-Merced Land use dataset

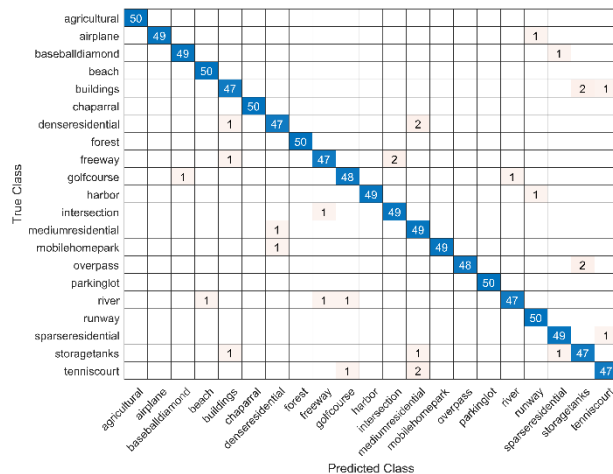


Figure 9: Confusion matrix of WNN classifier with Self-attention weights for the classification UC-Merced land use dataset

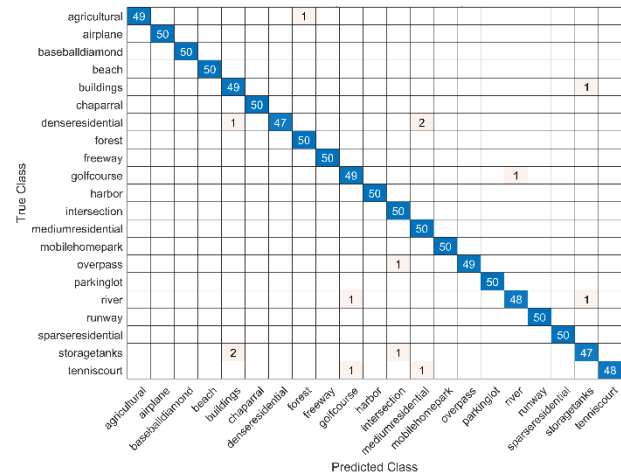


Figure 10: Confusion matrix of the proposed weighted entropy serial fusion for classification of UC-Merced dataset

In the last phase, chimp optimization is employed for the feature selection to reduce duplicate and redundant information from the fused weights. Table 5 shows the results of binary chimp optimization, with the highest accuracy of the WNN classifier at 99.0%. The value of the precision rate is 99.0%, the recall rate is 98.9%, the f1-score is 98.9%, FDR is 1.0, FNR is 1.1 and FM is 98.9%. The confusion matrix of this WNN classifier is shown in Figure 11. The execution time is noted for the listed classifiers; it is observed that after feature selection, the computation is significantly reduced. The TNN classifier takes 14.67 (s) for execution, whereas, in the previous experiment, the TNN takes 53.23 (s). Compared to the previous phases, the selection method shows the better computational time.

Table 5: Classification results of Chimp optimization for the UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1-SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	94.79	94.62	94.6	94.7	5.3	5.4	94.6	8.10
MNN	96.96	96.83	96.8	97.0	3.1	3.2	96.8	7.33
WNN	99.05	98.91	98.9	99.0	1.0	1.1	98.9	14.14
BNN	93.52	93.44	93.4	93.4	6.5	6.4	93.4	13.93
TNN	90.21	89.91	89.9	90.2	9.8	9.7	89.9	14.67

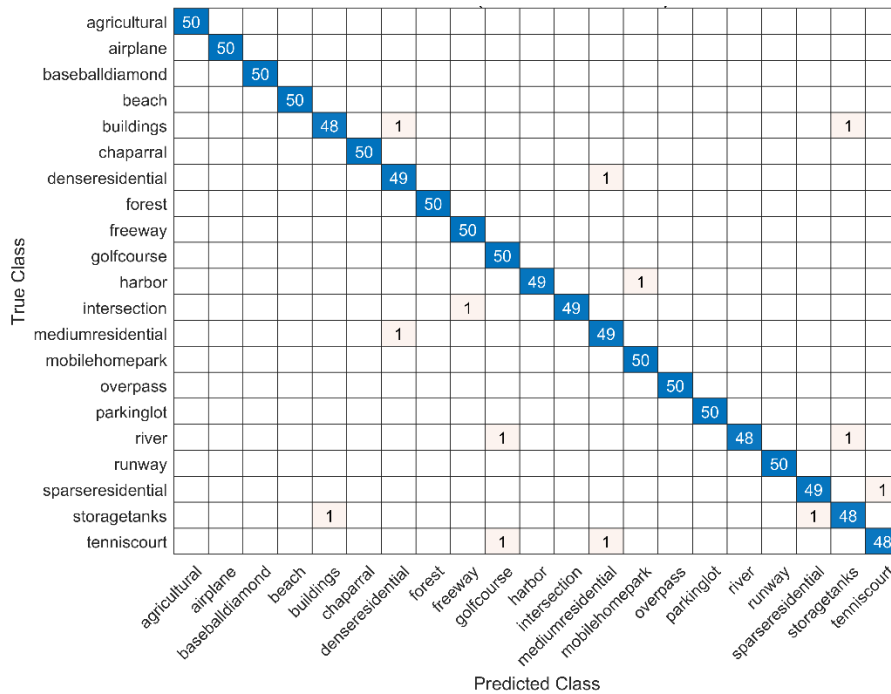


Figure 11: Confusion matrix of WNN classifier with chimp optimization for classification of UC-Merced land use dataset

B. Experimental results of Cloudy UC-Merced Dataset

In this subsection, the classification results of the cloudy UC-Merced dataset have been presented. In the initial phase, the bottleneck residual CNN and self-attention CNN were trained on a cloudy UC-Merced dataset, and the trained model was employed for the feature extraction. The classification results of bottleneck residual CNN are illustrated in Table 6. In this table, the WNN classifier attained the highest accuracy of 96.3%. The other computed parameters, including precision, recall, F1-score, FDR, FNR, and FM, have values of 96.3%, 96.2%, 96.2%, 3.7, 3.8, and 98.2%, respectively.

In addition, all classification results are measured by the self-attention CNN model described in Table 7. From this table, the WNN classifier achieves a higher accuracy of 95.0%, a precision rate is 95.0%, a recall rate is 94.9, F1-score is 94.9%, FDR is 5.0, FNR is 5.1, and FM is 94.9%. The bottleneck residual CNN with the WNN classifier was observed to outperform the previous model. Figure 13 shows the confusion matrixes for further clarification. The computation time is also noted for both methods. The bottleneck CNN with MNN classifier is executed in 17.48 (s), while the highest time is recorded from self-attention CNN with TNN, which was 76.27 (s).

The proposed weighted entropy serial fusion is employed in the next stage to improve the classification results. Table 8 shows the results of the WNN classifier with the highest accuracy of 99.4%. The other parameters are precision, recall, f1-score, FDR, FNR, and time. The values of these parameters are 99.45%, 99.34%, 99.3%, 0.6, 0.5, and 99.3%, respectively. The confusion matrix is presented in Figure 14 for the further verification of numerical analysis. By employing this fusion, the computation time of each classifier is significantly higher than in the previous experiments. The MNN classifier has the shortest execution time, which is 22.38 (s), while in the last two results, the MNN classifier has 17.48 (s) and 20.09 (s), respectively.

To reduce the computation time and irrelevant information from the fused weights. We employed chimp optimization for the best feature selection. The classification results of chimp optimization have been presented in Table 9. From this table, it was observed that the WNN classifier achieved 99.0% accuracy. The precision rate is 98.9%, the recall rate is 98.8%, the F1-score is 98.8%, FDR is 1.1, FNR is 1.0, and FM is 98.8%. The confusion matrix is visually illustrated in Figure 15. By this method, the accuracy is decreased by 0.3%, but the computation time of the WNN classifier is improved by 13.08 (s). The MNN classifier requires 6.97 (sec), considered the

shortest time from all experiments on the cloud UC-Merced dataset.

Table 6: classification results of bottleneck residual CNN with WNN classifier for the classification of cloudy UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1- SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	92.5	92.3	92.4	92.4	7.5	7.7	92.3	20.46
MNN	95.9	95.8	95.8	95.8	4.1	4.2	95.8	17.48
WNN	96.3	96.2	96.2	96.3	3.7	3.8	98.2	23.75
BNN	90.5	90.4	90.4	90.5	9.5	9.6	90.4	41.92
TNN	89.7	89.6	89.6	89.7	10.3	10.2	89.6	52.92

Table 7: Classification results of the proposed Self-attention CNN for cloud UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1- SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	87.0	86.91	86.9	87.0	13.0	13.1	86.9	24.66
MNN	93.7	93.63	93.6	93.6	6.3	6.2	93.6	20.09
WNN	95.0	94.95	94.9	95.0	5.0	5.1	94.9	24.17
BNN	83.5	83.43	83.4	83.4	16.5	16.6	83.4	47.27
TNN	80.9	80.87	80.8	80.9	19.1	19.2	80.8	76.29

Table 8: Classification results of the proposed weighted entropy serial fusion in cloudy UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1- SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	94.84	94.76	94.7	94.8	5.2	5.3	94.7	28.70
MNN	98.43	98.32	98.3	98.5	1.6	1.7	98.3	22.38
WNN	99.45	99.34	99.3	99.4	0.6	0.5	99.3	26.38
BNN	94.22	94.06	94.1	94.1	5.8	6.0	94.0	59.42
TNN	91.91	91.89	91.8	91.9	8.1	8.2	91.8	81.36

Table 9: Classification results of Chimp optimization for cloudy UC-Merced dataset

CLASSIFIER	PRECISION	RECALL	F1- SCORE	ACCURACY	FDR	FNR	FM	TIME
NNN	96.2	96.1	96.1	96.2	3.8	3.7	96.1	7.64
MNN	97.6	96.9	96.9	97.6	2.4	2.3	96.9	6.97
WNN	98.9	98.8	98.8	99.0	1.1	1.0	98.8	13.30
BNN	93.7	93.6	93.6	93.6	6.3	6.2	93.6	13.14
TNN	92.8	92.7	92.7	92.7	7.2	7.1	92.7	16.01

> REPLACE THIS LINE WITH YOUR PAPER IDENTIFICATION NUMBER (DOUBLE-CLICK HERE TO EDIT) <

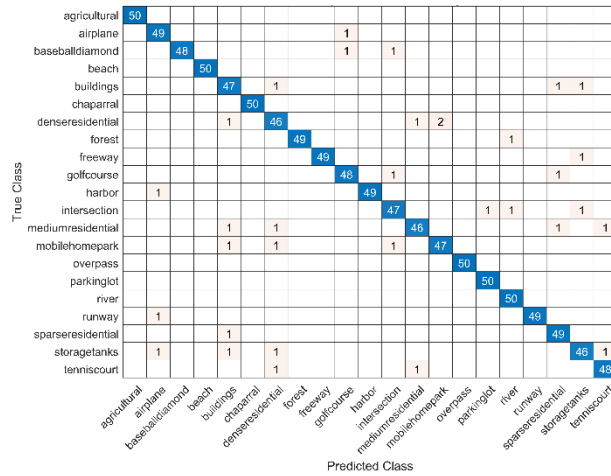


Figure 12: Confusion metric of bottleneck residual CNN with WNN classifier for cloudy UC-Merced dataset

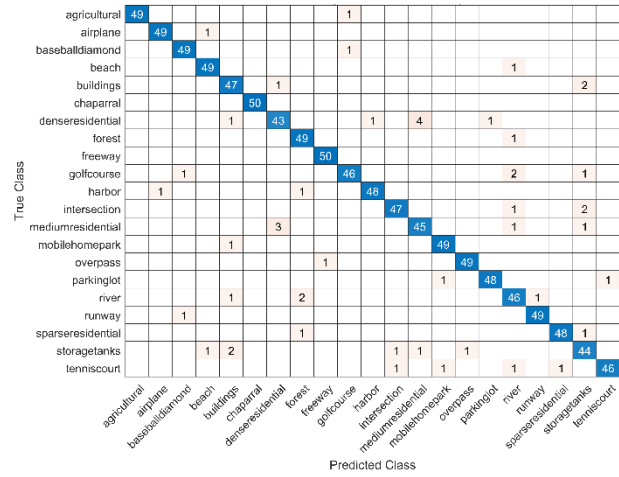


Figure 13: Confusion matrix of self-attention CNN with WNN classifier for cloudy UC Merced dataset

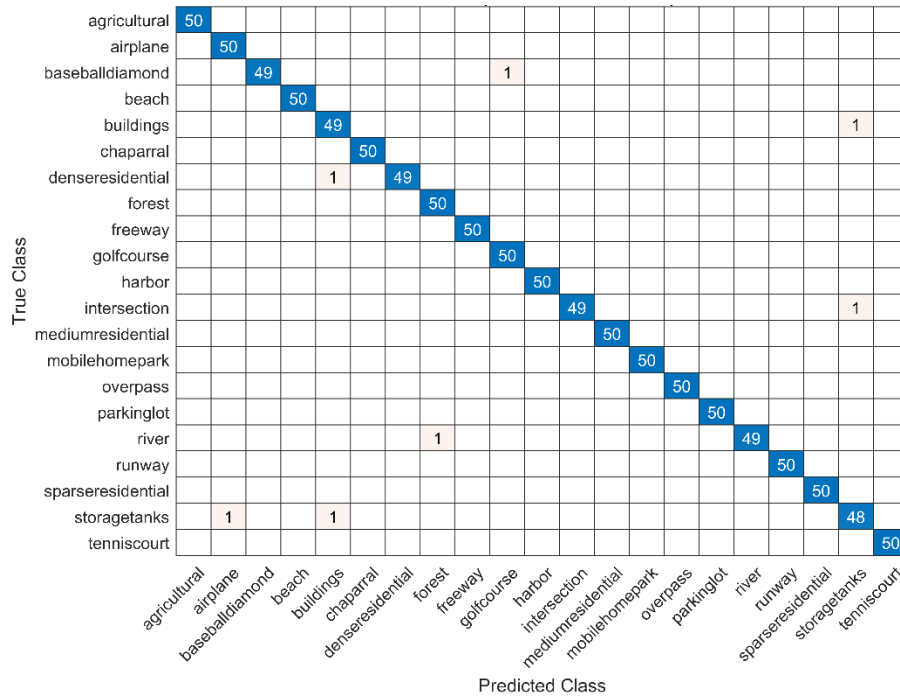


Figure 14: Confusion matrix of proposed feature fusion with WNN classifier for cloud UC-Merced dataset

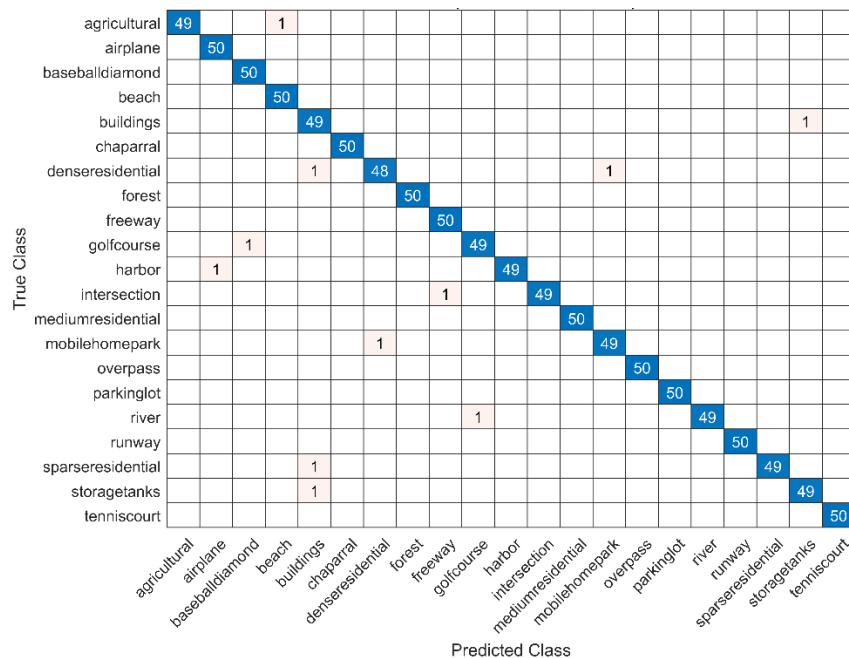


Figure 15: Confusion metric of chimp optimization on WNN classifier in cloudy UC-Merced dataset

C) Comparison with Recent Techniques

A Comprehensive comparison with recent techniques was conducted based on the UC-Merced dataset, presented in Table 10. The table shows that the proposed framework's performance outperforms the recent techniques listed. [52] published recently, in which the authors used a CNN-based hybrid system to classify land use images and achieved 93.3%. Similarly, [53], [54], and [49] employed the latest deep learning techniques and achieved 98.0%, 95.0%, and 91.8% accuracy, respectively. Meanwhile, [47] used the most recent quantum-based classical

image processing to classify land use images and achieved 85.5% accuracy. In addition, [48] employed the voting-based mechanism for land use image processing to achieve high precision in 2021, and they achieved 95.1% accuracy. At the end of the 2021 year, [49] a retrieval system was designed using fuzzy clustering, and they achieved 97.0% accuracy. In addition, a few labeled results of the proposed method have been illustrated in Figure 16. In this figure, it is shown that the predicted labels are generated using both CNN architectures.

Table 10: Comparison with state-of-the-art recent techniques

Reference	Year	Dataset	Method	Accuracy
[52]	2023	UC-Merced land use	CNN based hybrid system	93.3%
[53]	2023	UC-Merced land use	Comparative analysis of deep learning pre-trained models	98.0%
[54]	2023	UC-Merced land use	Multi-Branch Deep Learning Framework	95.0%
[55]	2022	UC-Merced land use	Quantum-classical image processing	85.5%
[56]	2022	UC-Merced land use	Voting based mechanism for high precision land use image processing	95.1%
[57]	2021	UC-Merced land use	Multi-label classification using deep models	91.8%
[58]	2021	UC-Merced land use	Satellite image retrieval system using fuzzy clustering	97.0%
Proposed			UC-Merced land use	99.0%

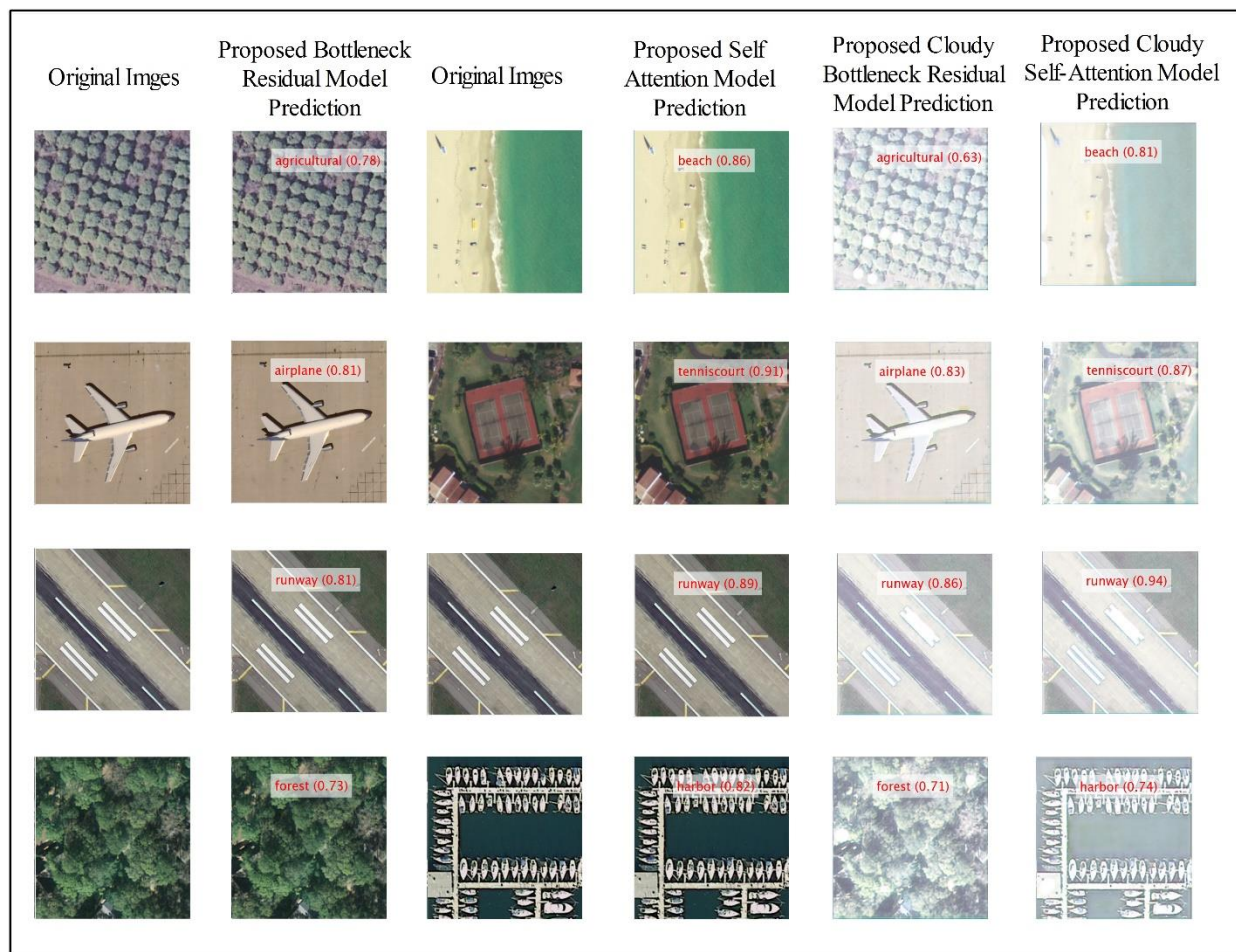


Figure 16: Prediction of the proposed bottleneck residual and self-attention model

V. CONCLUSION

The goal of scene classification is to classify an image using semantic categories. This is a particularly challenging problem since distinct objects may exist at different sizes and orientations, and there may be a lot of variation in the land covers that belong to a particular class. An intelligent architecture based on deep learning was presented in this work to classify land use in remote-sensing images. A neural-style architecture has been included in the suggested architecture to create new cloudy remote sensing images to create a new dataset that would aid in the generalizability of the proposed framework. Two new deep architectures have been proposed: 3-layered bottleneck CNN architecture and 3-layered bottleneck self-attention CNN architecture. Both architectures have been trained on the original and newly prepared cloudy datasets and extracted features from the deeper layers. The self-attention architecture performance was better at this stage. Later, we proposed a fusion technique based on weighted entropy that fused the information of both architectures. The fused features improved the performance; however, the computational time has been increased. To handle this problem, we proposed a Chimp Optimization method that selects the best features and removes the irrelevant information. The selection process improved the proposed architecture's performance and reduced computational time.

The drawback of this work is the weighted fusion process that added little redundant information; however, the strength of this work is better accuracy on a smaller amount of remote sensing images for the training. In the future, a comparative study will be considered that compares several architectures' performance on the selected datasets of this work.

ACKNOWLEDGMENT

This work is funded by the Researchers Supporting Project number (RSP2024R157), King Saud University, Riyadh, Saudi Arabia.

DATA AVAILABILITY

The selected datasets is UC-Merced land use which is publically available on (<https://captain-whu.github.io/BED4RS/>), and the second dataset is generated by UC-Merced by adding the cloudy effect in the images using fast neural style transfer.

REFERENCES

- [1] M. A. Shafaey, M. A.-M. Salem, H. M. Ebied, M. N. Al-Berry, and M. F. Tolba, "Deep learning for satellite image classification," in *International Conference on Advanced Intelligent Systems and Informatics*, 2018: Springer, pp. 383-391.

- [2] H. Ferdous, T. Siraj, S. J. Setu, M. M. Anwar, and M. A. Rahman, "Machine learning approach towards satellite image classification," in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering: Proceedings of TCCE 2020*, 2021: Springer, pp. 627-637.
- [3] N. Laban, B. Abdellatif, H. M. Ebied, H. A. Shedeed, and M. F. Tolba, "Multiscale satellite image classification using deep learning approach," *Machine Learning and Data Mining in Aerospace Technology*, pp. 165-186, 2020.
- [4] M. Alkhelaiwi, W. Boulila, J. Ahmad, A. Koubaa, and M. Driss, "An efficient approach based on privacy-preserving deep learning for satellite image classification," *Remote Sensing*, vol. 13, no. 11, p. 2221, 2021.
- [5] S. A. Boyle, C. M. Kennedy, J. Torres, K. Colman, P. E. Pérez-Estigarribia, and N. U. de la Sancha, "High-resolution satellite imagery is an important yet underutilized resource in conservation biology," *PLoS One*, vol. 9, no. 1, p. e86908, 2014.
- [6] P. Zhang, Y. Ke, Z. Zhang, M. Wang, P. Li, and S. Zhang, "Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery," *Sensors*, vol. 18, no. 11, p. 3717, 2018.
- [7] M. San Miguel *et al.*, "Challenges in complex systems science," *The European Physical Journal Special Topics*, vol. 214, pp. 245-271, 2012.
- [8] L. Cao, C. Wang, and J. Li, "Vehicle detection from highway satellite images via transfer learning," *Information Sciences*, vol. 366, pp. 177-187, 2016/10/20/ 2016, doi: <https://doi.org/10.1016/j.ins.2016.01.004>.
- [9] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geoscience Frontiers*, vol. 7, no. 1, pp. 3-10, 2016/01/01/ 2016, doi: <https://doi.org/10.1016/j.gsf.2015.07.003>.
- [10] Y. Yuan, L. Lin, Z.-G. Zhou, H. Jiang, and Q. Liu, "Bridging optical and SAR satellite image time series via contrastive feature extraction for crop classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 222-232, 2023.
- [11] B. Feizizadeh, D. Omarzadeh, M. Kazemi Garajeh, T. Lakes, and T. Blaschke, "Machine learning data-driven approaches for land use/cover mapping and trend analysis using Google Earth Engine," *Journal of Environmental Planning and Management*, vol. 66, no. 3, pp. 665-697, 2023.
- [12] J. Zheng *et al.*, "Surveying coconut trees using high-resolution satellite imagery in remote atolls of the Pacific Ocean," *Remote Sensing of Environment*, vol. 287, p. 113485, 2023.
- [13] H. Ahn, S. Lee, H. Ko, M. Kim, S. W. Han, and J. Seok, "Searching similar weather maps using convolutional autoencoder and satellite images," *ICT Express*, vol. 9, no. 1, pp. 69-75, 2023.
- [14] Y. G. Yuh, W. Tracz, H. D. Matthews, and S. E. Turner, "Application of machine learning approaches for land cover monitoring in northern Cameroon," *Ecological Informatics*, vol. 74, p. 101955, 2023.
- [15] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Chinese Conference on Biometric Recognition*, 2004: Springer, pp. 179-186.
- [16] J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 944-952, 2002.
- [17] X. Ma, J. Xu, J. Pan, J. Yang, P. Wu, and X. Meng, "Detection of marine oil spills from radar satellite images for the coastal ecological risk assessment," *Journal of Environmental Management*, vol. 325, p. 116637, 2023.
- [18] T. Hofmann, "Probabilistic latent semantic analysis," *arXiv preprint arXiv:1301.6705*, 2013.
- [19] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270-279.
- [20] H. Jelodar *et al.*, "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, pp. 15169-15211, 2019.
- [21] S. Bulut, A. Günlü, and G. Çakır, "Modelling some stand parameters using Landsat 8 OLI and Sentinel-2 satellite images by machine learning techniques: a case study in Türkiye," *Geocarto International*, vol. 38, no. 1, p. 2158238, 2023.
- [22] A. Becker, S. Russo, S. Puliti, N. Lang, K. Schindler, and J. D. Wegner, "Country-wide retrieval of forest structure from optical and SAR satellite imagery with deep ensembles," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 269-286, 2023.
- [23] I. Dimitrovski, I. Kitanovski, D. Koccev, and N. Simidjievski, "Current trends in deep learning for Earth Observation: An open-source benchmark arena for image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 18-35, 2023.
- [24] T. Zhang, T. Zeng, and X. Zhang, "Synthetic aperture radar (SAR) meets deep learning," vol. 15, ed: MDPI, 2023, p. 303.
- [25] A. Tariq and S. Qin, "Spatio-temporal variation in surface water in Punjab, Pakistan from 1985 to 2020 using machine-learning methods with time-series remote sensing data and driving factors," *Agricultural Water Management*, vol. 280, p. 108228, 2023.
- [26] A. A. Oliveira, M. S. Buckeridge, and R. Hirata Jr, "Detecting tree and wire entanglements with deep learning," *Trees*, vol. 37, no. 1, pp. 147-159, 2023.
- [27] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [28] Y. n. Zhou, J. Luo, L. Feng, Y. Yang, Y. Chen, and W. Wu, "Long-short-term-memory-based crop classification using high-resolution optical images and multi-temporal SAR data," *GIScience & Remote Sensing*, vol. 56, no. 8, pp. 1170-1191, 2019.
- [29] M. Tarasiou, E. Chavez, and S. Zafeiriou, "ViTs for SITs: Vision Transformers for Satellite Image Time Series," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10418-10428.
- [30] J. Shi, Z. Li, and H. Zhao, "Feature selection via maximizing inter-class independence and minimizing intra-class redundancy for hierarchical classification," *Information Sciences*, vol. 626, pp. 1-18, 2023.
- [31] S. Gadamsetty, R. Ch. A. Ch. C. Iwendi, and T. R. Gadekallu, "Hash-based deep learning approach for remote sensing satellite imagery detection," *Water*, vol. 14, no. 5, p. 707, 2022.
- [32] K. Vani, "Deep learning based forest fire classification and detection in satellite images," in *2019 11th International Conference on Advanced Computing (ICoAC)*, 2019: IEEE, pp. 61-65.
- [33] T. T. Nguyen *et al.*, "Monitoring agriculture areas with satellite images and deep learning," *Applied Soft Computing*, vol. 95, p. 106565, 2020.
- [34] A. Unnikrishnan, V. Sowmya, and K. P. Soman, "Deep learning architectures for land cover classification using red and near-infrared satellite images," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18379-18394, 2019/07/01 2019, doi: [10.1007/s11042-019-7179-2](https://doi.org/10.1007/s11042-019-7179-2).
- [35] J. Chen, C. Wang, Z. Ma, J. Chen, D. He, and S. Ackland, "Remote sensing scene classification based on convolutional neural networks pre-trained using attention-guided sparse filters," *Remote Sensing*, vol. 10, no. 2, p. 290, 2018.
- [36] D. Datta *et al.*, "A hybrid classification of imbalanced hyperspectral images using ADASYN and enhanced deep subsampled multi-grained cascaded forest," *Remote Sensing*, vol. 14, no. 19, p. 4853, 2022.
- [37] M. M. Khalid and O. Karan, "Deep Learning for Plant Disease Detection," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2, pp. 75-84, 2024.
- [38] C.-J. Zhang, X.-J. Wang, L.-M. Ma, and X.-Q. Lu, "Tropical cyclone intensity classification and estimation using infrared satellite images with deep learning," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2070-2086, 2021.
- [39] M. A. Kadhim and M. H. Abed, "Convolutional neural network for satellite image classification," *Intelligent Information and Database Systems: Recent Developments 11*, pp. 165-178, 2020.
- [40] H. Li, L. Deng, C. Yang, J. Liu, and Z. Gu, "Enhanced YOLO v3 tiny network for real-time ship detection from visual image," *Ieee Access*, vol. 9, pp. 16692-16706, 2021.
- [41] H. Wu, H. Zhang, J. Zhang, and F. Xu, "Fast aircraft detection in satellite images based on convolutional neural networks," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015: IEEE, pp. 4210-4214.
- [42] Q. Liu *et al.*, "DeepSat V2: feature augmented convolutional neural nets for satellite image classification," *Remote Sensing Letters*, vol.

- 11, no. 2, pp. 156-165, 2020/02/01 2020, doi: 10.1080/2150704X.2019.1693071.
- [43] T. N. Hannevik, Ø. Olsen, A. N. Skauen, and R. Olsen, "Ship detection using high resolution satellite imagery and space-based AIS," in *2010 International WaterSide Security Conference*, 2010: IEEE, pp. 1-6.
- [44] Y. Chaudhary, M. Mehta, N. Goel, P. Bhardwaj, D. Gupta, and A. Khanna, "YOLOv3 remote sensing SAR ship image detection," in *Data Analytics and Management: Proceedings of ICDAM*, 2021: Springer, pp. 519-531.
- [45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 2016: Springer, pp. 694-711.
- [46] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365-3385, 2019.
- [47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.
- [48] H. A. Madni *et al.*, "Improving Sentiment Prediction of Textual Tweets Using Feature Fusion and Deep Machine Ensemble Model," *Electronics*, vol. 12, no. 6, p. 1302, 2023.
- [49] R. Kundu and S. Chattopadhyay, "Deep features selection through genetic algorithm for cervical pre-cancerous cell classification," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13431-13452, 2023.
- [50] H. Jia, K. Sun, W. Zhang, and X. Leng, "An enhanced chimp optimization algorithm for continuous optimization domains," *Complex & Intelligent Systems*, pp. 1-18, 2021.
- [51] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert systems with applications*, vol. 149, p. 113338, 2020.
- [52] F. G. Y. Çiklaçandır and S. Utku, "Performance Comparison of CNN Based Hybrid Systems Using UC Merced Land-Use Dataset UC Merced Land-Use Veri Kümesi Kullanılarak CNN Tabanlı Hibrit Sistemlerin Performans Karşılaştırması."
- [53] A. A. Adegun, S. Viriri, and J.-R. Tapamo, "Review of deep learning methods for remote sensing satellite images classification: experimental survey and comparative analysis," *Journal of Big Data*, vol. 10, no. 1, p. 93, 2023/06/02 2023, doi: 10.1186/s40537-023-00772-x.
- [54] S. D. Khan and S. Basalamah, "Multi-Branch Deep Learning Framework for Land Scene Classification in Satellite Imagery," *Remote Sensing*, vol. 15, no. 13, p. 3408, 2023.
- [55] A. Chalumuri, R. Kune, S. Kannan, and B. Manoj, "Quantum-classical image processing for scene classification," *IEEE Sensors Letters*, vol. 6, no. 6, pp. 1-4, 2022.
- [56] J. Zhao *et al.*, "A high-precision image classification network model based on a voting mechanism," *International Journal of Digital Earth*, vol. 15, no. 1, pp. 2168-2183, 2022.
- [57] A. Kumar *et al.*, "Multilabel classification of remote sensed satellite imagery," *Transactions on emerging telecommunications technologies*, vol. 32, no. 7, p. e3988, 2021.
- [58] P. Kavitha and P. Vidhya Saraswathi, "Content based satellite image retrieval system using fuzzy clustering," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 5541-5552, 2021.

Authors BIOS

Ameer Hamza: He is currently Phd scholar at HITEC University, Pakistan. His major interest include object detection and recognition, video surveillance, medical, and agriculture using deep learning and machine learning. He has published four impact factor papers to date.

Muhammad Attique Khan: (Member IEEE) earned his Master's and Ph.D. degrees in Human Activity Recognition for Application of Video Surveillance and Skin Lesion Classification using Deep Learning from COMSATS University Islamabad, Pakistan. He is currently an Assistant Professor of the Computer Science Department at HITEC University Taxila, Pakistan. His primary research focus in recent years is medical imaging, COVID-19, MRI analysis, Video Surveillance, Human Gait Recognition, and Agriculture Plants

using Deep Learning. He has above 280 publications that have more than 10,000+ citations and an impact factor of 850+ with h-index 61 and i-Index 165. He is reviewer of several reputed journals such as IEEE transaction on Industrial Informatics, IEEE transaction of Neural Networks, Pattern Recognition Letters, Multimedia Tools and Application, Computers and Electronics in Agriculture, IET Image Processing, Biomedical Signal processing Control, IET Computer Vision, Eurasipe Journal of Image and Video Processing, IEEE Access, MDPI Sensors, MDPI Electronics, MDPI Applied Sciences, MDPI Diagnostics, and MDPI Cancers.

Shams ur Rehman received his bachelor master degree in computer science from HITEC University, Taxila, Pakistan. He is co-author of two journal papers and three papers are in under review. He is an excellent programmer and submitted several applications for customers. His research interest including machine learning and deep learning for remote sensing and medical imaging.

Dr Mohammed Al-Khalidi is a Senior Lecturer in Cyber Security and MSc Cyber Security Course Leader at the Department of Computing and Mathematics, Manchester Metropolitan University, UK. He is a senior member of IEEE and member of the British Computer Society. His past assignments include Lecturer at the Department of Computer Science, Edge Hill University, and Research Officer at the School of Computer Science and Electronic Engineering, University of Essex, where he also received his PhD degree. Prior to that, he worked in industry as a Telecommunications Engineer at several leading mobile telecommunication companies in the Middle East & north Africa. He has more than ten years of industrial experience in mobile core network performance optimization.

Ahmed Ibrahim Alzahrani is currently an Associate Professor with the Department of Computer Science, Community College, King Saud University. He acts as the Head of the Informatics Research Group, and a member of the scientific council-King Saud University.

Nasser Alalwan is currently an Assistant Professor of computer science with the Department of Computer Science, Community College, King Saud University. His current research interests include database, semantic web, ontology, electronic and mobile services, and information technology management.

Anum Masood received her Ph.D. degree in Computer Science and Engineering from Shanghai Jiao Tong University, Shanghai, China in 2019. Her B.Sc. and M.Sc. degrees in Computer Science are from the COMSATS University Islamabad, Islamabad, Pakistan. She worked as a Lecturer with the Department of Computer Science, COMSATS University Islamabad, from 2014 to 2020. Currently, she is a postdoctoral researcher at the Norwegian University of Science and Technology (NTNU), Norway, and affiliated with PET Centre, St. Olav's Hospital, Trondheim, Norway. She also worked as visiting researcher at Institute of Neuroscience and Medicine (INM), Forschungszentrum Jülich, Institute for Cardiogenetics (ICG), University of Luebeck, Germany and Liverpool John Moores University (LJMU), U.K. Her research interests include medical image analysis, automated cancer detection, machine learning, and image processing.