IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 December 2023, accepted 6 January 2024, date of publication 16 January 2024, date of current version 11 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3354976

RESEARCH ARTICLE

A Deeper Look Into Remote Sensing Scene Image Misclassification by CNNs

ANAS TUKUR BALARABE AND IVAN JORDANOV
School of Computing, University of Portsmouth, PO12 UP Portsmouth, U.K.
Corresponding author: Anas Tukur Balarabe (anas.tukur-balarabe@port.ac.uk)

This work was part of a Ph.D research supported by the Petroleum Technology Development Fund (PTDF)-Nigeria.

ABSTRACT As deeper and lighter variations of convolutional neural networks (CNNs) continue to break accuracy and efficiency records, their applications for solving domain-specific challenges continue to widen, particularly in computer vision and pattern recognition. The feat achieved by these end-to-end learning models can be attributed to their ability to extract local and global discriminative features for effective classification. However, in land use and land cover classification (*LULC*), inner-class variability and outer-class similarity could cause a classifier to confuse one image's discriminative features with another's, leading to inefficiency and poor classification. In this work, we deviate from the conventional approach of classifying high-resolution remote sensing images (*HRRS*) by proposing a framework for comparing and combining images of different simple classes into superclasses based on spatial, textural, and colour similarities. To achieve this, we implement the *Bhattacharyya* metric for colour-based similarity analysis, a combination of *LBP*s (Local Binary Pattern), the Earth Mover's Distance, and Euclidean Distance for the texture and spatial similarity analysis in addition to the structural similarity index (*SSIM*). A pre-trained CNN model (*Xception*) is then fine-tuned to classify the superclasses and the original classes of the Aerial Image (*AID*), the *UC Merced*, the Optical Image Analysis and Learning (*OPTIMAL-31*), and *NWPU-RESIS45* datasets. Results show that methodically combining overlapping classes into superclasses reduces the possibility of misclassifications and increases the efficiency of CNNs. The model evaluation further indicates that this approach can boost classifiers' robustness and significantly reduce the impact of inner-class variability and outer-class similarity on their performance.

INDEX TERMS Image similarity metrics, Euclidean distance, local binary pattern, transfer learning, scene classification.

I. INTRODUCTION

Satellite sensors acquire remote sensing images under varying altitudes and constantly changing atmospheric conditions. Because of the height at which the acquisition is made, each image usually covers a large area and land type, resulting in dataset inner-class variability and outer-class image similarity [1], [2], [3], [4], [5]. For the inner-class variability, images within the same class or category tend to vary due to factors such as the atmospheric conditions during acquisition, the angle at which an image is acquired, and the number of noninformative features that might appear in different parts of the images. These challenges make repurposing pre-trained deep learning models for classifying images challenging, especially in earth observation tasks [2]. Several techniques that attempt to improve the overall efficiency of deep learning classifiers have been developed over the years to address the issue [1], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Despite the record-breaking performance of the proposed models, creating an efficient scene image classifier that is robust against misclassification due to inner-class variability and outer-class similarity remains elusive [2]. Recently, deep learning [14], [15], and transfer learning models [18] have shown dramatic improvements in classification accuracy and efficiency. Some of the most prominent deep learning models that have been widely used in machine vision include *AlexNet* [19], *VGGNet* [20], *Xception* [21], *GoogleNet* [22], *MobileNet* [23], *EfficientNet* [24], *DenseNet* [25] and

The associate editor coordinating the review of this manuscript and approving it for publication was Gerardo Di Martino.

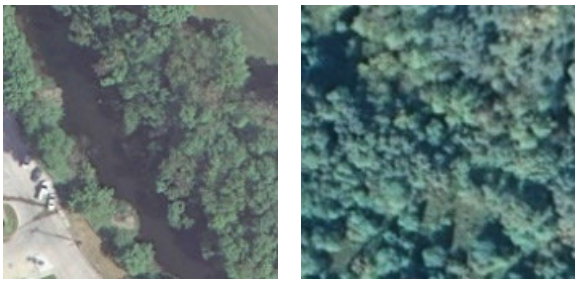


FIGURE 1. ***HRRS*** **scene classification images: (a) a river and (b) a forest.**

ShuffleNet [26]. The success of the transfer learning architectures in scene classification is connected to their ability to work with a small amount of training data and extract class-specific features that contain the discriminative information necessary for the accurate classification of scene images [12], [16], [22]. However, despite the efficacy demonstrated by these models, they can still be confused by inner-class variability and outer-class similarity in scene classification datasets. Fundamentally, satellite images usually contain aspects of other objects besides their salient features [2]. The additional objects are often the main features that characterise another scene. For example, in Fig.1, a significant part of image 1(a) of a *river* is covered by *vegetation*, which, on the other hand, represents the class *forest* as in image 1(b).

Such intricate similarities could mislead a classifier to assign the same label to the two distinct images. The misclassifications due to inner-class variability or outer-class similarity could affect the quality of decision-making where the outcome of a classifier forms an integral part of the decision-making process. Generally, *LULC* is done using the multiclass approach, where each category is assigned a label depending on the classes in a dataset [20], [21], [22], [23], [24], [25]. The drawback is that standard CNNs and fine-tuned transfer learning models could confuse one category with another due to the complex similarities and subtle differences among some classes [1]. On the one hand, the standard CNNs use the output of the last fully connected layer to predict the label of each image. However, this causes the images' local discriminative features to be discarded, leading to inaccurate classifications [2], [32]. On the other hand, the transfer learning models, which use a feature extraction strategy, are often modified to combine the local and global discriminative characteristics as an improvement to the traditional CNNs [1], [2], [14]. While there is room for improvement in the classification of *HRRS* scene images, other researchers are exploring new frontiers for applying images obtained in one city to classify land-use and land-cover scenes in another [33]. More application domains are evolving in smart cities, such as land-use management, urban planning, building counting, change detection, road extraction, and damage assessment. In damage assessment, for example, the goal is not to differentiate between the types of buildings that survived a natural disaster but to ascertain the number of still-standing buildings. Likewise, in road extraction, the primary aim is to extract a road infrastructure regardless of its make or type. As a result, the spatial complexities of the images and the scenes they represent could be leveraged to merge some classes into superclasses. For some tasks of land-use management, such as change detection [34], [35], [36], [37], building detection/extraction [38], [39], [40], [41], [42], [43], [44], vegetation detection [45], [46], [47], road detection/extraction [25], [48], [49], roof type classification [50], [51], a particular area of interest could contain several scenes considered as a single entity in scene classification datasets. In all these tasks, the general assumption is that the otherwise independent scenes represent a single entity; hence, they are assigned a single label regardless of the colour, textural or spatial differences. What we set out to do in this paper is to design a framework for comparing and combining images based on their spatial, textural, and colour similarities. To investigate the viability of this approach, selected scene classification dataset classes are incorporated into several superclasses using the scores given by the algorithm we developed. A fine-tuned transfer learning model is then used to classify the superclasses and the original classes; the results are critically analysed and compared.

In summary, the contributions of this paper are as follows:

1. A framework for comparing images based on their spatial, textural, and colour properties is proposed. This framework can also be applied for content-based image retrieval on natural images.
2. A transfer learning strategy is developed using the *Xception* model, which is not widely used despite its lightweight nature and efficient use of training parameters.
3. A new *HRRS* scene image classification technique that could be used in land-use management, early warning systems, disaster management, building counting, vegetation analysis, etc., is proposed and presented.

A. CONVOLUTIONAL NEURAL NETWORKS

The most critical components of a CNN model are its layers, which are the building blocks of any CNNs architecture and define most of its parameters [16], [51], [52], [53]. The arrangement of the layers to form a viable CNN depends on the problem at hand, the domain's expertise, and technical knowledge [55]. Generally, each layer of a CNN (apart from the last one or two fully connected layers) constitutes three significant operations: convolution, activation, and pooling [19]. The convolutional layers generate feature maps by convolving 2D filters (kernels, or a collection of kernels in the 3D case) over an image. The feature maps are passed through a non-linearity of an activation function, such as *ReLU*, *Sigmoidal*, or others, to determine the output of a neuron. The *ReLU* activation function is commonly used based on its lower susceptibility to saturation, leading to faster training and learning compared to other activation functions [16], [52], [56], [57]. The role of pooling is to

merge semantically similar features into a single segment and to achieve spatial invariance by reducing the resolution of the feature maps (downsampling). As images pass through a CNN architecture, blobs, edges, and other primitives are detected and combined locally to form motifs. These detected features are arranged to shape distinctive parts assembled to configure objects. The final (classification) layer, which usually consists of the *Softmax* transfer function, takes in the output of the previous (fully connected) layer and produces classification probabilities depending on the number of output classes.

B. TRANSFER LEARNING

Despite the notable popularity of machine learning, most of its models work on the principle that the training and the test data share the same feature space and distribution; therefore, any change to the feature space and the distribution of the training data affects the overall model prediction ability [58]. Similarly, training models from scratch requires a humungous amount of training data, which could be unavailable or prohibitively expensive to collect [59]. The transfer learning approach aims to solve this problem by transferring features learned from a 'source' to a 'target' domain [59], [60], [61]. Thus, the need to acquire extensive training data is addressed, and the model training time is reduced significantly [60].

C. TRANSFER LEARNING NOTATIONS AND TERMINOLOGIES

To better understand the mathematical representation of different aspects of transfer learning, some basic notations and definitions are briefly explained here.

Definition 1 (Domain [58]): A domain D is denoted by $\boldsymbol{D}=\{\boldsymbol{\chi}, \boldsymbol{P}(\boldsymbol{X})\}$ is defined by two components:

i. A feature space: $\boldsymbol{\chi}$;
ii. Marginal probability distribution $\boldsymbol{P}(\boldsymbol{X})$, such that $\boldsymbol{X}=\{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\} \in \boldsymbol{\chi}$.

Definition 2 (Task [58]): A Task is given by $\boldsymbol{T}=\{\mathbf{Y}, \mathbf{f}(\cdot)\}$ comprises of two parts:

i. A label space $\boldsymbol{Y}=\{\mathbf{y_1}, \ldots, \mathbf{y_m}\}$;
ii. An objective predictive function $\boldsymbol{f}(\cdot)$ that has not been observed or learned.

A corresponding label $\mathrm{f}(x_i)$ of an unlearned instance, x_i, can be predicted using the function $\mathrm{f}(\cdot)$. The function $\mathrm{f}(x_i)$ can also be presented as $\mathrm{P}(y_i|x_i)$ when considered from a probabilistic angle [58], [60].

Definition 3 (Transfer Learning [60]): Formally defined, transfer learning is such that given a source domain D_s, a learning task T_s and a target domain D_t, its goal is to make some improvement on the learning of a predictive function $\mathrm{f}(\cdot)$ in the target domain D_t, by leveraging the knowledge acquired from D_s and T_s, where $D_s \neq D_t$ and $T_s \neq T_t$ [58].

II. RELATED WORK

The main idea behind remote sensing is to acquire vital information about objects or scenes within an area of interest. It is a means of acquiring local, regional and global earth observation information [62]. Thus, its application in many areas, including land-use and land cover classification (*LULC*) [63], [64], [65]. As one of the most crucial subfields of remote sensing, *LULC* is widely applied in understanding the socio-economic land usage, interpreting the physical land features on earth, monitoring natural and artificial changes to land, and predicting how these changes can affect the environment and the socio-economic life of its inhabitants [66], [67], [68]. Research in scene image classification and its potential applications has recently gained traction, paving the way for more innovations in processing and utilising *HRRS* images [1], [68]. Previously, scene classification techniques were primarily based on handcrafted feature extraction methods [1], [2]. Some of the traditional approaches are based on colour extraction [69], texture motifs extraction [70], texture descriptors [71], object structure [72], shape and other properties [73]. However, these low-level and middle-level image feature extraction techniques have many constraints that limit their applicability to remote sensing scene image classification. Therefore, to fill the gap created by the traditional feature extraction models' weaknesses [2], attention was shifted to deep learning (DL) models for their remarkable improvements in classification accuracy, faster convergence speed and computational efficiency. One of the advantages of their layered architecture is that it allows the extraction of discriminative information and compression of data without throwing away the essential features of an image [74], [75]. Among the DL approaches, the transfer learning models are more efficient than the traditional CNNs models. Architectures such as *AlexNet* [19], *Xception* [21], *MobileNet* [23], and *ResNet* [76] have been widely repurposed and used for tasks of scene classification. As mentioned above, one of the main motivations for using the transfer learning models is the lack of enough training data [60].

Notwithstanding, using the transfer learning models trained on ImageNet might not always yield the desired accuracy level on remote sensing images because of the differences in resolution, spatial features, and texture between satellite and natural images. Therefore, architectural improvements are always necessary in order to use a pre-trained model for scene classification [77]. More so, scene classification images have two properties that confuse deep learning classifiers: inner-class variability and outer-class similarity [2]. As shown in Fig. 1, the classes exhibiting these peculiarities share some textural or spatial features. Since multiclass classification requires each category to be assigned a label and reorganising the categories with high-level similarity into clusters could not be carried out subjectively, we proposed using image similarity techniques to recombine the images into superclasses based on the colour spatial and textural similarities to establish whether a logical clustering of the basic classes of scene classification datasets into superclasses can reduce training time, improve classification accuracy and reduce computational complexity.

III. IMAGE SIMILARITY METRICS

A. IMAGE HISTOGRAM SIMILARITY METRICS

As the field of image processing evolves, the application of image similarity and recognition also continues to develop [78]. One of the most effective ways of assessing image similarities is through image histogram comparison [79]. A histogram H_x of a digital image x is a graphical representation of the frequency of occurrence of each grey level in the image x [75], or H_y can be easily represented as a one-dimensional vector with domain {0,..., K-1}, where K is the maximum pixel value of 256. The Distance between the image histograms is valuable for feature extraction, content-based image retrieval (*CBIR*), image pattern recognition, and feature clustering [74].

1) Bhattacharyya

is a bin-to-bin metric of histogram similarity, focusing on the probability distributions between two histograms. The image similarity value of this metric ranges from 0 to 1, where 0 indicates an accurate matching and 1 shows otherwise (the lower the value, the higher the similarity between the images) [79], [80].

$$D_H = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} \quad (1)$$

where $\boldsymbol{H}$ represents a histogram with $\boldsymbol{N}$ bins and $\boldsymbol{I}$ is the colour intensity.

B. IMAGE TEXTURE SIMILARITY

In machine vision and image processing, a texture is defined as a function of an image's local spatial patterns and grey-level pixel intensities [82], [83]. It also refers to the shape of the contents of an image [84]. There are several texture similarity analysis metrics; however, local binary pattern (*LBP*) has been selected for this research due to its efficiency and superior performance.

1) LOCAL BINARY PATTERN (*LBP*)

According to Huang et al. [85], the primary aim of the *LBP* operator is to give an efficient summary of the structure of an image. Texture analysis is used in machine vision for various applications such as pattern recognition, object detection, medical image analysis, etc. [83]. Image texture analysis techniques have been classified into four categories: model-based, structural, transformed-based, and statistical, in which the *LBP* falls [83]. The local binary pattern technique was initially developed for texture description and is widely used for its efficiency and low execution time [86]. The classic version of *LBP* generates a label for each pixel in an image by thresholding the 3×3 pixel neighbourhood and converting the result to a binary value [84], [86]. The set of binary values produced by each operation on an image patch is used as a local patch descriptor, and the combination of these descriptors gives the overall texture description of the image [87]. By converting the *LBP* binary codes to a decimal number, a feature vector that represents the textural information of an image is generated [88]. Subsequently, histograms created from these vectors form the basis for comparing the images using a distance metric [89]. Figures 2 and 3 show raw images of the *Building* superclass in the *AID* dataset, the Euclidean distance, the earth mover's distance values between the first image and each image in the class, and the corresponding histograms generated from the converted *LBP* codes.

$$LBP(x_c, y_c) = \sum_{n=0}^{n-1} 2^n s(i_n - i_c) \quad (2)$$

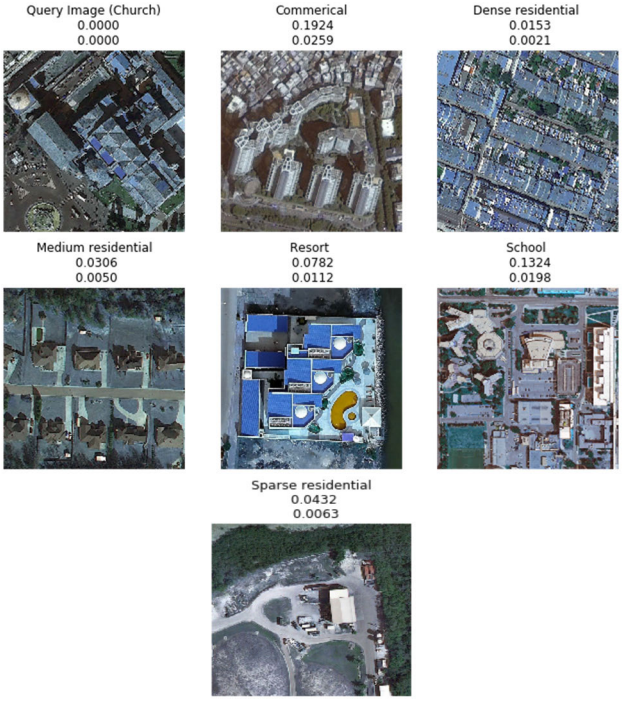


FIGURE 2. Images of *Buildings* superclass of *AID* and corresponding *Euclidean* and *Earth Mover's* distance values, respectively.

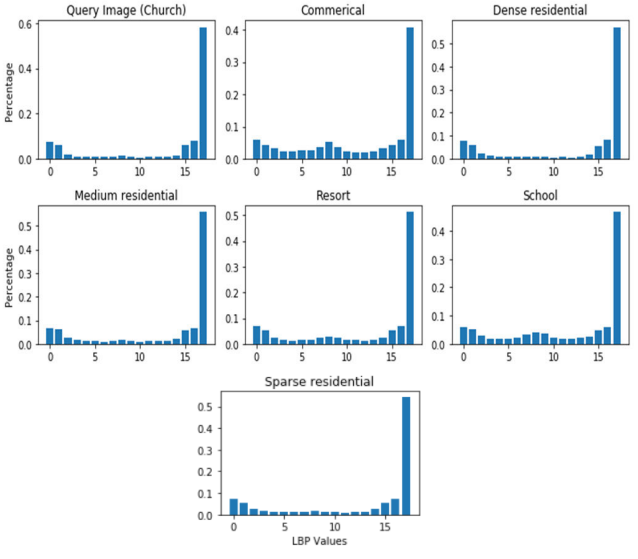


FIGURE 3. *Local Binary Pattern* (*LBP*) histograms for the 7 images of *Buildings* superclass of *AID* (from Fig.2).

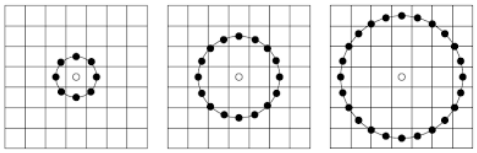


FIGURE 4. Multilevel *LBP* Operator showing different sampling pixels and radius values [84].

In equation (2), i_c is the grey level intensity of the central pixel per patch and the grey level value at position n. The result of $s(i_n\text{-}i_c)$ is 1 if $(i_n\text{-}i_c) \geq 0$ and 0 otherwise [85]. The main limitation of the classic *LBP* operator is the small neighbourhood considered (only 3×3), which could not cover the most prevalent features in an image [87]. To address this limitation, further improvement on *LBP* makes it possible to accommodate textures at different levels by using a neighbourhood of varying sizes [85], [87]. *LBP* has been applied with CNNs for remote sensing scene classification to improve image classification based on texture [90]. However, the core idea of this work, which is using texture to group classes into superclasses, has yet to be exploited.

In the improved version of *LBP*, a neighbourhood is defined by the number of sampling pixels. Depending on the size of the sampling pixels, a circle of evenly spaced points is formed around the central pixel, which is used to threshold the surrounding pixels [88]. Ojala et al. [82] found out that *LBP* patterns representing the key distinguishing features of images, such as edges, curves, line-ends, etc., occur more frequently than other patterns. Based on this observation, uniform patterns were adopted to reduce the feature vector's length and improve the method's statistical robustness [88], [91]. Fig. 4 shows an example of different radius values and sampling pixels of an *LBP* operation using 8 sampling pixels and 1 radius, 16, 2 and 24, 3 values, respectively.

The *Euclidean Distance* (*ED*) is an effective metric widely used for image retrieval due to its efficiency. The *ED* between two vectors is obtained by taking the square root of the sum of the squared absolute difference between the vectors. Because the *LPB* codes are in a one-dimensional vector, the Euclidean distance becomes the metric of choice since it works very well with low-dimensional data [92].

$$E_D = \sqrt{\sum_{i=1}^{N} (H_1 - H_2)^2} \tag{3}$$

The *Earth Mover's Distance* (EMD) measures the distance between two probability distributions, A and B, and it is estimated as the most negligible cost of converting one distribution into another [80], [93].

$$EMD = \frac{\sum_{i,j}^{N} d_{i,j} g_{i,j}}{\sum_{i,j}^{N} g_{i,j}} \tag{4}$$

Here, $d_{i,j}$ represents the dissimilarity between bin i and j, and $g_{i,j} \geq 0$ is the optimum flow between the two distributions such that the total cost $\sum_{i,j}^{N} d_{i,j} g_{i,j}$ is minimalised [93].

IV. METHODOLOGY

A. PROBLEM STATEMENT

Several attempts have been made to design deep learning techniques to address the issue of misclassifications due to inner-class variability and outer-class similarity by combining the local and global features of images and by improving the quality of the extracted features using content enhancement modules in some cases [2].Even so, misclassifications remain high, as mentioned by Li et al. [1]. To address the challenge of misclassifications by deep learning classifiers, we propose a technique for combining the categories of some selected land-use and land cover classification datasets into superclasses based on colour, spatial, and textural similarities. This approach reduces the need for a highly sophisticated model. It introduces an alternative to the traditional scene classification approach that has not been explored, with the potential of a wide range of applications in areas of earth observation such as agriculture, disaster management, change detection, town planning, defence, and other fields of earth observation.

B. INNER-CLASS VARIABILITY AND OUTER-CLASS SIMILARITY ASSESSMENT TECHNIQUES

For each class in a dataset, depending on the number of images per class, the *LBP* code of each image is extracted and converted to a histogram. A distance metric, *Euclidean Distance* or *Earth Mover's Distance*, is used to compare the *LBP* code histogram of each image with another. The similarity score is stored in a vector. Each class's minimum, maximum, mean, standard deviation and other statistical parameters are computed. Classes with similar spatial structure or structural functionality are grouped to form superclasses using the statistical measure as the criterion.

1. $\forall \boldsymbol{y} \in \boldsymbol{C}$: compute the colour histograms of each image and save the result in the vector $\vec{\boldsymbol{Q}}$;
2. $\forall \boldsymbol{C'} \in \vec{\boldsymbol{Q}}$: normalise the histograms;
3. $\forall \boldsymbol{q} \in \vec{\boldsymbol{Q}}$: compute the *Bhattacharyya*(1) histogram similarity value, D_H, from $\boldsymbol{q_i}$ to $\boldsymbol{q_{i+1}}$ *to* $\boldsymbol{n-1}$ and save the result to another vector $\vec{\boldsymbol{F}}$;
4. Using $\vec{\boldsymbol{F}}$, compute the *min*, *max*, *mean* and *standard deviation* of all the values in $\vec{\boldsymbol{F}}$.

   where:
   $\boldsymbol{y}$, $\boldsymbol{C}$,$\boldsymbol{C'}$,$\vec{\boldsymbol{Q}}$, $\vec{\boldsymbol{F}}$ and $\boldsymbol{q}$ are the images in a superclass, the superclass label, image histograms, the first vector, the second vector, and the histogram similarity values, respectively.

The classes are grouped by comparing markers on their respective box plots. The same steps are repeated for image texture similarity assessment using the *local binary pattern* and structural similarity index (*SSIM*). The images are grouped into superclasses depending on their levels of similarity according to the below steps.

1. $\forall \boldsymbol{y} \in \boldsymbol{C}$: compute the *LBP* using equation (3) and save the result to the vector $\vec{\boldsymbol{Q}}$;
2. $\forall \boldsymbol{C'} \in \vec{\boldsymbol{Q}}$: convert the *LBP* into histograms;

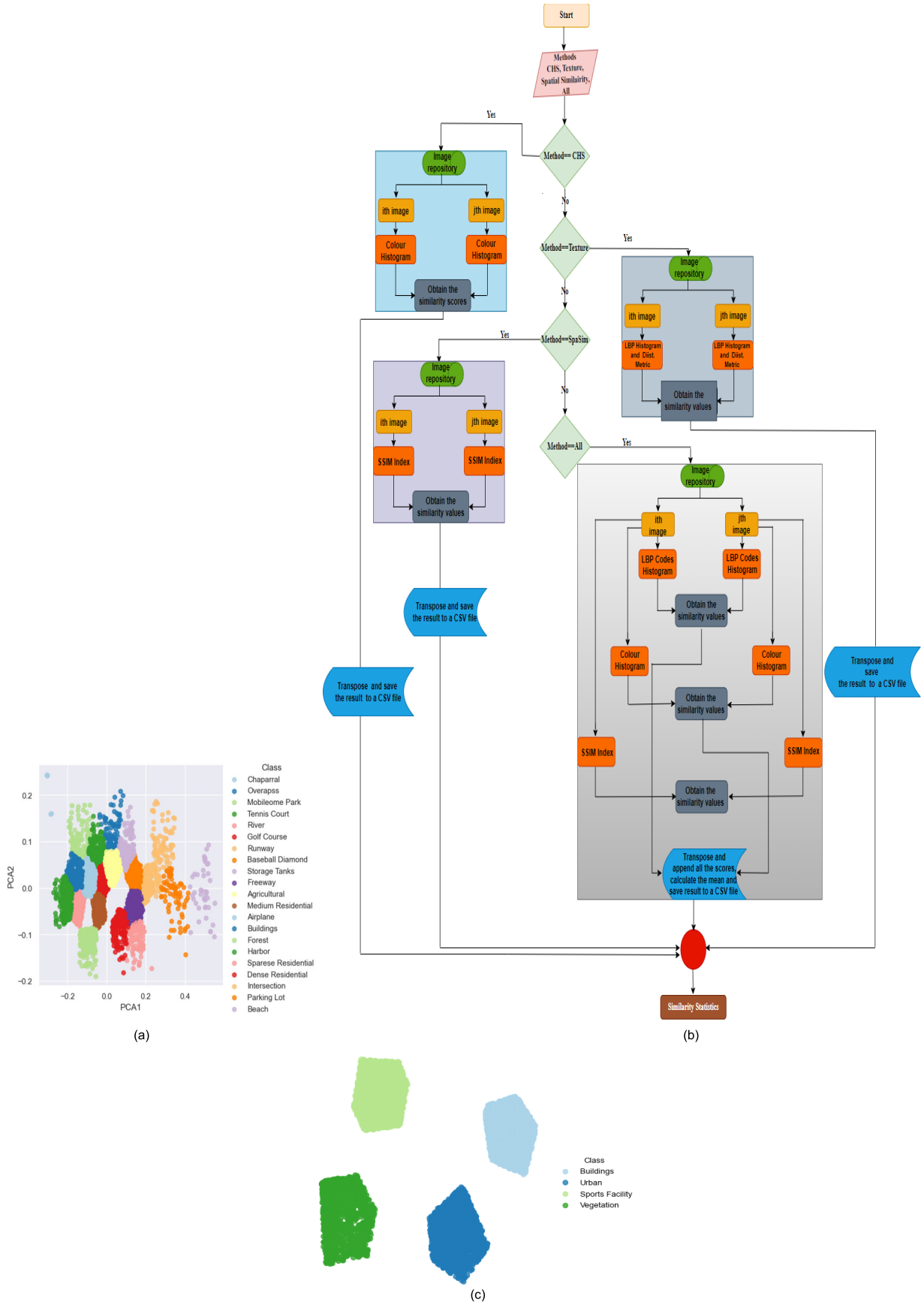


FIGURE 5. Different stages of the combined framework showing: (a) the original dataset; (b) the image similarity analysis framework; (c) the clustering stage; and (d) the fine-tuned *Xception* model.

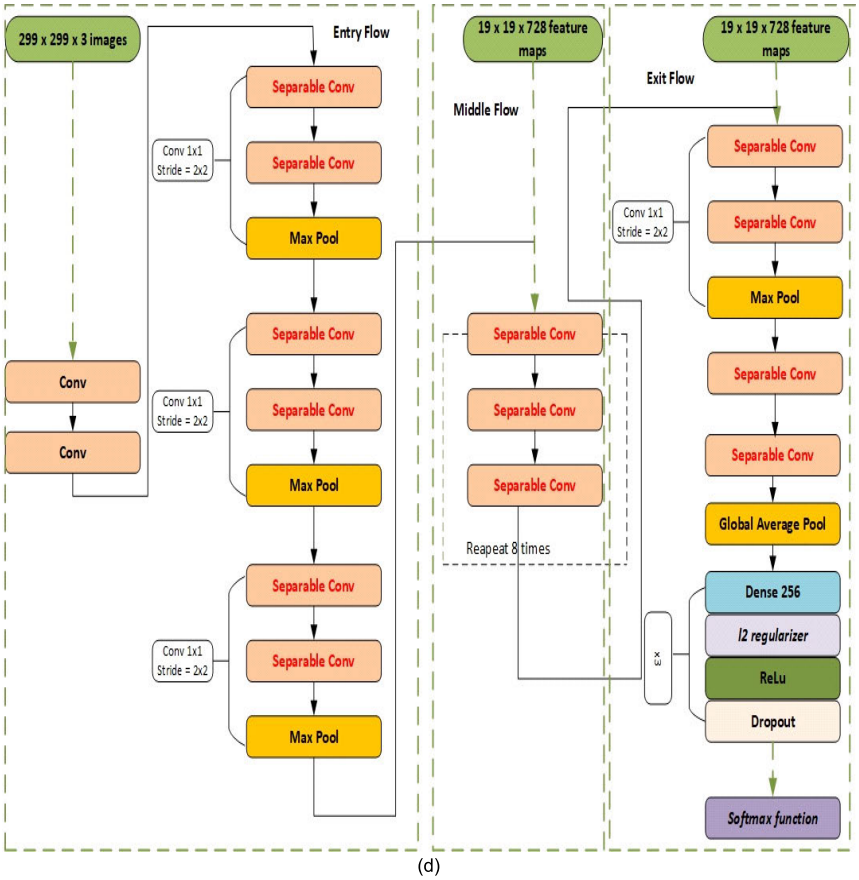


FIGURE 5. *(Continued.)* **Different stages of the combined framework showing: (a) the original dataset; (b) the image similarity analysis framework; (c) the clustering stage; and (d) the fine-tuned *Xception* model.**

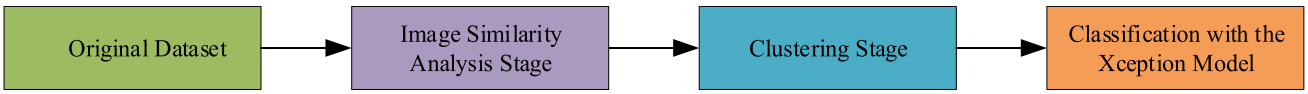


FIGURE 6. **The workflow diagram shows all the stages described in Figure 5 above.**

3. $\forall \boldsymbol{q} \in \vec{\boldsymbol{Q}}$: compute the histogram similarity, d using the *Euclidean* Distance, E_D (5) or *Earth Mover'* s Distance, *EMD* (6) from $\boldsymbol{x_i}$ to $\boldsymbol{x_{i+1}}$ *to* $\boldsymbol{n-1}$ and save the result to vector $\vec{\boldsymbol{F}}$;
4. Compute the *min*, *max*, the *mean* and *standard deviation* of all the values in $\vec{\boldsymbol{F}}$.

Where:

$\boldsymbol{y}$, $\boldsymbol{C}$,$\boldsymbol{C'}$, $\vec{\boldsymbol{Q}}$, $\vec{\boldsymbol{F}}$, and $\boldsymbol{q}$, are the images in a superclass, the superclass label, *LBP* codes, the first vector, the second vector, and the histogram values, respectively.

As shown in Fig. 5 and 6 above, the workflow has three data processing stages. First, the image similarity analysis framework takes input and compares them depending on the user's choice. There are four approaches for comparing the images in the framework: a comparison based on colour, texture, structural similarity, and a combination of these three methods. For the colour similarity analysis, we used the *Bhattacharyya* metric for comparing the images, considering its efficiency in measuring the similarity between a pair of statistical distributions. Each image is read, and the corresponding histogram is generated and saved in a vector. Subsequently, the histograms are compared one after the other, and the result is transposed from a row to a column vector and into a comma-separated values *(CSV)* file after each iteration.

TABLE 1. **Merged classes in each of the 4 Superclasses and the corresponding number of samples in the *UCM* dataset.**

	Class Members	#Samples
Buildings	*Buildings, Dense residential, Medium residential, Mobile home park, Sparse residential*	*400*
Sport	*Baseball diamond, Golf course, Tennis court*	*400*
Urban	*Airplane, Freeway, Intersection, Harbour, Overpass, Parking lot, Runway, Storage tank*	*800*
Vegetation	*Agriculture, Beach, Chaparral, Forest, River*	*500*

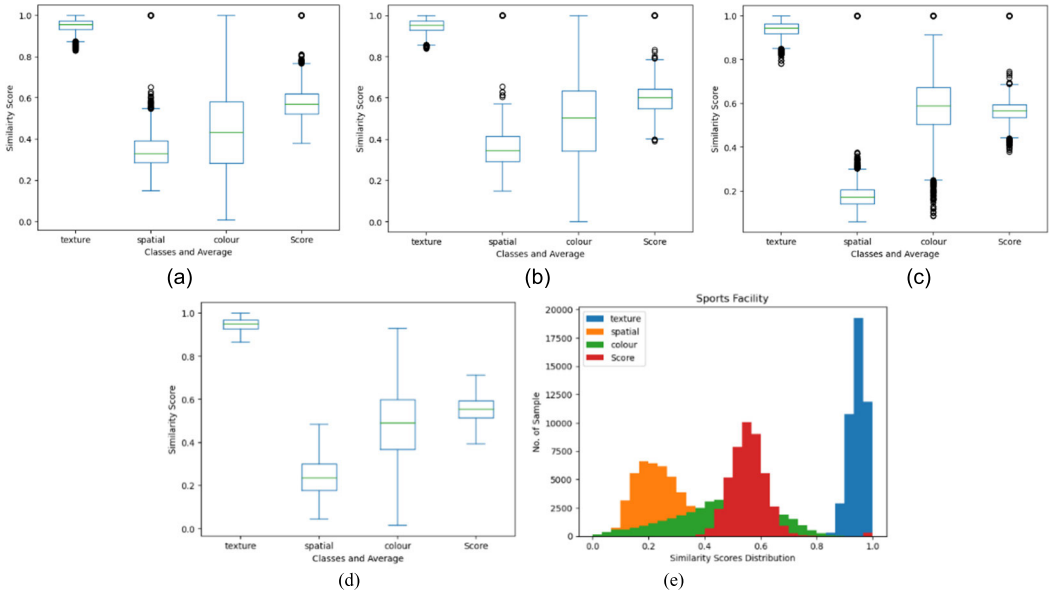


FIGURE 7. Box plots and similarity scores distribution for the sports facility superclass: (a) ***Baseball diamond***; (b) ***Golf course***; (c) ***Tennis court***; and (d) & (e) ***Sports facility***.

TABLE 2. Merged classes in each of the 7 Superclasses and the corresponding number of samples in the ***OPTIMAL-31*** dataset.

Superclass	Class Members	#Samples
Buildings	*Church, Commercial_area, Dense residential, Industrial_area, Medium residential, Mobile_home_park*	*360*
Desert	*Beach, Chaparral, Desert*	*180*
Road	*Bridge, Freeway, Intersection, Overpass, Railway, Roundabout, Runway*	*420*
Sportsfacility	*Baseball_diamond, Basket_ball_court, Golf_course, Ground_track_field,*	*240*
Urban	*Airplane, Airport, Harbor, Parking_lot*	*240*
Vegetation	*Circularfarmland, Forest, Meadow, Mountain, Rectangularfarmland*	*300*
Waterbody	*Lake, Island*	*120*

TABLE 3. Merged classes in each of the 7 Superclasses and the corresponding number of samples in the ***AID*** dataset.

Superclass	Class Members	#Samples
Buildings	*Church, Commercial, Dense residential, Medium residential, Resort, School, Sparse residential*	*2180*
Desert	*Bareland, Beach, Desert*	*1010*
Road	*Bridge, Railway, Viaduct*	*1040*
Sportsfacility	*Baseballdiamon, Park, Playground, Stadium*	*1230*
Urban	*Airport, Center, Industrial, Parking, Port, Square, Storagetanks*	*2470*
Vegetation	*Farmland, Forest, Meadow, Mountain, River*	*1650*
Waterbody	*Pond*	*420*

The mean, the standard deviation, the variance, and other relevant statistical measures are computed and saved on a separate file. Further additional similarity analysis approaches on texture and structural similarity follow the same procedure, using a combination of the Euclidean distance, local binary pattern (*LBP*) and structural similarity index (*SSIM*). However, for the texture similarity analysis, luminosity is first added to the images to sharpen them and improve the quality of their features. The *LBP* codes for each image are then extracted and converted into histograms before applying a distance metric to determine the similarity level between each pair.

Where all three features, colour, texture and structural similarity, are of interest, as in the case of this work, two images are uploaded and processed simultaneously, and the result of each similarity approach is saved into a row vector before being transposed into a column vector. The average of the three scores is then generated as the basis for comparing the images. From the average score and the scores for the other similarity comparisons, box and scatter plots of each result are plotted to help establish the degree of similarity between classes before combining them into clusters. Upon careful observation of the statistical markers on the plots, classes with values within specific ranges are grouped to form superclasses. The similarity result for each feature per class is appended to a new *CSV* file from which superclass box plots

are generated to compare the statistical correlation between each class and its superclass. The generated superclasses, each consisting of a range of subclasses, are then clustered further to establish the degree of inner-class similarity among them. In the final phase of the framework, the clustered classes are fed into the classification module.

V. EXPERIMENTS

A. XCEPTION MODEL AS THE BACKBONE FOR TRANSFER LEARNING

The idea of the *Xception* model [21] stemmed from the need to develop a deep separable convolutional neural network as an alternative to the regular CNNs models and the depth-wise separable convolution. The *Xception* model can be viewed as the culmination of improvements of some of the most predominantly used transfer learning models, such as *Inception* V1-4 [22], [94], [95], [96], and *VGGNet* [20]. Despite having the same parameters as *the Inception V3*, this model has proven more reliable for image classification due to its efficient use of parameters [21]. The choice of the model for this research is informed by its simple yet efficient structure and effective use of the model training parameters.

B. DATASETS

The UC Merced dataset (*UCM*) [97] is an *HRRS* image scene dataset of 21 classes containing 2100 images. Each category contains 100 RGB images with a size of 256×256 pixels and a uniform spatial resolution of 0.3 meters. Classifying images in the *UCM* dataset is challenging due to the high inner-class variability and outer-class similarity among the images in some categories. For example, the images in class *River* and those in class *Forest* have a high degree of similarity, which could cause misclassifications by the deep learning classifiers. The superclasses generated from the categories in this dataset are given in Table 1.

OPTIMAL-31 [98] is a scene classification dataset of 31 classes. Each category in this dataset contains the same number of images, 60, of size 256×256 pixels, totalling 1860 images from the categories given in Table 2. Like the *AID* dataset, images in *OPTIMAL-31* were also sourced from Google Earth Images.

The *Aerial Image Dataset* (*AID*) [99] is one of the most extensive scene classification datasets. It contains 30 classes of between 200 and 420 images per category (Table 3). The images in *AID* are of a uniform dimension of 600×600 pixels and spatial resolution ranging from 8 to 0.5 meters. There are 10,000 images from the categories given in Table 3. Unlike the *UCM* dataset, *AID* is a multi-source dataset gathered from Google Earth Images and other sources and, thus, more challenging to work on than *UCM* images.

NWPU-RESISC45 is a single-label dataset of 45 classes comprising 31,500 images of different observation scenes. The dataset was developed by the Northwestern Polytechnic University [15]. Each of the 45 dataset classes consists of 700 images of size 256×256 in RGB format and a spatial resolution ranging from 30 to 0.2m. All the images were acquired through Google Earth imagery across more than 100 countries and regions [77].

C. TRAINING STRATEGY

1) DATA AUGMENTATION

Considering the amount of data needed to train a deep learning model from scratch, which is one of the primary reasons for opting for the transfer learning approach, we decided to incorporate some data augmentation techniques to boost the size of the datasets. Eight more copies of each training image were generated by applying *vertical flip, horizontal flip, rotation, width shift, height shift, zooming, and shear* operations, generating 11,760 more training samples for the *UCM* and 56,000 for the *AID* and 10,416 for *OPTIMAL-31*.

2) TRAINING PARAMETERS

To fine-tune the *Xception* model for our task, the top layers of the model were cut off, leaving only the bottom layers from the input layer to the last convolutional layers of the model. This base model is used as the backbone for feature extraction; six more *layers,* besides the classification layer, were stacked atop the backbone. The first layer added is the *GlobalAveragePooling layer,* which takes the average of each feature map, followed by a *dense layer* with *256 filters*, an *l2 regulariser* with a *0.001* regularisation value and a *ReLU* activation function. In addition to the *l2*, to make the model more efficient and reduce the risk of *overfitting*, we added a *dropout layer* with a *0.25 dropout* value. The next block contains the same *layers*, filters, and *l2 regulariser* as in the previous layer. In the last layer (before the classification layer), the number of filters was increased to *512* to extract more discriminative features. The dropout value was also doubled to mitigate the possibility of overfitting. The *Adam* optimiser with a fixed learning rate of 10e-6 and a decay value of 10e-6 was used for the model training parameters. Since this is a multiclass classification task, the categorical cross-entropy loss function was employed to update the model's weights. We used the *softmax* function for the eventual classification of the images. For a class s_i, the *Softmax* function is calculated using the expression:

$$f\,(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}},$$

where s_j is the vector of the scores predicted by the model for each class in $\boldsymbol{C}$. The cost function for the model, using *categorical cross-entropy* loss, is computed as:

$$Cost\ function = -\sum_i^C t_i log\,(f\,(s)_i) + \frac{\lambda}{2} * \Sigma\,\|\boldsymbol{w}\|^{\mathbf{2}},$$

where $\boldsymbol{C}$is the number of classes, t_i is the index of the target ground truth in vector t,$f\,(s)_i$ is the inferred value for the class s_i, λ is the regularisation rate, and $\boldsymbol{w}$ is the model's weights.

Earlystopping was used to terminate the training if the metric of interest (validation loss) did not improve for 100 epochs. To split the data into test, train and validation

TABLE 4. Comparison between our approach and some state-of-the-arts on the *UCM* dataset.

Superclass	OA (%)
VGG9-Net+SVM [32]	94.70
CNN-FT-Full [100]	96.67
DCV [101]	90.50±1.20
DFB-LULC [102]	92.7±0.80
CNN-DA [52]	82.29
VGG-DV-16 [99]	95.20
GoogleNet [99]	94.30
CNN+FV [14]	98.60
AlexNet_MSCP [103]	97.29±0.63
VGG_D16+MSCP [103]	98.36±0.58
InceptionV3 [104]	91.00
VGGNet(VGG19) [104]	94.30
VGG19+Kmeans[32]	94.97
CNN+Genetic Algo. [105]	94.43±0.71
***Xception*-4Scs**	**99.05**
***Xception*-10Scs**	**98.64**
***Xception*-21 Classes**	**96.90**

TABLE 5. Comparison between our approach and some state-of-the-arts on the *OPTIMAL-31* dataset.

Superclass	OA (%)
ArchNetVGG16 [98]	87.45±0.45
Fine-tuned AlexNet [98]	82.57±0.12
Fine-tuned GoogleNet [98]	87.15±0.45
Fine-tuned VGG16 [98]	92.70±0.35
VGG-DV-16 [1]	89.12±0.35
KFBNet(VGGNet16) [1]	95.12±0.13
KFBNet(DenseNet121) [1]	95.60±0.63
GBNet+global feature [106]	93.28±0.27
CNN+DM Arch [107]	96.24±1.10
EfficientNet-B3 [108]	92.33
***Xception*-7Scs**	**97.31**
***Xception*-17Scs**	**97.04**
***Xception*-30 Classes**	**92.20**

TABLE 6. Comparison between our approach and some state-of-the-arts on the *AID* dataset.

Superclass	OA (%)
InceptionV3 [104]	94.10
VGGNet (VGG19) [104]	88.90
AID Benchmark (VGG) [102]	89.64±0.36
CNN+Genetic Algo. [105]	91.20±0.49
Search for CNN Arch [77]	96.10±0.18
CNN+LBP [90]	95.73±0.16
ArchNet+VGG16 [98]	93.10±0.55
VGG19+Kmeans [32]	95.29
GBNet+global feature [106]	95.45±0.21
***Xception*-7Scs**	**97.25**
***Xception*-17Scs**	**97.90**
***Xception*-30 Classes**	**95.75**

sets, we set aside 80% of the dataset for training and 20% for testing. Out of the training set, 15% was used for validation. Overall, the model is trained for 350 epochs. All experiments were run on *Google Collaboratory* with a high accelerator set as *GPU* and runtime shape as *High-RAM*. The models are modified and fine-tuned using *TensorFlow* [113] and *Keras* environment [114].

TABLE 7. Comparison between our approach and some state-of-the-arts on the *NWPU-RESISC45* dataset.

Superclass	OA (%)
MLFFN [109]	92.45±0.20
VGGNet-16 [15]	90.36±0.36
MLFCNetXt50 [110]	94.76±0.08
MLFCNet50 [110]	94.32±0.04
MLFC-VGG-1650 [110]	93.02±0.09
MILRDA [111]	92.87±0.26
SLGE CNN [77]	96.56±0.13
MRHNet-50 [112]	96.33
MRHNet-101 [112]	96.38
***Xception*-10Scs**	**98.00**
***Xception*-18Scs**	**97.20**
***Xception*-45 Classes**	**95.62**

TABLE 8. Ablation experiments conducted on the *UCM* dataset.

Model	Accuracy(%)
Baseline	70.25
***Xception* + 1 block**	95.71
***Xception* + 2 blocks**	98.00

TABLE 9. Ablation experiments conducted on the *OPTIMAL-31* dataset.

Model	Accuracy(%)
Baseline	70.25
***Xception* + 1 block**	85.71
***Xception* + 2 blocks**	88.00

TABLE 10. Ablation experiments conducted on the *AID* dataset.

Model	Accuracy(%)
Baseline	72.20
***Xception* + 1 block**	89.30
***Xception* + 2 blocks**	90.59

TABLE 11. Ablation experiments conducted on the *NWPU-RESISC45* dataset.

Model	Accuracy(%)
Baseline	73.99
***Xception* + 1 block**	85.71
***Xception* + 2 blocks**	88.00

D. ABLATION EXPERIMENT

To demonstrate the effectiveness of the proposed approach, we carried out ablation experiments on the four datasets. The experiments were performed to analyse the impact of the added blocks on the proposed model. The entire experiments can be divided into two parts.

1) Verify the effect of removing the added blocks on the performance of the proposed model. This was achieved by removing all the added blocks and retaining only the Xception feature extraction module and the classification layer. The removal of the added blocks shows that the feature extraction part of the proposed model cannot, on its own, achieve the desired result.

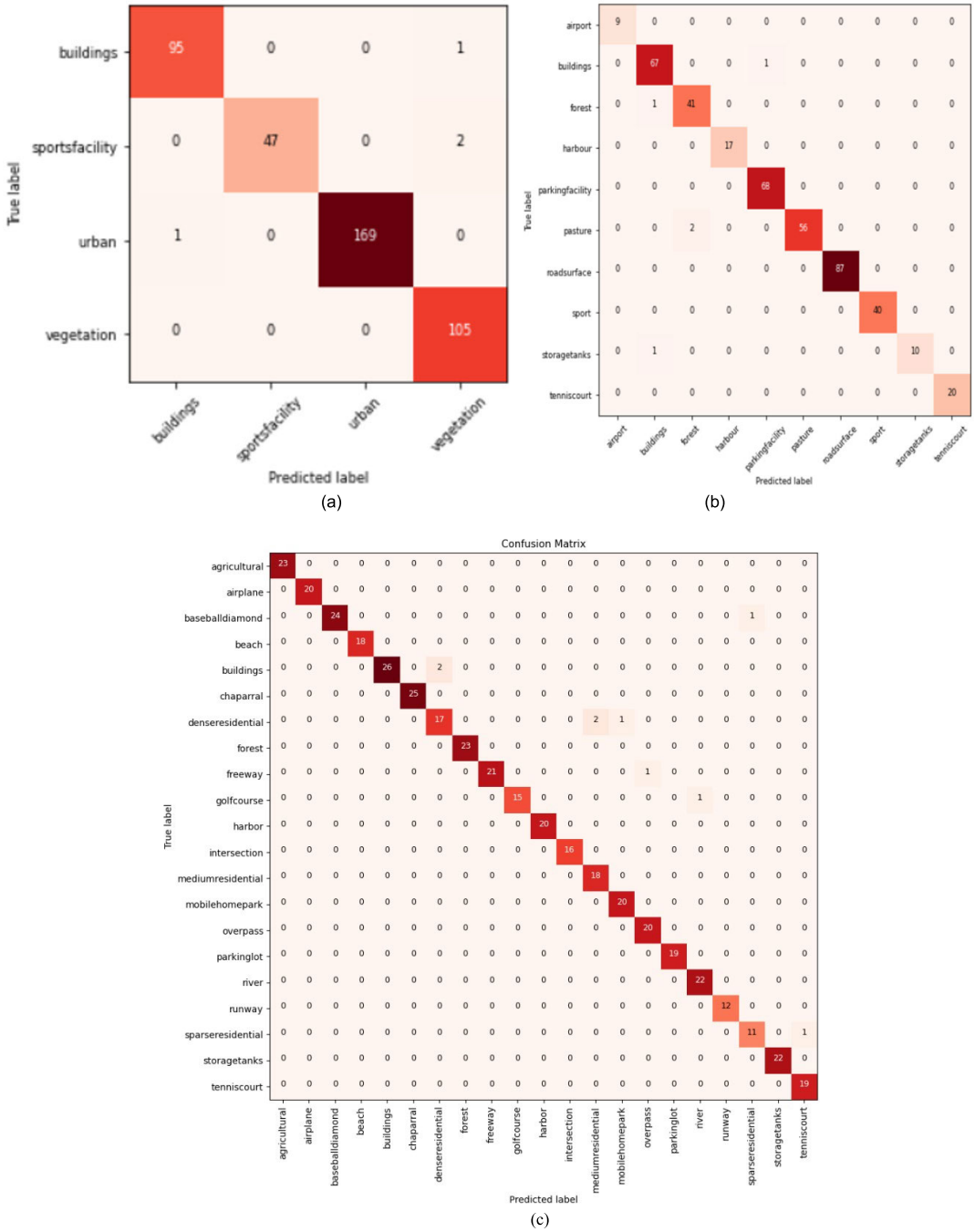


FIGURE 8. Confusion matrices for (a) 4 superclasses; (b) 10 superclasses; and (c) 21 classes of the *UCM* dataset.

2) Three blocks have been added to the baseline model to form the proposed model. Each block consists of a dense layer, an *l2 regularisation*, a *ReLU* and a dropout layer. In the second part of the ablation experiment, one block was removed, leaving the baseline model and two other blocks. Finally, two blocks were removed to assess the effectiveness of a single block of four layers on the proposed model.

E. EVALUATION METRICS

To evaluate the performance of the modified architecture, we employ the following metrics: overall accuracy, balanced accuracy, confusion matrix, F1 Score, Kappa Score, precision, recall, and ROC Curve. We considered balanced accuracy (BA) mainly because of randomness in how the test, train and validation sets are split.

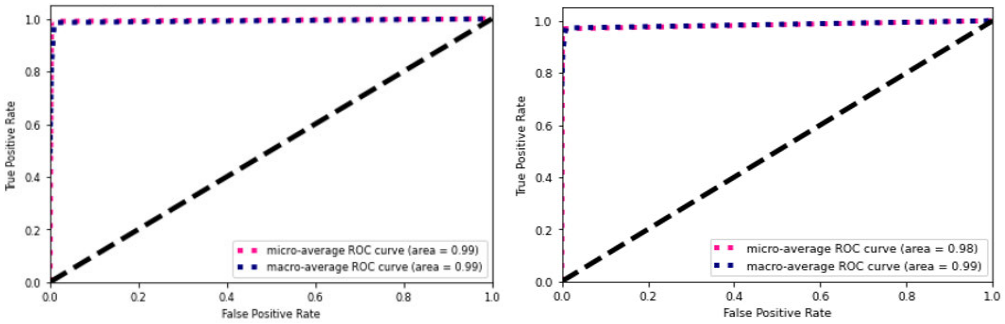


FIGURE 9. ***ROC*** **Curves for the (a) 4 superclasses and (b) 21 classes of the** ***UCM dataset.***

1) OVERALL ACCURACY (OA)

The overall accuracy is obtained by dividing the total number of correctly predicted test samples by the total number of samples in each category [115].

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

2) BALANCED ACCURACY (BA)

the average of the specificity and sensitivity. It gives information on the probability of classifying a class correctly. The critical difference between the BA and OA is how accuracy for balanced and imbalanced data is computed. If the data is balanced, the two converge and vice versa [115].

$$Balanced\ Accuracy = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2}$$

3) CONFUSION MATRIX

This visual performance evaluation metric shows how well a model classifies the categories in each dataset class. It is easy to interpret and shows the exact number of accurate classifications and misclassifications per class in a row/column [115].

4) F1 SCORE

It uses the confusion metric to aggregate precision and recall through harmonic mean. F1-Score can be viewed as the weighted average between precision and recall and returns a value of 1 for an ideal model and 0 for the worst performance [115].

$$F1 - Score = 2 \times \left(\frac{precison \times recall}{precision + recall}\right)$$

5) KAPPA SCORE

Cohen's Kappa score measures the agreement between the predicted and the actual labels in a classification task. In a multiclass classification, Cohen's Kappa works on the same principle as Matthews Correlation Coefficient [115].

$$Kappa = \frac{c \times s - \Sigma_k^K p_k \times t_k}{s^2 - \Sigma_k^K p_k \times t_k}$$

6) PRECISION AND RECALL

Precision is obtained by dividing the number of True Positives (TP) by all positively classified labels (TP and FP). The Recall (sensitivity or TPR) measures the proportion of actual positives categorised correctly by the model. The values of other metrics, such as accuracy, balanced accuracy and F1-Score, are obtained from these essential metrics [115].

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

7) ROC CURVE

The *Receiver Operating Characteristic* (*ROC*) is a probabilistic curve that illustrates the performance of a classification model at all classification thresholds (TPR vs. FPR at different classification thresholds). The *ROC* curve gives a summary of a classifier's performance. To visualise a model's performance, the area under the *ROC* curve (*AUC*) shows the degree of accuracy with which the model can separate among classes in a given task. In addition, the *AUC* is used to compare the performances of different classification models [116].

F. EXPERIMENTAL RESULTS

In addition to the *model's overall accuracy,* we extracted a few more performance evaluation metrics, such as *balanced accuracy*, to address the imbalance in the datasets (*UCM, AID, OPTIMAL-31,* and NWPU-RESISC45). Tables 4, 5 and 6 show our approach's comparisons with the state-of-art OA models. The *precision, recall, F1-Score, and Kappa scores* have all been calculated (see Tables 12, 13, 14, and 15). Table 16 gives the *AUC* for the three experiments under each dataset. The *UCM* has *AUC* values of *0.9913*, *0.9904,* and *0.9852* for the four superclasses, 10 superclasses and 21 classes, respectively. *AID* has *AUC* values of *0.9841, 0.9827,* and *0.9767* for the 7 superclasses, 17 superclasses, and 30 categories, respectively. *OPTIMAL-31* has *0.9846, 0.9821,* and *0.9591*, respectively. It is noteworthy that some of these *state-of-the-art* have different test-train ratios. For the *UCM* dataset (Table 4), the comparison shows that our approach of combining the classes into superclasses has led to desirable results, outperforming

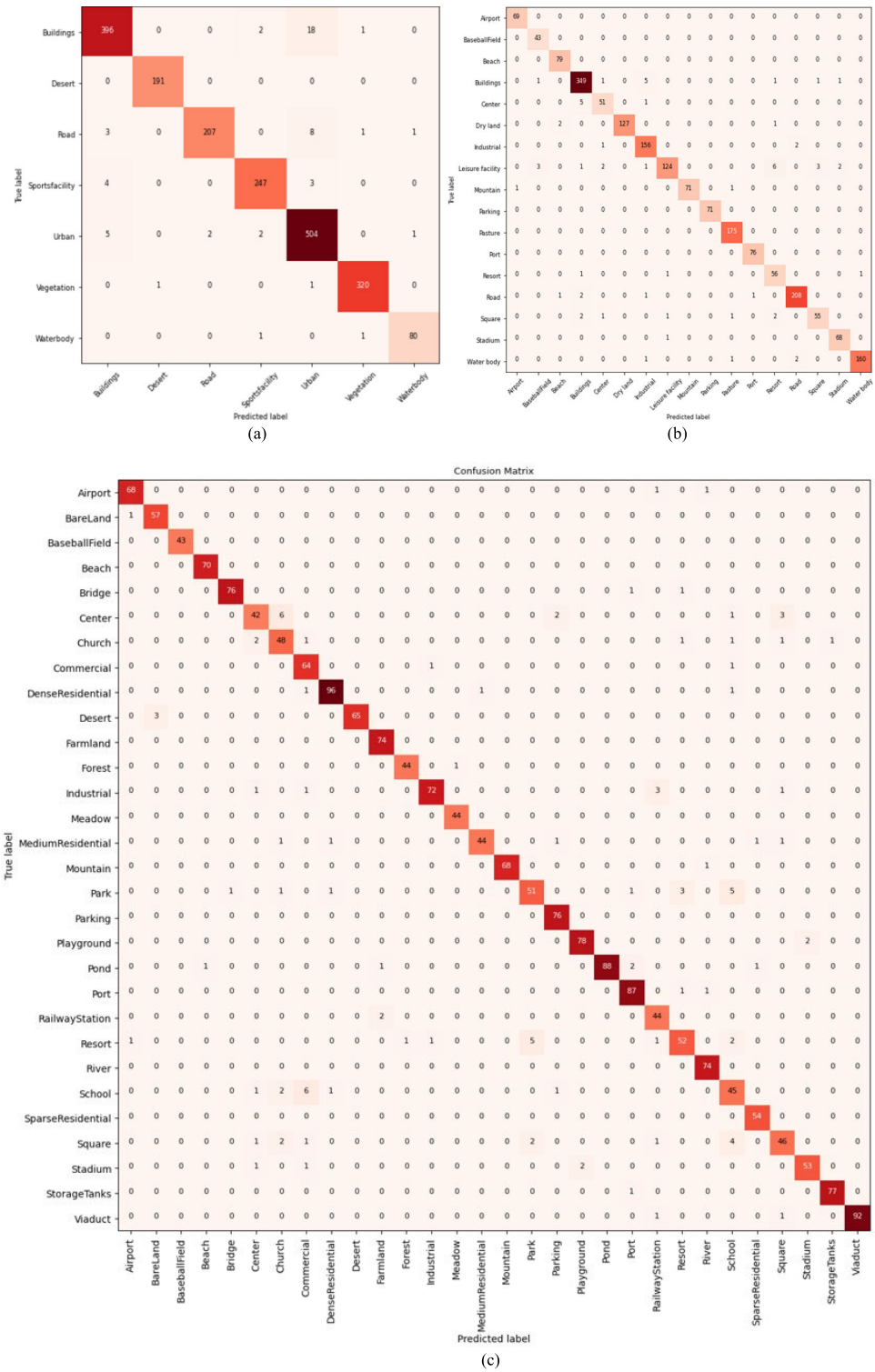


FIGURE 10. **Confusion matrices for (a) 7 superclasses; (b) 17 superclasses; and (c) 30 classes of the *AID* dataset.**

most of the state-of-the-art methods. The model performance improvement is due to the careful clustering of the classes into superclasses based on the similarity scores produced by the proposed image similarity analysis framework. Some state-of-the-art shown in the table contain feature fusion strategies that involve fusing the local and global image features to enhance models' classification ability. At the same time, our approach is simple yet highly efficient. The

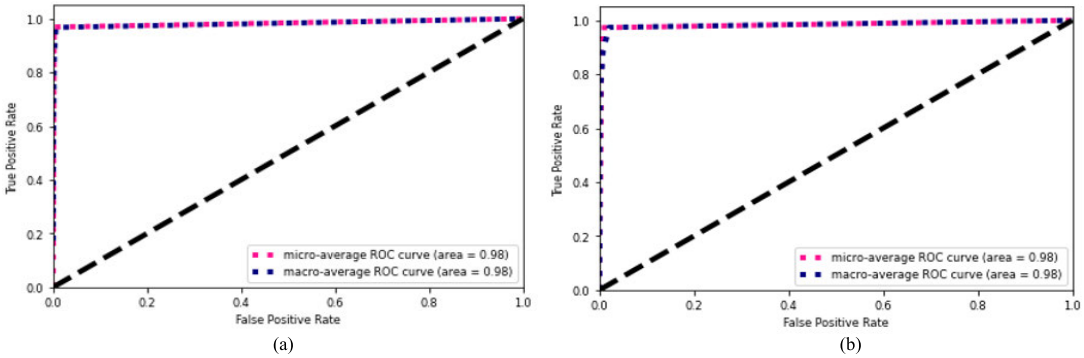


FIGURE 11. ***ROC*** **Curves for the: (a) 7 superclasses; (b) 30 classes of dataset *AID* dataset.**

architecture of our approach includes only the backbone network (the *Xception* model) and a few additional layers incorporated to stabilise and enable the model to extract more features and mitigate the risk of overfitting.

As can be observed from Table 7, the model's performance decreases as the number of classes expands towards the original number of the dataset's classes. This reduction in the model accuracy is due to the confusion induced by inner-class variability and outer-class similarity. Even in the 21 classes of the dataset, the model performance is quite competitive compared to the more sophisticated approaches used in some state-of-the-art. Our technique gives the best result and is arguably the most efficient for the purely fine-tuned models shown in Table 4. The trend of model performance deterioration is also apparent in Table 8, where the *AUC* for the 4 superclasses is 99.13% against 98.52% for the 21 classes of the dataset. The same performance pattern can be seen in Fig. 9(a) and (b), where the *ROC* curves values for the 4 superclasses and the 21 dataset classes differ by 1%. It is essential to mention that for most of the monitored performance metrics in the case of this dataset, the variations between the model's performance on the 4 superclasses and the 10 superclasses are negligible. This can be attributed to the size of the dataset and the close similarity in how the superclasses in the two experiments were formed.

The number of misclassifications due to similarities and variabilities between and among the classes of the dataset rises with the expansion of the classes, as shown by the confusion matrices in Fig. 8(a), 8(b) and 8(c). The number jumped from just 4 for the 4 superclasses to 9 for the 21 classes. The issue of misclassification due to inner-class variability and outer-class similarity can be reduced by carefully and logically clustering the overlapping classes of *HRSS* images into superclasses. This will result in more classification accuracy and increased efficiency.

For the *AID*, the performance comparison of our approach with the state-of-the-art models is given in Table 6. Compared to the state-of-the-art, our method gives the best result in the 7 and the 17 superclasses. It also shows competitive performance in the result for classifying the 30 classes of the dataset. Similarly, from Table 8, the performance of our model for each of the three experiments on *AID* can be observed. For all the metrics shown in the table, the proposed technique performs best on the 7 superclasses, followed by the 17 superclasses and the 30 classes of the dataset. An observable correlation among the metrics' values for all the datasets shows the model is stable. Like the pattern of performance shown in the three experiments with the *UCM* dataset, the model's performance also reduced as the number of classes moved towards the dataset's original size. In Table 10, the *AUC* also shows a similar trend of performance reduction with the increase in the number of classes. As the classes increase and more categories become independent, the tendency for the classifier to misclassify the images due to inner-class variability and outer-class similarity increases. Our approach has proven to be very effective for classifying images by grouping different images into superclasses based on the selected image properties. The confusion matrices, Fig. 10(a), (b) and (c), for the three experiments also corroborate the results in tables and figures. For the first experiment on the 7 superclasses, there are 55 misclassifications, with the *Buildings* and the *Urban* classes accounting for nearly 50% of these misclassifications. This is due to the intricate structural similarities between the superclass and the images of the class *Centre* in the *Urban* superclass. For the second experiment on the 17 superclasses, the model records 62 misclassifications, which is 7 more than in the previous experiment. The last experiment on the original classes of the dataset resulted in 105 misclassifications, which is almost 100% increase in misclassifications compared to the number in the first and the second experiments.

OPTIMAL-31 is a small dataset compared to the *UCM* and the *AID* datasets, and research on this dataset is less than on the *UCM* dataset. This dataset differs from the other two datasets (UCM and AID) because it has more categories but fewer samples per class. The performance comparison of our model and the state-of-the-art on this dataset is given in Table 5. Compared to *KBFNet*, which contains three branches and an inference aggregation module, our approach performs better in the first experiment on the 7 superclasses and the

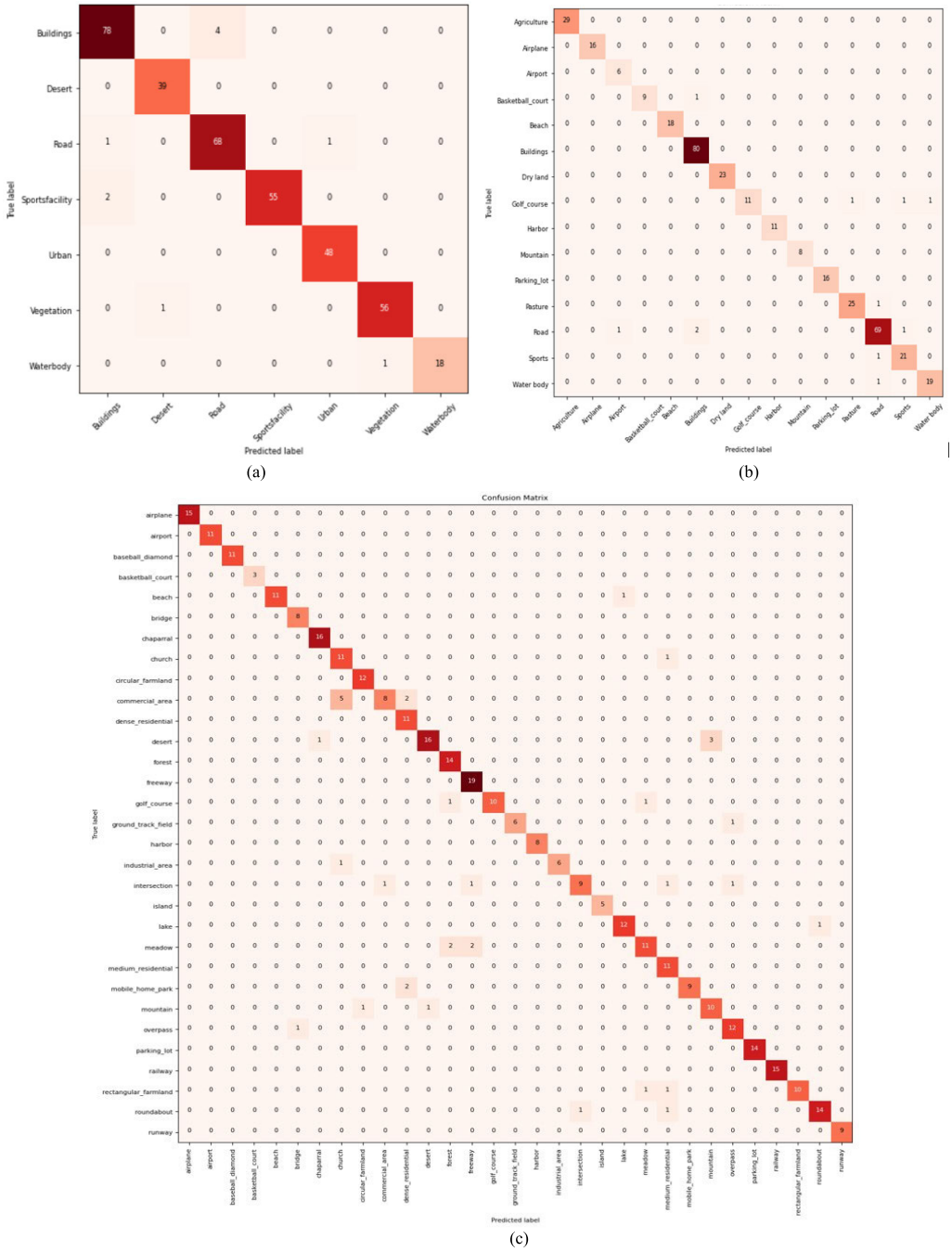


FIGURE 12. Confusion matrices for (a) 7 superclasses; (b) 15 superclasses; and (c) 31 classes of the *OPTIMAL-31* dataset.

TABLE 12. Performance evaluation metrics values for the three experiments on the *UCM* dataset.

Experiment	*OA*	*Precision*	*Recall*	*F1-Score*	*BA*	*Kappa*
4 Superclasses	99.05	99.04	98.57	98.81	98.57	98.65
10 Superclasses	98.64	99.10	98.36	98.72	98.42	98.24
21 Classes	96.90	97.13	97.19	97.16	97.18	96.74

third experiment on the 31 classes. The same performance pattern is also demonstrated by our approach in terms of overall accuracy, precision, recall, F1-Score, and Kappa, as can be observed in Table 9. For all the metrics, the

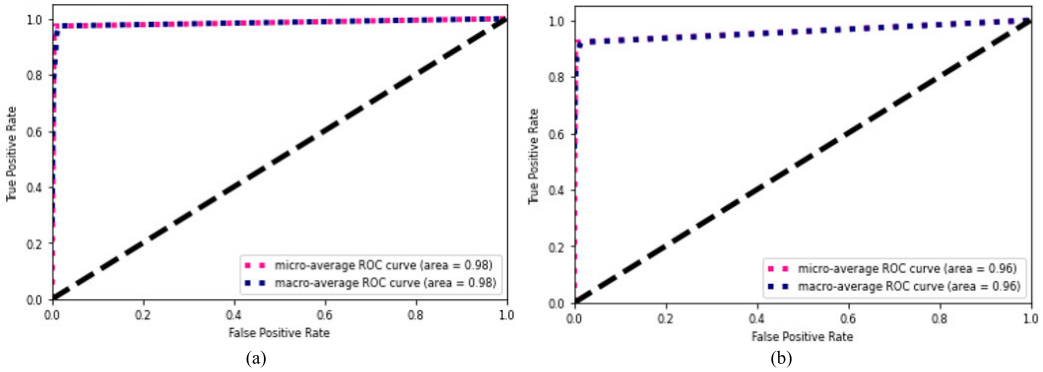


FIGURE 13. *ROC* curves for (a) 7 superclasses and (b) 31 classes of the *OPTIMAL-31* dataset.

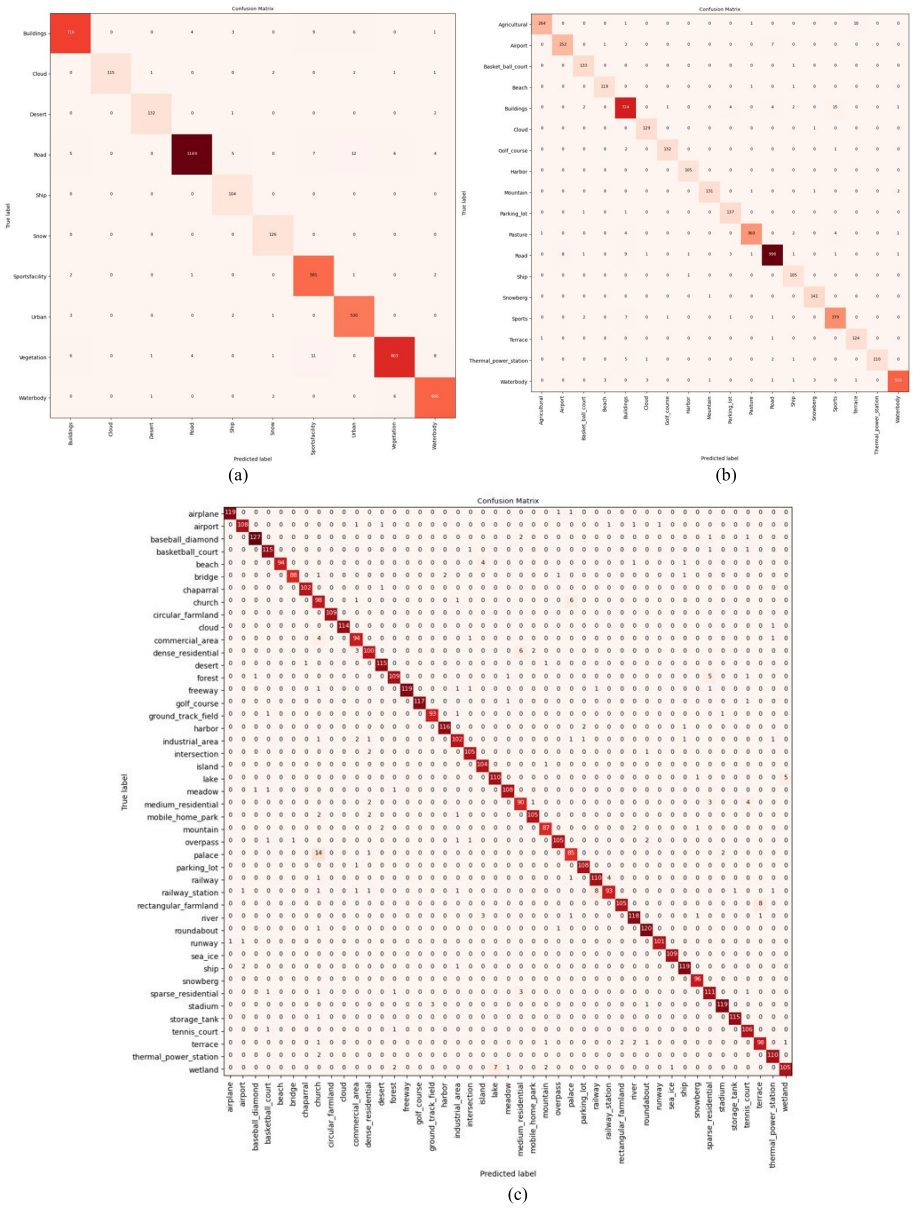
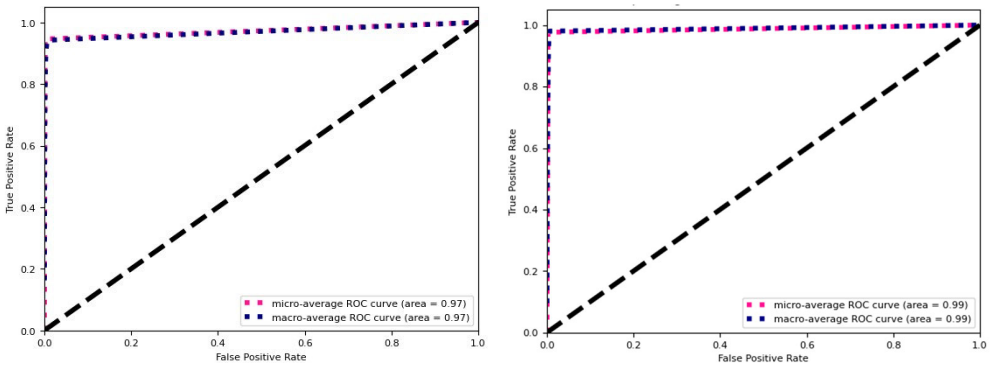


FIGURE 14. Confusion matrices for (a) 10 superclasses; (b) 18 superclasses; and (c) 45 classes of the NPUW-RESIC45 dataset.



(a) (b)

FIGURE 15. *ROC* curves for (a) 10 superclasses and (b) 45 classes of the *NWPU-RESISC45* dataset.

TABLE 13. Performance evaluation metrics values for the three experiments On the *AID* dataset.

Experiment	*OA*	*Precision*	*Recall*	*F1-Score*	*BA*	*Kappa*
7 Superclasses	97.25	97.80	97.33	97.56	97.33	96.66
17 Superclasses	96.90	96.17	96.73	96.45	96.73	96.61
30 Classes	95.75	95.52	95.48	95.50	95.48	95.60

TABLE 14. Performance evaluation metrics values for the three experiments on the *OPTIMAL-31* dataset.

Experiment	*OA*	*Precision*	*Recall*	*F1-Score*	*BA*	*Kappa*
7 Superclasses	97.31	97.78	97.39	97.58	97.39	96.80
15 Superclasses	97.04	97.35	96.64	97.00	96.64	96.65
31 Classes	92.20	92.46	92.08	92.27	92.08	91.93

TABLE 15. Performance evaluation metrics values for the three experiments on the *NWPU-RESISC45* dataset.

Experiment	*OA*	*Precision*	*Recall*	*F1-Score*	*BA*	*Kappa*
10 Superclasses	98.00	97.00	97.83	97.31	98.03	97.30
18 Superclasses	97.20	96.80	97.55	97.17	97.60	96.90
45 Classes	95.62	95.62	95.60	95.62	95.60	95.51

TABLE 16. *AUC* values for the three datasets for each of the three experiment.

Dataset	*AUC*		
	4 Superclasses	10 Superclasses	21 classes
UCM	0.9913	0.9904	0.9852
	7 superclasses	17 superclasses	30 classes
AID	0.9841	0.9827	0.9767
	7 superclasses	15 superclasses	31 classes
OPTIMAL-31	0.9846	0.9821	0.9591
	10 superclasses	18 superclasses	45 classes
NWPU-RESIS45	0.9888	0.9846	0.9808

performance is affected by the expansion of the classes. Similarly, the *ROC* curves in Fig. 13(a) and 13(b) also show that the *AUC* decreases with the number of classes. This demonstrates that inner-class variability and outer-class similarity could degrade the model's performance by causing to confuse one class with another. The *AUC* values for *OPTIMAL-31* also decline with the expansion of the classes, as given in Table 10. The reduction in the model's performance across all the experiments demonstrates the effect of inner-class variability and outer-class similarity on the deep-learning classifiers. Consequently, where, for example, the goal is to count the number of buildings instead of their type and detect vegetation regardless of its make, this approach could be more effective than the traditional method of independently classifying each class.

NWPU-RESISC45 is one of the most extensive datasets of scene classification images. It has more categories and images compared to the other datasets used in this experiment. The proposed approach shows the same performance pattern for the superclasses and the 45 classes as in the

other three datasets. The model classified the 10 superclasses with higher accuracy compared to the 18 superclasses and the 45 classes. As shown in Fig. 14(a), (b) and (c), the composition of the superclasses reduces the number of misclassifications and improves the model's overall performance. For the 10 superclasses, 79 misclassifications have been recorded, while more than 120 misclassifications occurred after expanding the classes back to the dataset's original size. In terms of other metrics, Table 7 gives.

VI. CONCLUSION

In this article, we propose a framework for combining classes of remote sensing scene classification datasets into superclasses based on textural, spatial, and colour similarities to improve the model classification performance. For the textural and spatial similarity analysis, we used the *Local Binary Pattern* in combination with the *Euclidean Distance*, *Earth Mover's Distance* and *Bhattacharyya* for image histogram similarity assessment and the *Structural Similarity Index* (*SSIM*) for spatial similarity comparison. We also present a fine-tuned model based on the *Xception* model, an improved version of successive transfer learning techniques. The *Xception* has not received its fair share of attention from researchers despite its simple yet highly efficient architecture. We demonstrate that a systematic clustering based on some image properties, such as texture, colour, and other spatial properties, can potentially reduce the risk of misclassifications (due to inner-class variability or outer-class similarity) by deep learning classifiers. The experiment results showed that texture plays a critical role in determining the most intrinsic features of an image than colour and structural similarity. This approach can be considered a proof of concept for adoption for use in object detection, change detection, and scene classification, where objects of interest share some features. More so, it has the potential to be applied in urban mapping, settlement identification and risk analysis, particularly in areas where certain facilities should not be situated in proximity due to risk hazards. In the future, we plan to automate the image clustering process and include a decision module to merge classes into superclasses based on the similarity scores. For the fine-tuned model, we plan to add a few more branches that would take input from some of the model layers and pass it into a content enhancement module for more accurate classification.

REFERENCES

[1] F. Li, R. Feng, W. Han, and L. Wang, "High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8077–8092, Nov. 2020, doi: 10.1109/TGRS.2020.2987060.

[2] Z. Li, K. Xu, J. Xie, Q. Bi, and K. Qin, "Deep multiple instance convolutional neural networks for learning robust scene representations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3685–3702, May 2020, doi: 10.1109/TGRS.2019.2960889.

[3] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[4] T. A. Balarabe and I. Jordanov, "Interpolation and context magnification framework for classification of scene images," in *Proc. Int. Conf. Comput. Graph., Visualizat., Comput. Vis. Image Process. (CGVCVIP)*, 2022, pp. 93–100.

[5] A. T. Balarabe and I. Jordanov, "A hybrid dilation approach for remote sensing scene image classification," *J. Comput. Inf. IADIS Int. J. Comput. Sci. Inf. Syst.*, vol. 17, no. 2, pp. 1–15, Jul. 2022.

[6] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 2108–2123, Apr. 2016, doi: 10.1109/TGRS.2015.2496185.

[7] G. Sheng, W. Yang, T. Xu, and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *Int. J. Remote Sens.*, vol. 33, no. 8, pp. 2395–2412, Apr. 2012, doi: 10.1080/01431161.2011.608740.

[8] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015, doi: 10.3390/rs71114680.

[9] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111322, doi: 10.1016/j.rse.2019.111322.

[10] Y. Yu and F. Liu, "A two-stream deep fusion framework for high-resolution aerial scene classification," *Comput. Intell. Neurosci.*, vol. 1, Jan. 2018, Art. no. 8639367, doi: 10.1155/2018/8639367.

[11] H. Wang and Y. Yu, "Deep feature fusion for high-resolution aerial scene classification," *Neural Process. Lett.*, vol. 51, no. 1, pp. 853–865, Feb. 2020, doi: 10.1007/s11063-019-10119-4.

[12] X. Zhang, Y. Wang, N. Zhang, D. Xu, and B. Chen, "Research on scene classification method of high-resolution remote sensing images based on RFPNet," *Appl. Sci.*, vol. 9, no. 10, p. 2028, May 2019, doi: 10.3390/app9102028.

[13] Q. Bi, K. Qin, H. Zhang, Z. Li, and K. Xu, "RADC-Net: A residual attention based convolution network for aerial scene classification," *Neurocomputing*, vol. 377, pp. 345–359, Feb. 2020, doi: 10.1016/j.neucom.2019.11.068.

[14] X. Zheng, Y. Yuan, and X. Lu, "A deep scene representation for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4799–4809, Jul. 2019, doi: 10.1109/TGRS.2019.2893115.

[15] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017, doi: 10.1109/JPROC.2017.2675998.

[16] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, 10.1038/nature14539.

[17] B. Li, W. Su, H. Wu, R. Li, W. Zhang, W. Qin, S. Zhang, and J. Wei, "Further exploring convolutional neural networks' potential for land-use scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 10, pp. 1687–1691, Oct. 2020.

[18] L. Xu, Y. Chen, J. Pan, and A. Gao, "Multi-structure joint decision-making approach for land use classification of high-resolution remote sensing images based on CNNs," *IEEE Access*, vol. 8, pp. 42848–42863, 2020, doi: 10.1109/ACCESS.2020.2976484.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.

[21] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[24] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 10691–10700.

[25] X. Yang, X. Li, Y. Ye, X. Zhang, H. Zhang, X. Huang, and B. Zhang, "Road detection via deep residual dense U-Net," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7, doi: 10.1109/IJCNN.2019.8851728.

[26] X. Zhang, X. Zhou, and M. Lin, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856. Accessed: Jun. 18, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_ShuffleNet_An_Extremely_CVPR_2018_paper.html

[27] Q. Wang, W. Huang, Z. Xiong, and X. Li, "Looking closer at the scene: Multiscale representation learning for remote sensing image scene classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1414–1428, Apr. 2022, doi: 10.1109/TNNLS.2020.3042276.

[28] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2217–2226, Jul. 2019, doi: 10.1109/JSTARS.2019.2918242.

[29] B. Petrovska, E. Zdravevski, P. Lameski, R. Corizzo, I. Štajduhar, and J. Lerga, "Deep learning for feature extraction in remote sensing: A case-study of aerial scene classification," *Sensors*, vol. 20, no. 14, p. 3906, Jul. 2020, doi: 10.3390/s20143906.

[30] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, "An efficient and lightweight convolutional neural network for remote sensing image scene classification," *Sensors*, vol. 20, no. 7, p. 1999, Apr. 2020, doi: 10.3390/s20071999.

[31] T. Ishii, E. Simo-Serra, S. Iizuka, Y. Mochizuki, A. Sugimoto, H. Ishikawa, and R. Nakamura, "Detection by classification of buildings in multispectral satellite imagery," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3344–3349, doi: 10.1109/ICPR.2016.7900150.

[32] Y. Yuan, J. Fang, X. Lu, and Y. Feng, "Remote sensing image scene classification using rearranged local features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1779–1792, Mar. 2019, doi: 10.1109/TGRS.2018.2869101.

[33] Y. Zhang, R. Zong, J. Han, H. Zheng, Q. Lou, D. Zhang, and D. Wang, "TransLand: An adversarial transfer learning approach for migratable urban land usage classification using remote sensing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 1567–1576, doi: 10.1109/BigData47090.2019.9006360.

[34] M. I. Haque and R. Basak, "Land cover change detection using GIS and remote sensing techniques: A spatio-temporal study on Tanguar Haor, Sunamganj, Bangladesh," *Egyptian J. Remote Sens. Space Sci.*, vol. 20, no. 2, pp. 251–263, Dec. 2017, doi: 10.1016/j.ejrs.2016.12.003.

[35] Q. Wang, X. Zhang, G. Chen, F. Dai, Y. Gong, and K. Zhu, "Change detection based on faster R-CNN for high-resolution remote sensing images," *Remote Sens. Lett.*, vol. 9, no. 10, pp. 923–932, Oct. 2018, doi: 10.1080/2150704x.2018.1492172.

[36] M. Hughes, S. Kaylor, and D. Hayes, "Patch-based forest change detection from Landsat time series," *Forests*, vol. 8, no. 5, p. 166, May 2017, doi: 10.3390/f8050166.

[37] Y. Katta, N. Datla, S. S. Kilaru, and T. Anuradha, "Change detection in vegetation cover using deep learning," in *Proc. Int. Conf. Commun. Electron. Syst. (ICCES)*, Jul. 2019, pp. 621–625, doi: 10.1109/ICCES45898.2019.9002581.

[38] Y. You, S. Wang, Y. Ma, G. Chen, B. Wang, M. Shen, and W. Liu, "Building detection from VHR remote sensing imagery based on the morphological building index," *Remote Sens.*, vol. 10, no. 8, p. 1287, Aug. 2018, doi: 10.3390/rs10081287.

[39] G. Pasquali, G. C. Iannelli, and F. Dell'Acqua, "Building footprint extraction from multispectral, spaceborne Earth observation datasets using a structurally optimized U-Net convolutional neural network," *Remote Sens.*, vol. 11, no. 23, p. 2803, Nov. 2019, doi: 10.3390/rs11232803.

[40] H. L. Yang, J. Yuan, D. Lunga, M. Laverdiere, A. Rose, and B. Bhaduri, "Building extraction at scale using convolutional neural network: Mapping of the United States," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 8, pp. 2600–2614, Aug. 2018, doi: 10.1109/JSTARS.2018.2835377.

[41] Z. Huang, G. Cheng, H. Wang, H. Li, L. Shi, and C. Pan, "Building extraction from multi-source remote sensing images via deep deconvolution neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 1835–1838.

[42] A. Femin and K. S. Biju, "Accurate detection of buildings from satellite images using CNN," in *Proc. 2nd Int. Conf. Electr., Commun. Comput. Eng., (ICECCE)*, Jun. 2020, pp. 12–13, doi: 10.1109/ICECCE49384.2020.9179232.

[43] J. Kang, M. Körner, Y. Wang, H. Taubenböck, and X. X. Zhu, "Building instance classification using street view images," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 44–59, Nov. 2018, doi: 10.1016/j.isprsjprs.2018.02.006.

[44] Z. Guo, X. Shao, Y. Xu, H. Miyazaki, W. Ohira, and R. Shibasaki, "Identification of village building via Google Earth images and supervised machine learning methods," *Remote Sens.*, vol. 8, no. 4, p. 271, Mar. 2016, doi: 10.3390/rs8040271.

[45] B. Ayhan, C. Kwan, B. Budavari, L. Kwan, Y. Lu, D. Perez, J. Li, D. Skarlatos, and M. Vlachos, "Vegetation detection using deep learning and conventional methods," *Remote Sens. (Basel)*, vol. 12, no. 15, p. 2502, Aug. 2020, doi: 10.3390/RS12152502.

[46] R. Nijhawan, H. Sharma, H. Sahni, and A. Batra, "A deep learning hybrid CNN framework approach for vegetation cover mapping using deep features," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Dec. 2017, pp. 192–196, doi: 10.1109/SITIS.2017.41.

[47] S. Fan, Y. Li, Z. Yan, L. Guo, and X. Wang, "Vegetation recognition based on deep learning with feature fusion," in *Proc. Int. Conf. Adv. Image Process.*, Aug. 2017, pp. 19–23, doi: 10.1145/3133264.3133276.

[48] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018, doi: 10.1109/LGRS.2018.2802944.

[49] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7209–7220, Sep. 2019, doi: 10.1109/TGRS.2019.2912301.

[50] M. Axelsson, U. Soderman, A. Berg, and T. Lithen, "Roof type classification using deep convolutional neural networks on low resolution photogrammetric point clouds from aerial imagery," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1293–1297, doi: 10.1109/ICASSP.2018.8461740.

[51] D. Rueda-Plata, D. González, A. B. Acevedo, J. C. Duque, and R. Ramos-Pollán, "Use of deep learning models in street-level images to classify one-story unreinforced masonry buildings based on roof diaphragms," *Building Environ.*, vol. 189, Feb. 2021, Art. no. 107517, doi: 10.1016/j.buildenv.2020.107517.

[52] R. Stivaktakis, G. Tsagkatakis, and P. Tsakalides, "Deep learning for multilabel land cover scene categorization using data augmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1031–1035, Jul. 2019, doi: 10.1109/LGRS.2019.2893306.

[53] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/j.neucom.2017.06.023.

[54] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020, doi: 10.1109/TEVC.2019.2916183.

[55] B. Ma, X. Li, Y. Xia, and Y. Zhang, "Autonomous deep learning: A genetic DCNN designer for image classification," *Neurocomputing*, vol. 379, pp. 152–161, Feb. 2020, doi: 10.1016/j.neucom.2019.10.007.

[56] K. Neshatpour, H. Homayoun, and A. Sasan, "ICNN: The iterative convolutional neural network," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 6, p. 119, 2019, doi: 10.1145/3355553.

[57] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[58] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015, doi: 10.1016/j.knosys.2015.01.010.

[59] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, 2016, Art. no. 9, doi: 10.1186/s40537-016-0043-6.

[60] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.

[61] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," in *Proc. 32nd Conf. Graph., Patterns Images Tuts. (SIBGRAPI-T)*, Oct. 2019, pp. 47–57, doi: 10.1109/SIBGRAPI-T.2019.00010.

[62] T. Akram, B. Laurent, S. R. Naqvi, M. M. Alex, and N. Muhammad, "A deep heterogeneous feature fusion approach for automatic land-use classification," *Inf. Sci.*, vol. 467, pp. 199–218, Oct. 2018, doi: 10.1016/j.ins.2018.07.074.

[63] C. Zhang, I. Sargent, X. Pan, H. Li, A. Gardiner, J. Hare, and P. M. Atkinson, "Joint deep learning for land cover and land use classification," *Remote Sens. Environ.*, vol. 221, pp. 173–187, Feb. 2019, doi: 10.1016/j.rse.2018.11.014.

[64] Z. Zhao, J. Li, Z. Luo, J. Li, and C. Chen, "Remote sensing image scene classification based on an enhanced attention module," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1926–1930, Nov. 2021.

[65] A. T. Balarabe and I. Jordanov, "LULC image classification with convolutional neural network," in *Proc. Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2021, pp. 5985–5988.

[66] L. Cassidy, M. Binford, J. Southworth, and G. Barnes, "Social and ecological factors and land-use land-cover diversity in two provinces in Southeast Asia," *J. Land Use Sci.*, vol. 5, no. 4, pp. 277–306, Nov. 2010, doi: 10.1080/1747423x.2010.500688.

[67] X. Liu, J. He, Y. Yao, J. Zhang, H. Liang, H. Wang, and Y. Hong, "Classifying urban land use by integrating remote sensing and social media data," *Int. J. Geograph. Inf. Sci.*, vol. 31, no. 8, pp. 1675–1696, Aug. 2017, doi: 10.1080/13658816.2017.1324976.

[68] J. E. Patino and J. C. Duque, "A review of regional science applications of satellite remote sensing in urban settings," *Comput., Environ. Urban Syst.*, vol. 37, pp. 1–17, Jan. 2013, doi: 10.1016/j.compenvurbsys.2012.06.003.

[69] H. Li, H. Gu, Y. Han, and J. Yang, "Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine," *Int. J. Remote Sens.*, vol. 31, no. 6, pp. 1453–1470, Mar. 2010, doi: 10.1080/01431160903475266.

[70] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3706–3715, Dec. 2006, doi: 10.1109/TGRS.2006.881741.

[71] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 3023–3034, May 2014, doi: 10.1109/TGRS.2013.2268736.

[72] Z. Li and L. Itti, "Saliency and gist features for target detection in satellite images," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 2017–2029, Jul. 2011, doi: 10.1109/TIP.2010.2099128.

[73] W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang, "Object detection in high-resolution remote sensing images using rotation invariant parts based model," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 74–78, Jan. 2014, doi: 10.1109/LGRS.2013.2246538.

[74] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial–spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017, doi: 10.1109/TGRS.2017.2693346.

[75] L. Zhang, L. Zhang, and V. Kumar, "Deep learning for remote sensing data," *IEEE Geosci. Remote. Sens. Mag.*, vol. 4, no. 2, pp. 22–40, 2016.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[77] C. Broni-Bediako, Y. Murata, L. H. B. Mormille, and M. Atsumi, "Searching for CNN architectures for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4701813, doi: 10.1109/TGRS.2021.3097938.

[78] M. A. Aljanabi, Z. M. Hussain, and S. F. Lu, "An entropy-histogram approach for image similarity and face recognition," *Hindawi J. Math. Problems Eng.*, vol. 2018, pp. 1–18, Jul. 2018. [Online]. Available: https://www.hindawi.com/journals/mpe/2018/9801308/

[79] S.-H. Cha and S. N. Srihari, "On measuring the distance between histograms," *Pattern Recognit.*, vol. 35, no. 6, pp. 1355–1370, Jun. 2002, doi: 10.1016/s0031-3203(01)00118-2.

[80] P. A. Marín-Reyes, J. Lorenzo-Navarro, and M. Castrillón-Santana, "Comparative study of histogram distance measures for re-identification," 2016, *arXiv:1611.08134*.

[81] N. Naik, S. Patil, and M. Joshi, "A scale adaptive tracker using hybrid color histogram matching scheme," in *Proc. 2nd Int. Conf. Emerg. Trends Eng. Technol.*, Dec. 2009, pp. 279–284, doi: 10.1109/ICETET.2009.19.

[82] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002, doi: 10.1109/TPAMI.2002.1017623.

[83] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods—A review," 2019, *arXiv:1904.06554*.

[84] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.

[85] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," *IEEE Trans. Syst. Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 765–781, Nov. 2011.

[86] M. Pietikainen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[87] X. Tan and B. Triggs, "Fusing Gabor and LBP feature sets for kernel-based face recognition," in *Analysis and Modeling of Faces and Gestures* (Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4778. Berlin, Germany: Springer, 2007, pp. 235–249, doi: 10.1007/978-3-540-75690-3_18.

[88] M. Pietikäinen and G. Zhao, *Two Decades of Local Binary Patterns—A Survey*, 2016, pp. 1–34.

[89] O. A. Vatamanu and M. Jivulescu, "Image classification using local binary pattern operators for static images," in *Proc. 8th IEEE Int. Symp. Appl. Comput. Intell. Inform. (SACI)*, no. 1996, May 2013, pp. 173–178, doi: 10.1109/SACI.2013.6608962.

[90] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, Apr. 2018, doi: 10.1016/j.isprsjprs.2018.01.023.

[91] L. Liu, P. Fieguth, G. Zhao, M. Pietikäinen, and D. Hu, "Extended local binary patterns for face recognition," *Inf. Sci.*, vols. 358–359, pp. 56–72, Sep. 2016, doi: 10.1016/j.ins.2016.04.021.

[92] F. Malik and B. Baharudin, "Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 25, no. 2, pp. 207–218, Jul. 2013, doi: 10.1016/j.jksuci.2012.11.004.

[93] Q. Zhang and R. L. Canosa, "A comparison of histogram distance metrics for content-based image retrieval," *Proc. SPIE*, vol. 9027, Mar. 2014, Art. no. 90270, doi: 10.1117/12.2042359.

[94] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing," *J. Mol. Struct.*, vol. 1134, pp. 63–66, Jan. 2015.

[95] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Oct. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[96] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[97] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2010, pp. 270–279.

[98] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019, doi: 10.1109/TGRS.2018.2864987.

[99] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

[100] B. Huang, B. Zhao, and Y. Song, "Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery," *Remote Sens. Environ.*, vol. 214, pp. 73–86, Sep. 2018, doi: 10.1016/j.rse.2018.04.050.

[101] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Hierarchical coding vectors for scene level land-use classification," *Remote Sens.*, vol. 8, no. 5, p. 436, May 2016, doi: 10.3390/rs8050436.

[102] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Deep filter banks for land-use scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1895–1899, Dec. 2016, doi: 10.1109/LGRS.2016.2616440.

[103] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6899–6910, Dec. 2018.

[104] R. P. de Lima and K. Marfurt, "Convolutional neural network for remote-sensing scene classification: Transfer learning analysis," *Remote Sens.*, vol. 12, no. 1, p. 86, Dec. 2019, doi: 10.3390/rs12010086.
[105] J. Chen, H. Huang, J. Peng, J. Zhu, L. Chen, W. Li, B. Sun, and H. Li, "Convolution neural network architecture learning for remote sensing scene classification," 2020, *arXiv:2001.09614*.
[106] H. Sun, S. Li, X. Zheng, and X. Lu, "Bidirectional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 82–96, Sep. 2020.
[107] J. Shen, T. Zhang, Y. Wang, R. Wang, Q. Wang, and M. Qi, "A dual-model architecture with grouping-attention-fusion for remote sensing scene classification," *Remote Sens.*, vol. 13, no. 3, p. 433, Jan. 2021, doi: 10.3390/rs13030433.
[108] H. Alhichri, A. S. Alswayed, Y. Bazi, N. Ammour, and N. A. Alajlan, "Classification of remote sensing images using EfficientNet-B3 CNN model with attention," *IEEE Access*, vol. 9, pp. 14078–14094, 2021, doi: 10.1109/ACCESS.2021.3051085.
[109] X. Wang, L. Duan, A. Shi, and H. Zhou, "Multilevel feature fusion networks with adaptive channel dimensionality reduction for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3070016.
[110] D. Wang, C. Zhang, and M. Han, "MLFC-Net: A multi-level feature combination attention model for remote sensing scene classification," *Comput. Geosci.*, vol. 160, Mar. 2022, Art. no. 105042, doi: 10.1016/j.cageo.2022.105042.
[111] X. Wang, H. Xu, L. Yuan, W. Dai, and X. Wen, "A remote-sensing scene-image classification method based on deep multiple-instance learning with a residual dense attention ConvNet," *Remote Sens.*, vol. 14, no. 20, p. 5095, Oct. 2022, doi: 10.3390/rs14205095.
[112] C. Li, Y. Zhuang, W. Liu, S. Dong, H. Du, H. Chen, and B. Zhao, "Effective multiscale residual network with high-order feature representation for optical remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3075257.
[113] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283.
[114] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io
[115] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," 2020, *arXiv:2008.05756*.
[116] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.

ANAS TUKUR BALARABE received the B.Sc degree in computer science from Usmanu Danfodiyo University, Sokoto, Nigeria, in 2010, the M.Sc. degree in networking and data communications from Kingston University, London, U.K., and the Ph.D. degree from the University of Portsmouth, U.K. His research interests include data communications, image processing, deep learning and image classification using traditional CNNs, and transfer learning models, object detection, and semantic segmentation.

IVAN JORDANOV received the B.Sc. degree in mechanical and electrical engineering from Naval University, Bulgaria, the M.Sc. degree in applied mathematics from the Technical University of Sofia, and the Ph.D. degree in computer-aided optimization. He is a Professor of Computational Intelligence and the Leader of the Computational Intelligence Research Group, School of Computing, University of Portsmouth, where he joined, in 2003, after three years with De Montfort University as a Senior Lecturer; two years with the University of Wales Institute, Cardiff, as a Senior Researcher; and 16 years with the Technical University Sofia as an Associate Professor.