Adaptive Multiscale Superpixel Embedding Convolutional Neural Network for Land Use Classification

Huaizhong Zhang, Callum Altham, Marcello Trovati, Ce Zhang, Iain Rolland, Lanre Lawal, Dozien Wegbu, and Nemitari Ajienka

Abstract—Currently, a large number of remote sensing images with different resolutions are available for Earth observation and land monitoring, which are inevitably demanding intelligent analysis techniques for accurately identifying and classifying land use (LU). This article proposes an adaptive multiscale superpixel embedding convolutional neural network architecture (AMUSE-CNN) for tackling LU classification. Initially, the images are parsed via the superpixel representation so that the object-based analysis (via a superpixel embedding convolutional neural network scheme) can be carried out with the pixel context and neighborhood information. Then, a multiscale convolutional neural network (MS-CNN) is proposed to classify the superpixel-based images by identifying object features across a variety of scales simultaneously, in which multiple window sizes are used to fit to the various geometries of different LU classes. Furthermore, a proposed adaptive strategy is applied to best exert the classification capability of the MS-CNN. Subsequently, two modules are developed to fully implement the AMUSE-CNN architecture. More specifically, Module I is to determine the most suitable classes for each window size (scale) by applying majority voting to a series of MS-CNNs Module II carries out the classification of the classes identified in Module I for the given scale used in the MS-CNN and, therefore, complete the LU classification of the entire classes. The proposed AMUSE-CNN architecture is both quantitatively and qualitatively validated using experimental results show the superior performance of our approach against several state-of-the-art techniques.

Index Terms—Convolutional neural network (CNN), land use (LU) classification, superpixel embedding CNN, very fine spatial resolution remotely sensed imagery.

I. INTRODUCTION

INDUSTRIAL advancement and a constant need for energy have negatively impacted the environment, which has led to severe degradation to the natural vegetation ecosystem. In turn, this has created substantial effects on the land use (LU), such as loss of prime agricultural lands, destruction of important wetlands, and increased flood risk [1]. However, having access to reliable LU maps, categorizing scene images into a discrete set of meaningful LU classes, greatly assists these tasks in natural resource monitoring and sustainable land management [2]. As the rate of LU change grows ever faster, particularly in developing countries [3], the importance of having accurate up-to-date LU information grows too. Although these maps can be produced by hand, this is both time-consuming and expensive to do [4]. As access to high-power computing resources has significantly increased in recent years, it is perhaps unsurprising that various computational techniques have been presented, which aim to automate this task. In early days, due to the low spatial resolution of satellite images (such as Landsat series), the pixel-level classification paradigms dominated the studies in the field. Some popular machine learning methods had been successfully applied for categorizing scene images at pixel level, including support vector machines (SVMs) and random forests [5]. However, while achieving a fine spatial resolution of remote sensing images, the performance of these pixel-level methods has not improved due to the inherent issues in high-resolution images, such as the within-class variability and low between-class separability [6]. As a result, the object-level delineation and the analysis of remote sensing images have emerged to study semantic entities or scene components rather than individual pixels, as the solution paradigm named object-based image analysis (OBIA) [7], [8], [9]. A typical method, OBIA-SVM [10], [11], which incorporates this object-based paradigm into the traditional SVM solution has successfully moved from pixel-based techniques toward object-based representation for better LU classification.

In the last decade, however, the successes of deep learning methods in many areas have led those in the remote sensing community to investigate their potential to become the new state of the art in the field. From convolutional neural networks (CNNs) [12], [13], to recurrent neural networks [14], to autoencoders [15], it is now commonplace to find deep learning methods being used to process remote sensing data, such as multispectral imaging [6], [16] and hyperspectral imaging [17], [18], and then, they often perform tasks such as LU classification with phenomenal precision [5], [6]. Among these methods, the pixelwise CNN had been widely applied by combining...
a pixel-based classifier with a context-based classifier, such as the multilayer-perceptron-based CNN [19], [20] providing different feature representations with strong complementarity. Recently, an object-based CNN (OCNN) [16] method has also been broadly accepted in LU classification, which incorporates the OBIA technique into the CNN framework for better learning LU objects via within-object and between-object information.

However, while deep learning methods are capable of producing high-accuracy LU classification maps, they are rising to the challenge that they often require large volumes of labeled data to do so, which can be nearly as time-consuming and expensive as producing LU classifications by hand [5]. There have, therefore, been various studies that have implemented techniques such as transfer learning [21] and active learning [22] in order to attempt to replicate high-accuracy results with smaller quantities of training data; however, their performance improved is limited especially for the data with complicated scenes. To address this challenge, a two-stage approach for automatic LU classification, decent-sized training data, and high-accuracy performance has been developed. More specifically, the multiscale superpixel CNN (MS-CNN) model is first proposed for allowing object features to be learnt across multiple scales simultaneously, and then, an adaptive strategy based on majority voting is developed to exert the role of the MS-CNN to find the most suitable scale for each class in order to obtain best LU classification. To this end, the MS-CNN is designed to replicate the ability of deeper CNN models to extract features [23] but does so without using such a deep network in order to avoid requiring the very large quantities of training data necessary to protect against overfitting and to ensure that the model generalizes well. The MS-CNN is, therefore, built upon an initial image segmentation using superpixels [24], where the superpixel segmentation is used to extract image patches in order to inform targeted sampling locations such that fewer locations need CNN classification. Thus, the MS-CNN is an object-based CNN with contextual features for LU classification. Nonetheless, in the viewpoint of its model architecture, the MS-CNN aims to extract multiscale features to classify all the objects without considering the geometric characteristics of each class; for example, residential class may be better identified using a small scale, but highway class could be well recognized using a large scale in this study. For exploring this scale adaptability in the MS-CNN and improving the performance of LU classification, an Adaptive Multiscale Superpixel Embedding Convolutional Neural Network (AMUSE-CNN) architecture is developed to sufficiently exert the classification capabilities of the MS-CNN when dealing with complicated geographical situations. To this end, the AMUSE-CNN architecture is designed to consist of two modules that each one performs its own respective function. Module I aims to find the most suitable classes for each window size (scale) by applying the majority voting to the outcomes of a series of MS-CNNs. Module II is used to apply the scale identified in Module I to the associated classes so that these classes are able to be better classified compared to the use of a singular MS-CNN. With the seamless integration of the MS-CNN model and the adaptive strategy of scale selection, the proposed AMUSE-CNN architecture is effective and efficient to perform LU classification, as an adaptive, multiscale, and object-based LU classification system.

We compare our method against the state-of-the-art techniques, OCNN [16], OBIA-SVM [11], and pixelwise CNN [20], on two real world datasets in Nigeria.

The main contributions of this study can be summarized as follows.

1) An adaptive and object-based LU classification system is developed in terms of a superpixel embedding representation of remote sensing images.
2) For better extracting object features with regard to diverse geometries of LU objects, the MS-CNN model is designed by applying the multiple window sizes to perform LU classification.
3) An adaptive strategy of scale selection is proposed to improve the performance of the MS-CNN using majority voting. The generated LU classification system is named AMUSE-CNN.
4) Two urban areas (Lagos and Kano) are used and annotated to validate the proposed AMUSE-CNN architecture.

II. PROPOSED METHOD

The proposed method is composed of two parts. First, an MS-CNN is designed to initially implement the LU classification capability with multiple window sizes. Based on this MS-CNN model, an AMUSE-CNN is proposed to provide an overall adaptive and accurate schematic solution for LU classification, by considering the fact that the MS-CNN capability is limited due to being reliant on the input of all data into predetermined window sizes.

A. AMUSE-CNN Architecture

As mentioned above, the MS-CNN is reliant on window sizes that, therefore, results in a flawed LU classification as not all the individual classes perform best when just using a given window size for a given class. This concern has driven the design and implementation of the AMUSE-CNN architecture with an adaptive strategy for enhancing the scale capability of the MS-CNN in LU classification. In fact, the AMUSE-CNN architecture consists of two modules whose block diagrams are shown in Fig. 1. Module I aims to automatically assign each LU class to its most suitable window size through majority voting that can then result in the training of further MS-CNN models in Module II, where each class is trained to its highest potential using its most suitable window size obtained from Module I. The relevant functions of each module are described below.

1) Module I: In this module, three specified window sizes are used to separately conduct the best selection of window size for each class. In this study, the window sizes include 48, 96, and 120. As shown in Fig. 1, two functions are designed to carry out the selection implementation for each specified window size.

a) Initial model training: A series of four separate MS-CNN architectures are trained with a given window size in order to generate a series of initial accuracies for each class, resulting in $4 \times N$ votes for each class, where $N$ is the number of window sizes, which is 3 in this study.

b) Majority voting: Majority voting is applied to determine the most suitable window size for each class by comparing
the accuracies obtained during initial training, with the highest voted window size for each class being assigned to said class.

2) Module II: This module is created to conduct the new MS-CNN model training so that the accurate LU classification can be achieved. To this end, an MS-CNN model is trained for each window size, in which only the previously assigned classes in Module I are involved in order to gain the highest possible accuracies for these classes; for example, in this study, Classes 1, 3, and 6 get involved in the model training for the case of the window size 48, as shown in Fig. 1. Afterward, the models obtained are used to carry out the classification of the respective classes so that all the LU classes are well finally classified, and the classification results are then output into a single table, as shown in Fig. 1.

B. MS-CNN Model

This MS-CNN model is proposed to implement a multiscale LU classification CNN, in which input data are rescaled in order to produce data at various sizes. In this study, three window sizes are used, namely, full, half, and quarter resolutions, which means that when any image is entered into the MS-CNN model as input data, the image dimensions will be halved and quartered in order to be provided as input to each individual CNN block. The block diagram of the MS-CNN model can be seen in Fig. 2.
1) **LU Classification Problem Descriptor:** The set $X$ of input data consists of $A \times B$ vectors, where $A$ and $B$ are the height and width of the data image, respectively. This can be expressed as

$$X = \{x_1, \ldots, x_{A \times B}\}, \quad x_i \in \mathbb{R}^3.$$  

(1)

Each $x_i$ also has a corresponding $u_i$ and $v_i$ corresponding to the pixel’s vertical and lateral position in the image data, respectively, where

$$0 \leq u_i < A, \quad 0 \leq v_i < B.$$  

(2)

The set $Y$ of output data contains each pixel corresponding to the pixel’s vertical and lateral position in the image data, respectively, where

$$0 \leq u_i < A, \quad 0 \leq v_i < B.$$  

(3)

The training set, $\mathcal{L} \subset X$, is composed of a subset of labeled pixels for which target labels $Y_L \subset Y$ are given. The task of LU classification is to estimate the unknown labels.

2) **Superpixel-Based LU Classification:** By grouping pixels that have sufficient similarity to one another (in terms of their spatial proximity and spectral homogeneity), it can be expected that such groups of pixels will tend to belong to a common output class. It is this assumption upon which superpixel-based segmentation methods are based. Exactly what metric is used to measure similarity and what is deemed “sufficient” similarity will vary with the choice of the segmentation algorithm and its parameters, but it suffices for now to say that algorithms that attempt to perform such a task exist.

Using the term superpixel to refer to these segmented pixels, each superpixel set $S$ is a subset of $X$. The set $X$ is divided into nonoverlapping superpixel sets whose collective union equals $X$, i.e.,

$$S_\alpha \cap S_\beta = \emptyset \quad \forall \beta \neq \alpha$$  

(4)

and

$$S_\alpha \cup S_\beta \cup \ldots = X.$$  

(5)

The shorthand notation $i \in S$ will be used to refer to the set $\{i \mid x_i \in S\}$. The common output class assumption supposes that

$$y_i = y_j \quad \forall i \in S, \forall j \in S.$$  

(6)

3) **LU Classification of Superpixels Using Data Windows:** In order to harness the power of convolutional layers to perform data-driven feature extraction, a second assumption is made, which considers that the LU label of the pixels within a superpixel depends only on a local neighborhood of the input data, a square window, $S_w$, centered on the segment’s centroid. This allows any superpixel of arbitrary geometry to be classified using a square of the input data in its vicinity. Put probabilistically

$$P(y_i | S) = P(y_i | S | x_i \in S_w).$$  

(7)

The superpixel centroid vertical and horizontal positions, denoted $u_S$ and $v_S$, respectively, can be found according to

$$u_S = \frac{\sum_{i \in S} u_i}{|S|}, \quad v_S = \frac{\sum_{i \in S} v_i}{|S|}.$$  

(8)

where $|S|$ refers to the cardinality of the superpixel set or equivalently the number of pixels contained within the superpixel. The shorthand $i \in S_w$, used to index pixels contained within the superpixel local neighborhood, refers to the set

$$\{i : -W/2 \leq u_i - u_S < W/2, -W/2 \leq v_i - v_S < W/2\}.$$  

(9)

where $W$ is the width and height in pixels of the data neighborhood considered relevant for LU classification.

4) **Data-Driven Feature Extraction Using CNNs:** CNNs are designed such that successive convolutional layers are capable of learning increasingly abstract and translationally invariant features using relatively few parameters [25]. This makes them very powerful in many computer vision tasks, particularly performing LU classification such as OCNN [16] and pixelwise CNN [20]. While these methods are capable of producing state-of-the-art performance, their ability to learn multiscale features is limited by how many filters they contain and the depth of their architecture, which are in turn, limited by the amount of training data available. With insufficient training data, deeper CNNs are liable to overfit training data and perform poorly on unseen data regions [19].

5) **MS-CNN Architecture:** In order to more easily allow a model to identify features at various scales, we propose the MS-CNN architecture, as shown in Fig. 2, which directly operates at multiple scales simultaneously. For a given scale, if features that would distinguish one class from another exist at that scale, the CNN block (see Fig. 2) that has filters of size similar to those features ought to more readily learn to identify them.

The design of the MS-CNN model is based upon the notion that, when working with data resolution as high as 50 cm per pixel, some of the features useful for performing LU classification might be comparatively large in terms of the number of pixels they make up. While a deep CNN would likely be capable of identifying these features, the MS-CNN model should be able to learn these features with comparatively fewer layers used in succession.

Fig. 2 illustrates that the proposed MS-CNN architecture has three main components: Multiple Scales, CNN Blocks, and Classification Block. Multiple Scales provide multiscale routes for the MS-CNN to learn features and parallel pathways are used, where each pathway operates at a different scale (e.g., half or quarter resolution). The component of CNN Blocks goes to extract the features for each pathway with the respective scale. Each convolutional block is made up of repeated units, consisting of a convolutional layer, a maximum pooling, and a rectified linear unit (ReLU) activation function. In Classification Block, the yielded feature maps of the parallel pathways are then combined using a simple linear combination, which is encompassed within the “weighted feature combine” layer, which has one scalar parameter per resolution pathway such that its output, $\tilde{X}$, is given by

$$\tilde{X} = \sum_i w_i x_i$$  

(10)

where $X_i$ refers to one of its inputs coming from the output of the $i$th pathway. This combines the features identified at each
of the network scales into one feature map according to a pathway importance weight \( w_i \), whose value is based on empirical knowledge. In this study, each pathway is assumed to be equally important, so \( w_i = 1/3 \) is given for each \( i \) as three pathways are involved. Subsequently, as shown in Fig. 2, the resulting feature map is flattened and classified using fully connected layers and finally applying a softmax activation function to the model output.

### III. RESULTS AND ANALYSIS

#### A. Area and Data of Study

Using very fine spatial resolution remotely sensed optical images that were released as part of Digital Globe’s Open Data Program, two cities of Lagos and Kano, Nigeria, were selected for LU classification analysis. The data, seen in Fig. 3, have three bands (red, green, and blue) and a spatial resolution of 50 cm per pixel.

#### B. Data Annotation

The OpenStreetMap (https://www.openstreetmap.org) definition of LU (“what the area is used for by humans”) will be adopted as the definition of LU used in this study. Following a consideration of the LU classes recognized by OpenStreetMap and a visual analysis of the Lagos and Kano data, the set of LU classes described by Tables I and II was adopted.

By labeling polygons within the raster, which were accurate representations of each of these LU classes, a set of labels was obtained, which could be used to represent both the training and validation sets. The data annotation was not exhaustive, with the total annotated area making up 36.8% of the total area.

The labeled polygons were randomly assigned into one of five sets, with each acting as a validation set for one experiment in a fivefold cross-validation analysis, with the others acting as that particular k-fold experiment’s training data.

#### C. Initial Image Segmentation

Different segmentation algorithms place different importances on various measures of pixel homogeneity (e.g., texture, color, and spatial proximity). In order to segment the data into superpixels that would represent locally coherent subregions with a shared LU class [as per the assumption of (6)], the mean-shift algorithm [26], implemented using the open-source software Orfeo Toolbox, was selected. The parameters of the algorithm were adjusted in order to obtain superpixels that were small enough to sufficiently separate individual LU classes but not so small that the quantity of superpixels that made up the image was unnecessarily high, which would increase inference times (the model inference time increases approximately linearly with the number of superpixels in the data). The mean-shift algorithm has three main hyperparameters: the spatial radius, the range radius, and the minimum region size. Following a trial-and-error approach, these were set to 30, 8, and 196, respectively.

#### D. Parameter Settings

1) **MS-CNN Model:** For the purposes of this investigation, MS-CNN models were trained using three scales (full, half, and quarter resolution). Various models with various window sizes have been trained, in order to determine the optimal window size for performance. The CNN block shown in Fig. 2 for the models trained consisted of three repeated blocks, each consisting of a convolutional layer (32 \( 3 \times 3 \) kernels), a maximum pooling layer (2 \( \times \) 2 window), and a ReLU activation function. The first dense layer had 32 neurons and the second had seven neurons (one for each LU class).

2) **AMUSE-CNN Architecture:** For effectively implementing the adaptive strategy, four separate MS-CNN models are used to conduct the initial model training, and thus, 4 \( \times \) 3 = 12 outcomes participate in majority voting (here, three window sizes used in this study).

3) **Sample Size:** With the annotated data introduced in Section III-B, the number of samples used for training and testing MS-CNN and AMUSE-CNN is shown in Table III.
TABLE III

| LU Class (Lagos)   | Training | Test | LU Class (Kano) | Training | Test |
|-------------------|----------|------|----------------|----------|------|
| Brownfield/Construction | 5592     | 1894 | Road/open land  | 16688    | 3065 |
| Highway           | 7120     | 1893 | Commercial     | 37472    | 3044 |
| Industrial        | 13038    | 1841 | Industrial     | 18408    | 3184 |
| Lagoon            | 46672    | 1815 | Residential    | 14752    | 3074 |
| Natural Conservation | 17544  | 1001 | Retail         | 23928    | 3235 |
| Residential       | 6320     | 1865 | Road           | 20136    | 3108 |
| Retail            | 9760     | 1843 | Vegetation     | 32144    | 2985 |
| Total             | 107646   | 13052| Total          | 319710   | 22114 |

E. LU Classification Results and Analysis

For well evaluating the AMUSE-CNN model, three state-of-the-art LU classification models were used as benchmarks to conduct the performance comparison: OBIA-SVM [11], Pixelwise CNN [19], and OCNN [16]. In addition, the performance of the MS-CNN at each scale was also presented for showing the role of the adaptive strategy used in the AMUSE-CNN. In the following subsections, a quantitative analysis is first carried out, followed by a qualitative comparison.

1) Quantitative Results: Tables IV and V show the classification accuracies achieved for each of the models assessed. The classification accuracy is measured as the percentage of the area in the validation set representing a given class that is correctly classified. Average accuracy (AA) is calculated by averaging the classification accuracies over all the classes. Equivalently, the overall accuracy (OA) is measured as the percentage of the area of all the polygons in the validation set that is correctly classified. The accuracies presented are the averages achieved over fivefold cross validation.

It is clearly seen in Tables IV and V that the AA 93.25% of AMUSE-CNN in Kano is significantly better than that 85.44% in Lagos, while its OA 90.16% in Kano is obviously inferior to that 96.71% in Lagos. In fact, these accuracy differences are determined by the geographical complexities of two cities and the nature of the relevant LU classes. In Lagos, some classes are very similar in color and geometries such as Residential and Retail, which leads to the ineffective differentiation of the relevant classes, resulting in lower AA in Lagos compared to Kano. Fig. 8 presented in Section III-E2 visually shows how this issue affects the performance of the relevant methods. On the other hand, the lagoon class takes most of the area in Lagos, so its high detection accuracy dominates the OA of Lagos, which, therefore, greatly helps Lagos achieve better OA than Kano.

To demonstrate the benefits of the adaptive strategy applied in the AMUSE-CNN, the results for the MS-CNN model at each single scale are presented in Tables IV and V. The AMUSE-CNN clearly outperforms the MS-CNN in both Lagos and Kano. For example, for Residential in Lagos, the MS-CNN with the scale 120 achieves the best accuracy 60.99%, while the AMUSE-CNN reaches 78.34%; for Industrial in Kano, 72.81% is the best accuracy obtained by the MS-CNN with the scale 96, while 91.11% is achieved by the AMUSE-CNN.

Compared to the state-of-the-art methods, the traditional handcrafted method, OBIA-SVM, is clearly ineffective to deal with LU classification in these two datasets due to its AAs for both the datasets being 42.67% and 46.60%, respectively. In particular, Retail (0.17%) in Lagos is almost all misclassified due to being similar to Residential in color. As for Pixelwise CNN and OCNN, their performance is much better than OBIA-SVM. However, our AMUSE-CNN significantly outperforms these two methods in both AA and OA such as AA: 85.44% versus 78.59% versus 82.20% in Lagos and 93.25% versus 81.70% versus 83.70% in Kano. Tables IV and V show that the AMUSE-CNN is able to well classify some difficult classes; however, Pixelwise CNN and OCNN perform poorly, for example, Residential and Retail in Lagos, and Industrial and Residential in Kano.

On a class-by-class basis, the AMUSE-CNN optimally classifies six of the seven LU classes for Lagos except for Brownfield Construction (Pixelwise CNN being best 85.43%). The nearest AA accuracy benchmark, OCNN, can classify the classes averagely better than any of the other methods presented for both Lagos and Kano. For the most part, the Highway and Lagoon LU classes in Lagos, and the Road and Vegetation in Kano are easiest to classify (with all models achieving >75% accuracy).

The classes that have proved most difficult to classify (Industrial and Residential) are both classes where the AMUSE-CNN has greatly excelled relative to the benchmarks particularly in Kano.

For further demonstrating the overall performance and strength of AMUSE-CNN, the receiver operating characteristic (ROC) curve is plotted for each model used in this study, which Figs. 4 and 5 are presented for Lagos and Kano, respectively. These figures obviously illustrate that the AMUSE-CNN gets higher TPR (sensitivity) and lower FPR (1 − specificity) at any threshold setting in comparison to the benchmarks. Thus, the classification performance of the AMUSE-CNN in both Lagos and Kano appears markedly superior to those of the benchmarks.

2) Qualitative Results: In order to visually compare the performance, Fig. 6 presents one randomly selected region of Lagos (2000 × 2000 pixel area) alongside the ground truth (GT) annotations for that region and LU classification output for each of the models trained. Where used, the subscripts refer to the window size of the model. This visual results show that the overall performance of the AMUSE-CNN is much better than the benchmarks and also the MS-CNN although the classification of some classes is still not best conducted due to the similarity between classes such as a small portion of “Retail” being misclassified as “Industrial.” Fig. 7 shows another selected region of the Kano data for visually showing the performance of our method and the benchmarks. It is clearly illustrated that the AMUSE-CNN well classified most classes except for a small portion of “Retail” being misclassified as “Residential.” Promisingly, referring to the GT, as shown in Fig. 7, the AMUSE-CNN can perform the classification much more accurate than the benchmarks and the MS-CNN. For noticeably looking into the classification details and issues of the given classes, we zoom on the selected parts of the images and present the relevant classification results of these zooming-in portions as follows.

“Residential” and “Retail” are the classes with similar geographical information in Lagos, and the benchmarks thus often wrongly classify these two classes. Fig. 8 illustrates the classification results of a selected region in which the AMUSE-CNN
TABLE IV
CLASSIFICATION ACCURACY (%) COMPARISON AMONG OBIA-SVM, PIXELWISE CNN, OCNN, MS-CNN, AND AMUSE-CNN FOR LAGOS

| LU Class                      | OBIA-SVM | Pixelwise CNN | OCNN | MS-CNN_{48} | MS-CNN_{96} | MS-CNN_{120} | AMUSE-CNN_{48, 96, 120} |
|-------------------------------|----------|---------------|------|-------------|-------------|--------------|------------------------|
| Brownfield/construction       | 34.01    | 85.43         | 73.55| 73.88       | 72.05       | 75.77        | 77.37                  |
| Highway                       | 80.62    | 80.33         | 78.14| 75.34       | 77.43       | 64.41        | 85.39                  |
| Industrial                    | 16.28    | 73.65         | 61.07| 56.70       | 75.70       | 60.22        | 76.85                  |
| Lagoon                       | 59.31    | 99.84         | 99.79| 99.87       | 99.84       | 99.86        | 99.95                  |
| Natural Conservation          | 11.70    | 84.65         | 92.63| 91.42       | 88.13       | 90.98        | 97.62                  |
| Residential                   | 56.66    | 51.56         | 46.09| 51.26       | 57.35       | 60.99        | 78.34                  |
| Retail                        | 0.17     | 74.69         | 48.25| 66.24       | 68.41       | 79.60        | 82.56                  |
| AA                            | 42.57    | 78.59         | 82.20| 73.53       | 76.59       | 75.98        | 85.84                  |
| OA                            | 81.80    | 93.71         | 92.89| 93.73       | 93.49       | 94.43        | 96.71                  |

The presented accuracies (AA and OA) are the averages achieved over fivefold cross validation.

TABLE V
CLASSIFICATION ACCURACY (%) COMPARISON AMONG OBIA-SVM, PIXELWISE CNN, OCNN, MS-CNN, AND AMUSE-CNN FOR KANO

| LU Class                     | OBIA-SVM | Pixelwise CNN | OCNN | MS-CNN_{48} | MS-CNN_{96} | MS-CNN_{120} | AMUSE-CNN_{48, 96, 120} |
|------------------------------|----------|---------------|------|-------------|-------------|--------------|------------------------|
| Bare/open land               | 29.2     | 85.9          | 87.2 | 76.3        | 81.0        | 82.57        | 98.41                  |
| Commercial                   | 43.8     | 77.5          | 82.0 | 72.4        | 78.9        | 78.73        | 92.62                  |
| Industrial                   | 15.5     | 62.8          | 71.1 | 62.6        | 72.8        | 60.59        | 91.11                  |
| Residential                  | 29.8     | 67.4          | 72.9 | 59.2        | 75.5        | 61.85        | 85.07                  |
| Retail                       | 50.7     | 89.7          | 85.8 | 80.6        | 88.1        | 78.80        | 91.95                  |
| Road                         | 78.8     | 95.4          | 91.9 | 91.2        | 94.2        | 91.59        | 97.29                  |
| Vegetation                   | 78.8     | 95.4          | 91.9 | 91.2        | 94.2        | 91.59        | 97.29                  |
| AA                           | 46.6     | 81.7          | 83.7 | 76.3        | 83.5        | 77.84        | 93.25                  |
| OA                           | 32.9     | 73.3          | 77.7 | 68.19       | 76.32       | 69.38        | 90.16                  |

The presented accuracies (AA and OA) are the averages achieved over fivefold cross validation.

Fig. 4. ROC curves showing the performance of all the methods: the AMUSE-CNN produces excellent classification performance for all the classes in comparison with other methods in Lagos.

In Lagos, “Industrial” is the class with the lowest performance of the AMUSE-CNN. Fig. 9 illustrates that the indistinguishable boundary of “Highway” and “Retail” with “Industrial” actually intervenes in the classification, thus significantly reducing the performance of the AMUSE-CNN in “Industrial.” Even so, the AMUSE-CNN obviously outperforms the benchmarks, as visually presented in Fig. 9.

Fig. 10 presents the classification results of the selected region in Kano that the performance of the AMUSE-CNN in “Industrial” and “Residential” is with the lowest accuracy in comparison with other classes. It illustrates that the benchmarks is prone to classifying “Residential” as “Industrial,” while a small portion of “Industrial” is wrongly classified as “Residential” with the AMUSE-CNN. However, our proposed method is clearly better at conducting the classification than the benchmarks for these two classes.
F. Ablation Study

For examining the benefits of the superpixel solution, the nonsuperpixel-based MS-RCNN is employed to conduct one ablation study. Without loss of generality, the model is trained at the window size 48, and the results obtained are shown in Table VI. Compared with the results shown in Tables IV and V, the superpixel-based MS-CNN$_{48}$ performs superior to the pixelwise-based MS-CNN in Lagos while performing similar to the pixelwise version in Kano. On the other hand, the pixelwise MS-CNN is massively time-consuming. Particularly, its prediction time (5.41 h for Lagos) is greater dozen times over than that of our superpixel-based MS-CNN (0.13 h for Lagos).

Thus, we can conclude that the superpixel-based MS-CNN is more accurate and efficient than the nonsuperpixel-based MS-CNN. The superpixel-based MS-CNN method is, therefore, the most suitable for use in the AMUSE-CNN.

Another ablation study aims to demonstrate the role of Module II in the AMUSE-CNN, which we conducted LU classification only using Module I of the AMUSE-CNN. Table VII presents the results generated from Module I, in which the “Best”
Fig. 7. 2000 × 2000 pixel Kano image region and the corresponding GT as well as LU classification for all the methods.

Fig. 8. Results of the selected region showing the interference between “Residential” and “Retail” in Lagos.

Fig. 9. Results of the selected region showing the interference of “Highway” and “Retail” on the classification of “Industrial” in Lagos.
Fig. 10. Results of the selected region showing the classification interference between “Industrial” and “Residential” in Kano.

| Table VII |
|-----------|
| RESULTS OBTAINED FROM MODULE I |

| LU Class (Lagos) | 48 (%) | 96 (%) | 120 (%) | Best (%) | LU Class (Kano) | 48 (%) | 96 (%) | 120 (%) | Best (%) |
|------------------|--------|--------|---------|----------|------------------|--------|--------|---------|----------|
| Brownfield/Construction | 80.17  | 75.64  | 76.68   | 80.17    | Bare/open land  | 77.10  | 79.22  | 76.90   | 79.22    |
| Highway          | 77.02  | 77.56  | 72.94   | 77.56    | Commercial      | 76.34  | 80.67  | 82.56   | 82.56    |
| Industrial       | 67.55  | 59.91  | 62.92   | 67.55    | Industrial      | 64.47  | 73.88  | 75.36   | 75.36    |
| Lagoon           | 99.79  | 99.72  | 99.75   | 99.79    | Residential     | 62.50  | 78.40  | 79.28   | 79.28    |
| Natural Conservation | 90.42  | 92.47  | 90.64   | 90.64    | Retail          | 71.59  | 80.07  | 78.90   | 80.07    |
| Road             | 38.43  | 46.40  | 39.49   | 46.40    | Road            | 92.65  | 93.27  | 94.48   | 94.48    |
| Retail           | 56.65  | 54.53  | 56.06   | 56.65    | Vegetation      | 97.33  | 98.06  | 98.09   | 98.09    |
| AA               | 72.86  | 72.32  | 71.21   | 74.11    | AA              | 77.43  | 83.57  | 83.65   | 84.15    |
| OA               | 94.07  | 93.02  | 93.75   | 91.57    | OA              | 68.72  | 79.33  | 79.95   | 80.28    |

| Table VIII |
|-------------|
| COMPUTATION TIMES TO TRAIN EACH OF THE CLASSIFICATION ALGORITHMS PRESENTED FOR THE LAGOS DATASET AND MAKE LU CLASSIFICATION PREDICTIONS FOR THE COMPLETE DATA |

| Obia-SVM | Pixelwise CNN | OCNN | MS-CNN16 | MS-CNN36 | MS-CNN120 | AMUSE-CNN |
|----------|---------------|------|----------|----------|-----------|-----------|
| Train time (h) | 1.03 | 0.07 | 0.19 | 0.10 | 0.23 | 0.32 | 3.25 |
| Predict time (h) | 1.13 | 0.27 | 0.22 | 0.13 | 0.25 | 0.55 | 4.60 |

| Table IX |
|----------|
| COMPUTATION TIMES TO TRAIN EACH OF THE CLASSIFICATION ALGORITHMS PRESENTED FOR THE KANO DATASET AND MAKE LU CLASSIFICATION PREDICTIONS FOR THE COMPLETE DATA |

| Obia-SVM | Pixelwise CNN | OCNN | MS-CNN16 | MS-CNN36 | MS-CNN120 | AMUSE-CNN |
|----------|---------------|------|----------|----------|-----------|-----------|
| Train time (h) | 1.33 | 0.09 | 0.31 | 0.16 | 0.43 | 0.65 | 5.93 |
| Predict time (h) | 2.0 | 12.34 | 0.35 | 0.35 | 0.45 | 0.63 | 7.15 |

column shows the result of the best performance selected from all the scale-based results for each class. By comparison, the performance of the AMUSE-CNN with the added Module II, shown in Tables IV and V, is overall 10% higher than the best performance obtained from Module I presented in Table VII. This is evident that Module II is indispensable in the AMUSE-CNN for achieving the superior performance.

G. Computational Environment and Complexity

To compare the computational costs of the AMUSE-CNN with those of the benchmarks, an analysis that examines the amount of time to train each model and for each model to perform inference on the complete data is presented in Tables VIII and IX. The experiments have been carried out on a machine with Intel i5-9400F CPU, 32-GB RAM, and an NVIDIA Quadro M400 GPU.

Overall, the amount of time needed to train/perform inference is at a similar level for the MS-CNN and the OCNN. Among the benchmarks, Obia-SVM is quite time-consuming, while the pixelwise CNN stands out as the fastest training but the slowest inference model. For the AMUSE-CNN, due to the multiple MS-CNN models being involved in the adaptive strategy, its computation is costly and pretty time-consuming; however, in
compensation, its accuracy of LU classification is highly improved in comparison with the benchmarks.

Referring to the CNN time computation model introduced in [27] and [28], the computational complexity of the MS-CNN is estimated as \( O(N \times D \times S^2 \times M^2) \), where \( N \) is the number of the scales used, \( D \) is the number of the layers, \( S \) is the spatial size (length) of the kernel, and \( M \) is the spatial size of the output feature map. For the AMUSE-CNN, due to the calculation of majority voting, its computational complexity is approximately \( O(N^2 \times D \times S^2 \times M^2) \).

### IV. DISCUSSION

In the proposed solution scheme, the MS-CNN model is first proposed to improve upon the benchmark methods by offering direct routes for multiscale feature identification through the use of scale-specific CNN blocks. Furthermore, the AMUSE-CNN architecture is designed to explore the capability of LU classification with the use of majority voting to implement an adaptive strategy. The experimental results shown in Sections III-E1 and III-E2 demonstrate that the AMUSE-CNN does accurately outperform the state-of-the-art methods. However, for a better LU classification, there is much room to improve the AMUSE-CNN whatever the classification accuracy or the computational efficiency is.

In this study, data preprocessing is mainly based on homogeneity representation using superpixels and the mean-shift method. However, it is worth noting that remote-sensed satellite data are prone to suffering from spectral variabilities, such as vertical artifacts [29]. Some potential techniques, for example, the augmented linear mixing model [30], could be employed for improving data quality at the preprocessing stage.

It is clear that the overall performance of the AMUSE-CNN is greatly dependent on that of the MS-CNN. In Tables IV and V, the improved AA and OA offered by the MS-CNN with different scales suggests that there may be cause for further investigation into multiscale architectures for LU classification. That being said, the improvements observed are not seen across every class, and therefore, it is proved that some LU classes are better identified using multiscale feature learning than others by applying the adaptive strategy, which is the essence of the AMUSE-CNN architecture proposed. The further work could consider to use a different model selection scheme instead of majority voting.

With regard to the window sizes used, although MS-CNN models presented have used half and quarter resolution as their additional scales, this choice was somewhat arbitrary and further investigation might analyze how the choice of scales in the MS-CNN affects results. This should, therefore, result in improving the performance of the AMUSE-CNN. Similarly, the choice of operating with three scales was also somewhat heuristic, and an extension of this work might assess the balance between performance and increased computational costs by comparing MS-CNNs, which operate with more or fewer scales. In addition, four MS-CNN models are used to conduct majority voting in the current study. We could give a try to taking different number of models involved in the scale selection process such as three or five in further studies.

For sample size in this study, Table III shows the number of training and testing samples used for each class. It is worth noting that the number of samples used is largely dependent on that of the dataset available. As introduced in Section III-B, annotated data were with only 36.8% of the entire dataset, which results in what may seem to be lower numbers of each class, such as that of the retail class used in the Lagos dataset. However, the number of samples that are available for each class are influenced by how populated each urban dataset is with this particular class; thus, the number of the retail class in Lagos was incredibly small. From our empirical experience, a well balance between the numbers of samples used for each class is needed for ensuring that performance is kept as high as possible while preventing an imbalance in the data leading to issues during training such as overfitting.

To better validate the AMUSE-CNN method, although two different urban datasets are used in this study, it is anticipated that the improved results could be replicated for more datasets. The next step would be to produce similar improvements using datasets collected from other continents. This should not only offer some challenges unique to new datasets, but also present some of the same complexities faced with classifying the Lagos and Kano datasets.

Another big challenge to our proposed method is how to reduce the computational complexity and make the solution scheme more efficient. The inherent drawback of majority voting is actually seriously bringing down the calculation efficiency of the AMUSE-CNN; thus, some more efficient method of scale choice should be used in future. In addition, we could consider to improve the computational capability with better hardware such as TPU for more efficient calculation.

### V. CONCLUSION

This article presents an adaptive LU classification CNN architecture, namely, AMUSE-CNN, based on a multiscale feature learning model (MS-CNN) proposed as a means for better identifying the subtle visual cues that allow a region’s LU to be determined. Using superpixels obtained via a segmentation of the data to represent regions of locally homogenous LU, the MS-CNN learns to identify features at multiple scales simultaneously in order to classify superpixels and obtain a semantic segmentation of the data. Moreover, based on the MS-CNN, the AMUSE-CNN is designed and put forward to find the most suitable scale for each class and, thus, obtain incomparable classification accuracy by resorting to a scale selection process of majority voting. The AMUSE-CNN performance is well examined with two urban datasets, Lagos and Kano in Nigeria, by comparing to three state-of-the-art benchmarks. A barrier to the AMUSE-CNN is the time-consuming step of performing the scale selection; two ways of improving work for reducing the computational complexity could be valuably attempted in future. One could use another efficient scale section scheme instead of majority voting. Another one may optimize the calculation process with more efficient code along with the use of high-performance hardware units such as TPU. In addition, for alleviating the effect of spectral variability, such as vertical artifacts [29], an augmented linear mixing model [30] could be employed for data preprocessing.
REFERENCES

[1] U. Aid and USGS, “West Africa: Land-Use and Land-Cover Dynamics,” U.S. Aid and USGS. Accessed: Aug. 2021. [Online]. Available: https://eros.usgs.gov/westafrica/land-cover/land-use/land-cover-and-trends-nigeria

[2] A. Rahman, S. Kumar, S. Fazal, and A. Siddiqui, “Assessment of land use/land cover change in the north-west district of Delhi using remote sensing and GIS techniques,” J. Indian Soc. Remote Sens., vol. 40, no. 4, pp. 689–697, 2012.

[3] A. Rahman, S. P. Aggarwal, M. Netzband, and S. Fazal, “Monitoring urban sprawl using remote sensing and GIS techniques of a fast growing urban centre, India,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 1, pp. 56–64, Mar. 2011.

[4] P. Langat, L. Kumar, R. Koech, and M. Ghosh, “Monitoring of land use/land-cover dynamics using remote sensing: A case of Tana River Basin, Kenya,” Geocarto Int., vol. 8, pp. 1–20, 2019.

[5] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” ISPRS J. Photogramm. Remote Sens., vol. 152, no. 4, pp. 166–177, 2019.

[6] G. Cheng, X. Xie, J. Han, L. Guo, and G. Xia, “Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 13, pp. 3735–3756, 2020.

[7] T. Blaschke, “Object based image analysis for remote sensing,” J. Photogramm. Remote Sens., vol. 65, pp. 2–16, 2010.

[8] 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. 29, pp. 1453–1470, 2010.

[9] T. Blaschke et al., “Geographic object-based image analysis—Towards a new paradigm,” J. Photogramm. Remote Sens., vol. 87, pp. 180–191, 2014.

[10] A. Tzotsos, “A support vector machine approach for object based image analysis,” in Proc. 1st Int. Conf. Object-Based Image Anal., 2006, pp. 1–6.

[11] S. Zylshal, S. Sulma, F. Yulianto, J. Nugroho, and T. Sofan, “A support vector machine object based image analysis approach on urban green space extraction using Pleiades-1A imagery,” Model. Earth Syst. Environ., vol. 2, no. 54, 2016, Art. no. 54.

[12] Z. Shao and J. Cai, “Remote sensing image fusion with deep convolutional neural network,” “IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 5, pp. 1656–1669, May 2018.

[13] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, “Graph convolutional networks for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[14] D. H. T. Minh et al., “Deep recurrent neural networks for winter vegetation quality mapping via multitemporal SAR sentinel-1,” IEEE Geosci. Remote Sens. Lett., vol. 15, no. 3, pp. 464–468, Mar. 2018.

[15] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, “Two-stream deep architecture for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[16] C. Zhang et al., “An object-based convolutional neural network (OCNN) for urban land use classification,” Remote Sens. Environ., vol. 216, no. 5, pp. 57–70, 2018.

[17] D. Hong et al., “More diverse means better: Multimodal deep learning meets remote-sensing imagery classification,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 5, pp. 4340–4354, May 2021.

[18] X. Wu, D. Hong, and J. Chanussot, “Convoluational neural networks for multimodal remote sensing data classification,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2022, Art. no. 5517010, doi: 10.1109/TGRS.2021.3129413.

[19] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, “Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[20] C. Zhang, X. Pan, H. Li, A. Gardiner, J. Hare, and P. M. Atkinson, “A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification,” Photogramm. Remote Sens., vol. 140, no. 3, pp. 133–144, 2017.

[21] H. Lyu et al., “Long-term annual mapping of four cities on different continents by applying a deep information learning method to landsat data,” J. Remote Sens., vol. 10, no. 3, pp. 2072–4292, 2018.

[22] P. Liu, H. Zhang, and K. B. Eom, “Active deep learning for classification of hyperspectral images,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 10, no. 2, pp. 712–724, Feb. 2017.

[23] M. Jigou, M. Madhakila, G. Divya, R. Meghana, and S. Apoorva, “Feature extraction using convolution neural networks (CNN) and deep learning,” in Proc. IEEE 3rd Int. Conf. Recent Trends Electron., Inf. Commun. Technol., 2018, pp. 2319–2323.

[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 11, pp. 2274–2282, Nov. 2012.

[25] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

[26] D. Comaniciu and M. P., “Mean shift: A robust approach toward feature space analysis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May 2002.

[27] K. He and J. Sun, “Convolutional neural networks at constrained time cost,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 5353–5360.

[28] V. Sze, Y. Chen, T. Yang, and S. Emer, Efficient Processing of Deep Neural Networks. San Rafael, CA, USA: Morgan & Claypool, 2020.

[29] M. Coffera et al., “Vertical artifacts in high-resolution WorldView-2 and WorldView-3 satellite imagery of aquatic systems,” J. Photogramm. Remote Sens., vol. 10, no. 3, pp. 591–599, 2015.

[30] D. Hong, L. Gao, N. Yokoya, J. Chanussot, and X. Zhu, “An augmented linear mixing model to address spectral variability for hyperspectral unmixing,” IEEE Trans. Image Process., vol. 28, no. 4, pp. 1923–1938, Apr. 2019.