Hybrid Deep Learning Architecture for Land Use: Land Cover Images Classification with a Comparative and Experimental Study

Salhi Wiam[1], Tabiti Khouloud[2], Honnit Bouchra[3], SAIDI Mohamed Nabil[4], KABBAJ Adil[5]
Research Laboratory in Information Systems, Intelligent Systems and Mathematical Modeling,
National Institute of Statistics and Applied Economics,
Rabat, Morocco[1,2]
LPRI Multidisciplinary Research and Innovation Laboratory, Moroccan School of Engineering Sciences EMSI,
Casablanca, Morocco[3]
Research Laboratory in Information Systems, Intelligent Systems and Mathematical Modeling,
National Institute of Statistics and Applied Economics, Rabat, Morocco[4,5]

***Abstract*—Deep Learning algorithms have become more popular in computer vision, especially in the image classification field. This last has many applications such as moving object detection, cancer detection, and the classification of satellite images, also called images of land use-land cover (LULC), which are the scope of this paper. It represents the most commonly used method for decision making in the sustainable management of natural resources at various geographical levels. However, methods of satellite images analysis are expensive in the computational time and did not show good performance. Therefore, this paper, on the one hand, proposes a new CNN architecture called Modified MobileNet_V1 (MMN) based on the fusion of MobileNet_V1 and ResNet50. On the other hand, it presents a comparative study of the proposed model and the most used models based on transfer learning, i.e. MobileNet_V1, VGG16, DenseNet201, and ResNet50. The experiments were conducted on the dataset Eurosat, and they show that ResNet50 results emulate the other models.**

***Keywords*—Deep Learning; image classification; land use-land cover; MobileNet; ResNet; satellite images**

I. Introduction

Nowadays, the artificial intelligence domain is knowing an important development, especially in image classification. It is one of the most common challenges in computer vision [1]. Indeed, it refers to the categorization of images into one of many predetermined classes. A single image can be classified into a number of different categories. However, manually inspecting and classifying images can be time-consuming, especially when the images have a large size. Thus, automating the process would be extremely beneficial [2].

Image classification is used in several domains [3], [4] e.g. medicine, videos surveillance, economy, and agriculture especially for the classification and categorization of Land Use-Land Cover (LULC) images. Moreover, at the local, regional, and national levels, the LULC classification plays an important role in program planning, management, and monitoring [5]. On the one hand, The LULC information helps in understanding the land occupation issues, and on the other hand, it is crucial in the constitution of policies and programs that are necessary for development planning. Furthermore, it is important to track the LULC pattern's evolution throughout time in order to ensure long-term development. To accomplish sustainable urban growth and to control the random expansion of cities, urban development authorities need such planning models that allow every available piece of land to be used in the most reasonable and optimal way possible. It requires knowledge of the area's current and previous LULC. In addition, the LULC maps can be used to track changes in our ecosystem and environment [6], [7], [8].

Currently, image processing of LULC using deep learning algorithms is gaining more attention [9], [10], [11]. Since, they outperformed the classical approaches, even if the interpretation of LULC images requires good agricultural experience.

The main objective of this work is to study the efficiency of deep convolutional neural network (CNN) architectures for LULC image classification. Thus, our work aims to solve the following research questions: (1) Is there any deep learning technique that consistently outperforms the other ones? (2) Is it possible to use deep learning to correctly categorize LULC images and surpass previous methods? (3). What is the highest level of accuracy that DL can attain with LULC images ?

The contributions of our paper are the following: we design a comparison between different fine-tuned DL architectures (VGG16, DenseNet201, ResNet50 and MobileNet_V1) and a Modified MobileNet_V1 (MMN) architecture that combines MobileNet_V1 and ResNet50 on several levels namely: performance, and amount of parameters etc. Since there is a lack of LULC image datasets, the different models were evaluated on the Eurosat dataset [4] of LULC images and on the same dataset using data augmentation techniques. The rest of this paper is organized as follows: The second section presents the literature review, the section three provides the used methods and materials. Section four exposes the experimental results. Section five provides the discussion, and the paper is concluded in the last section.

II. Literature Review

A. LULC Classification Datasets

There are several LULC classes, such as developed or urban areas, farmland, and wooded areas, and so on. LULC

maps have many applications such as conservation of natural resources, entry of GIS data, delineation of tax and property boundaries, etc. Unfortunately, the existing and published datasets are rarely labeled, which makes the classification process very challenging. Thus, some datasets have been proposed, but each of them has a disadvantage. For example, the UC Merced dataset [12], [13], [14], [15], [16], which has been suggested by [16], contains 21 land use land cover classes with 100 images per class and is extracted from USGS National Map Urban Area Imagery. However, this dataset is a bad choice for a comparison based on deep learning models because it has a very small number of images.

Although PatternNet [17] and NWPU-RESISC [18] have high resolutions, they share the same issue of having 100 images per class. Additionally, SAT-6, which was developed by [19], has six classes: barren land, trees, grassland, roads, buildings, and water, and has a resolution of 1 m/pixel and a size of 28×28 and the images were produced using imagery from the National Agriculture Imagery Program (NAIP). Moreover, the AID Dataset, which was first presented by [15], includes 30 classes with 200-400 images each with a size of 600×600.

B. LULC Classification Methods

Several methods have been proposed in the field of LULC classification. There are the traditional methods based on color analysis, shape and texture, or a combination of all this information [20], [21], but their effectiveness in the real world is still constrained. To solve the limitations of the old methods, numerous modeling tools, including dynamic, statistical methodologies, have been employed to create accurate simulations that consider economic, spatiotemporal and social factors [22]. Remote sensing imaging classification, anomaly detection, and prediction issues can be resolved using machine learning modeling. Maximum likelihood classifiers, Markov chain models, support vector machines, Markov chain models and other machine learning algorithms have been historically used to classify images [23], [7], [8], [11], [24]. But, in recent years, with the advancement in the performance of computing units, several approaches in the context of deep learning have been used. In order to provide superior performance for the treatment and classification of images (categorical image mapping and classification using convolutional neural networks (CNN)), deep learning models outperform traditional models in extracting spatial characteristics at various levels of remote sensing images [25]. A method in [10] presents an architecture called Joint Deep Learning (JDL), which integrates a multilayer perceptron (MLP) and a convolutional neural network (CNN) and is implemented via a Markov process.

In [6], Scale Sequence Joint Deep Learning (SS-JDL) is introduced as a new DL method for LULC classification. The effectiveness of this SS-JDL method has been tested on the digital aerial photography of three complex and heterogeneous landscapes. Thus, In [26] a simple discriminative CNN (D-CNN) method is proposed to improve the performance of remote sensing image scene classification. Furthermore, the spectral and spatial information content of remote sensing images is captured and utilized in [27] and employing a multi-attention method using a bidirectional long-term memory network. To address the imbalance in LULC classes, a recent work in [28] recommended the use of oversampling, while in [29], the authors suggest using consensus-based collaborative multi-label learning to balance labels. To explain the model predictions in [30], the authors tested various interpretable artificial intelligence methodologies while using the DenseNet model [31]. The framework for learning deep representations of spectral bands for the same purpose is finally proposed in [32].

The deep CNNs utilized in this paper were improved using a pre-trained network in the context of deep learning. The ILSVRC-2012 image classification challenge dataset served as the main source of pre-training data for the networks. Even though these pre-trained networks were developed on images from a completely unrelated field, the features generalized effectively [33], [34]. As a result, it was discovered that the pre-trained networks were appropriate for classifying remote sensing images. Consequently, we use for comparison some deep CNN models like VGG16 [35], ResNet50 [36], MobileNet_V1 [37] and DenseNet201 [31] which represent the state of the art for the classification of the introduced LULC classes.

III. Materials and Methods

Recent studies have proved that DL algorithms are very efficient in image processing and computer vision [38]. They have been applied in various remote sensing imaging modalities with high performance [39] in segmentation, detection and classification. Although these technologies have shown promising results in remote sensing, they require a lot of data. Motivated by the success of DL and remote sensing image processing, our work will present a comparative study between a new CNN architecture that combines the MobileNet_V1 architecture with the concept that characterizes the ResNet50 architecture and different DL architectures (VGG16, ResNet50, MobileNet_V1 and DenseNet201) (application in the LULC domain).

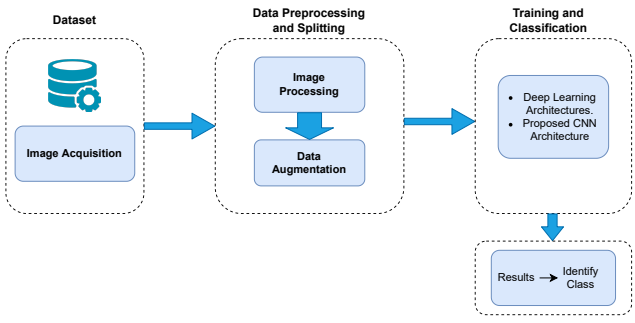

Fig. 1. Process of LULC Classification

Fig. 1 illustrates the used processes to compare the various models. The first step is image acquisition. Then, the preprocessing and splitting of data. Finally, training, and classification.

A. LULC classification

LULC data indicates the extent of an area covered by forests, wetlands, impervious surfaces, agriculture, and other

land and water types [40]. It refers to the categorizing or classification of human activities and natural landscape elements over a period of time using scientific methodologies.

With this information, it is possible to better understand the effects of natural phenomena and human use of the landscape [40]. Maps can be also used to assess urban growth, model water quality issues, predict flood and storm surge impacts, track wetland loss and potential sea level rise impacts, prioritize conservation efforts, and compare land-cover changes with environmental effects or linkages in socioeconomic changes like population growth [23].

In the literature, there exist several approaches of features extraction from remote sensing images. In this context, CNNs are a sort of neural network [16], that is considered as the state of the art of image classification approaches [4] due to its good results.

B. Data Preprocessing

Even if some DL architectures can handle 13 bands of images as an input, they can not use the TIFF files. Therefore, all the images should be converted to RGB format. In order to avoid over-fitting, data augmentation was applied after the preprocessing and only to the training data. Additionally, some geometrical transformations were included i.e. rescaling, rotations, shifts, shears, zooms, and flips. We also used multiple augmentation approaches to build a new image from each input image.

C. Training and Classification

1) Convolutional neural network: CNN is not merely a deep neural network with many hidden layers. It is a deep network that analyses and recognizes images based on the brain's visual cortex. This is how CNN differs from previous neural networks in terms of idea and function [41]. It also gets its name from an operation called convolution, a linear mathematical action involving matrices. CNN has several layers, including a convolutional layer, a non-linearity layer, a pooling layer, and a fully-connected layer [1].

The pooling and non-linearity layers do not contain any parameters, while the convolutional and fully-connected layers do. In terms of machine learning issues, CNN performs exceptionally well. Specifically, in applications dealing with high resolution images [42]. CNN collects features from images automatically by constructing many convolution layers [43] resulting in a feature hierarchy. The shallower front convolution layer employs a smaller perceptual domain, allowing it to learn some features of the local image, whereas the deeper back convolution layer employs a broader perceptual domain, allowing it to learn more features [1].

2) Deep learning architectures:

a) VGG16: Simonyan and Zisserman proposed the Visual Geometry Group (VGG) CNN architecture in 2014, and it won the ILSVR competition. VGG16 enhances the decision function by replacing the huge filters with numerous 3×3 filters one after the other and 2×2 for max pooling, resulting in a more discriminative decision function (decrease in the number of parameters). VGG16 is easy to understand and utilize because it only has two layers: convolution and pooling [35], [44]. There are a total of 16 weighted layers, as indicated by the number 16.

b) ResNet50: ResNet50 is a deep residual network created by K. He, X. Zhang, S. Ren, and J. Sun in 2015, and it won the ILSVRC 2015. It's a model that tackles the problem of disappearing gradients (gradient tends to zero quickly) by using a novel notion called skip connection (stacking convolution layers and adding the original input to the convolution block's output). ResNet50 is made up of five stages, each with a convolution and identity block and three convolution layers. ResNet50 uses images with a resolution of 224×224 pixels and has 50 residual networks [45], [36].

c) MobileNet_V1: MobileNet_V1 is a 28-layer architecture that accepts input images that are $224 \times 224 \times 3$ pixels in size. It introduced the notion of Depthwise Separable Convolution, in which two 1D convolutions with two kernels are used instead of a single 2D convolution to minimize the model's size (fewer parameters) and complexity (fewer multiplications and additions). MobileNet_V1 also adds two new global hyperparameters (width multiplier and resolution multiplier) that allow model developers to trade latency or accuracy for speed and compact size based on their needs [37], [46].

d) DenseNet201: The dense convolutional network (DenseNet201) is a CNN with 201 depth layers that takes a 224×224 input image. DenseNet201 is a ResNet improvement that incorporates dense layer connections. Each layer receives more inputs from the all preceding layers and delivers its own feature maps to the layers below it. Concatenation is used by each layer to obtain "collective knowledge" from all previous layers. DenseNet outperforms ordinary networks by reducing processing requirements, reducing the number of parameters, encouraging feature reuse, and improving feature propagation [31], [47].

D. Proposed Architecture Modified MobileNet_V1 (MMN)

The input layer, convolutional layers, pooling layers, fully connected layers, and the output layer are the five layers that make up a CNN model. Furthermore, a CNN model can be trained from end to end to enable feature selection, extraction and prediction or classification. It's difficult to figure out how a network understands and analyzes an image. However, features extracted by layers of a network have been demonstrated to outperform human-built features [48].

Our model is a combination of the MobileNet_V1 and ResNet50 architectures. We took the same architecture of the MobileNet_V1 which uses the notion of depthwise separable convolution, and we optimized it by adding the notion of skip connection, which characterizes the ResNet50 model. It means that we add the output of a layer to the next one. This is to try to improve the results of MobileNet_V1 by providing an alternate gradient path, and it has been experimentally proven that these additional paths are frequently beneficial for model convergence.

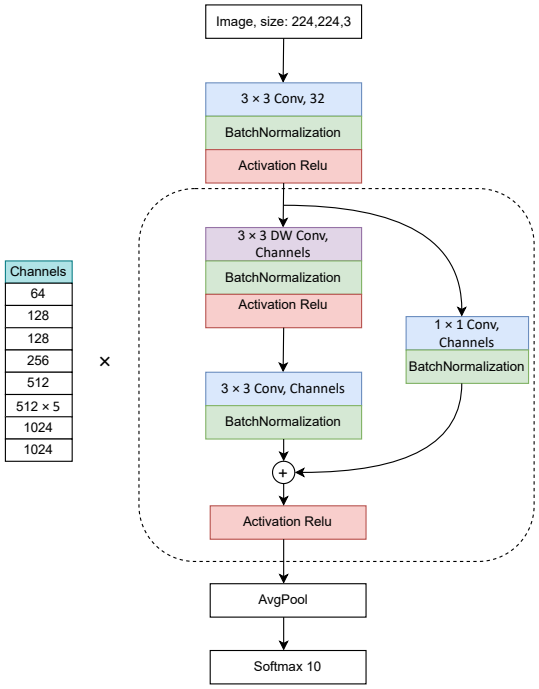

Fig. 2. Architecture of proposed CNN MMN

The proposed CNN MMN has the following architecture (Fig. 2):

- Input layer: the inputs are images with dimension (244×244).
- Convolutional layers: a convolution is a linear process that involves the input and a set of weights. It's made for two-dimensional inputs, with multiplication taking place between a two-dimensional array of weights (filters) and an array of input data. We have three sorts of layers in the proposed architecture MMN : one with a 3×3 size filter and the same padding, one with a 1×1 size filter and the same padding, and one with a 1×1 filter and valid padding.
- Depthwise separable convolution : This layer was employed in our architecture with the same padding and a 3×3 filter.
- Pooling layers: a technique for subsampling feature maps by aggregating the presence of features across feature map patches. Average pooling and maximal pooling are two different types of pooling algorithms. To calculate the average value in each patch for each feature map in the proposed design, we used avg-pooling.
- BatchNormalization: This method involves refocusing and rescaling layer inputs to normalize them and makes artificial neural networks faster and more stable.
- After each BatchNormalization, we employed Rectified Linear Unit Layers (ReLU).
- Fully connected layers: they treat input data as a single vector and produce a single vector as output.

IV. Results

A. The Used Dataset

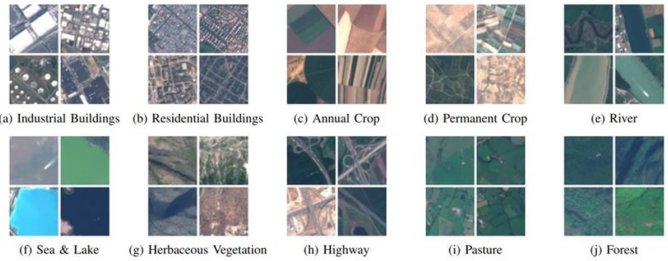

Fig. 3. Dataset classes [4]

TABLE I. EUROSAT DATA DISTRIBUTION

Class name	Number of images
Annual Crop	3000
Forest	3000
Herbaceous Vegetation	3000
Highway	2500
Industrial	2500
Pasture	2000
Permanent Crop	2500
Residential Buildings	3000
River	2500
Sea and Lake	3000

In this study, We used the EuroSAT database of Sentinel-2 images of differents European cities. It contains 27000 of 64×64 images. It covers several classes [4] i.e., Industrial Buildings, Residential Buildings, Sea and Lake, Herbaceous Vegetation, Annual Crop, Permanent Crop, River, Highway, Pasture, Forest (Fig. 3, Table I).

TABLE II. SENTINEL-2 BANDS, WAVELENGTH, AND RESOLUTION [4]

Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation Red Edge	0.705	20
Band 6 – Vegetation Red Edge	0.740	20
Band 7 – Vegetation Red Edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Vegetation Red Edge	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrus	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

Sentinel-2 data is multispectral, with 13 bands covering the visible, near-infrared, and shortwave infrared spectrum (Table II). Since these bands are available in various spatial resolutions ranging from 10 m to 60 m, the images can be classified as high-medium resolution. Although data from other satellites with higher resolution (1 m to 0.5 cm) is available. Sentinel-2 data is free and has a long revisit duration (5 days), making it a good choice for LULC monitoring.

B. Evaluation Metrics

After training the various architectures, the final step is to evaluate the performance of the used architectures. Among the various classification performance properties, our research uses the following benchmark metrics: Accuracy (ACC), Precision (PRE), Recall, F1-score (F1) [49].

– Accuracy :
Accuracy is the proportion of true results among the total number of cases examined.

$$ACC = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

– Precision:
Precision is the proportion of predicted positives that are actually positive.

$$PRE = \frac{(TP)}{(TP + FP)} \quad (2)$$

– Recall:
Recall is the proportion of actual positives that are correctly classified.

$$PRE = \frac{(TP)}{(TP + FN)} \quad (3)$$

– F1-score:
The F1-score is a number between 0 and 1 and is the harmonic mean of precision and recall.

$$PRE = 2 \times \frac{(precision \times recall)}{(precision + recall)} \quad (4)$$

Where: TP stands for: True Positive. FP: False Positive.TN: True Negative, and FN: False Negative.

C. Results without Data Augmentation

Our research was conducted on a publicly available image dataset of LULC images (Eurosat Dataset [4]) and on the same dataset using data augmentation. During the experimental study, 80% of images were used for training and 20% for testing and validation, and this operation was repeated 20 times randomly, the average of the results was retained. This using the following experimental parameters for classification: All images in the dataset were 64×64 pixels, with the exception of the proposed model image, which was scaled to 224×224 pixels. Thus for the data augmentation techniques, we trained the different models with an image size of 224×224. We used a batch size of 32 and a total of 20 epochs to train the models. For optimization, Adam is used with $\beta 1$ = 0.9, $\beta 2$ = 0.999, and the learning rate is set to 0.0001. As a result, a ReduceLROnPlateau with a min_lr of 1e-20 and val_accuracy as the monitor is employed. We utilized the ReLU to train a fully connected layer, and we changed the last dense layer in all models to yield 10 classes corresponding to the distinct classes in the Eurosat database [4], rather than the 1000 classes used by ImageNet. The results of the Eurosat image classification with the following architectures (Proposed Architecture, Fine–tuning the top layers of VGG16, ResNet50, DenseNet201, and MobileNet_V1) are shown in this section. Several experiments were carried out to evaluate the performance and resilience of each given model, based on the cited metrics in Section IV-B and the confusion matrix.

TABLE III. DIFFERENT METRICS OF THE PROPOSED ARCHITECTURE MMN

Metrics	Training	Testing
Accuracy	99.09%	94.01%
F1-Score	99.08%	94.02%
Recall	98.96%	93.9%
Precision	99.2%	94.8%
Loss	0.03	0.18

1) Proposed Architecture Modified MobileNet_V1 (MMN): According to Table III, we observe that the accuracy of the training data is remarkably high compared to testing, with values of 99.09% and 94.01% respectively, and this is the case for all the other metrics (F1-score, Recall, Precision). However, the loss metric has been significantly increased from training to testing with values of 0.03 and 0.18, respectively.

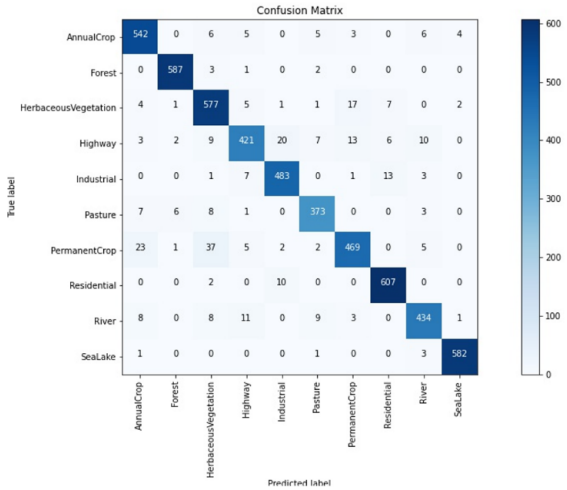

Fig. 4. Confusion matrix of proposed architecture MMN

From the confusion matrix (Fig. 4), we note that the most recognized class is “Residential” with 607 images. Therefore, for “Pasture” the model was able to identify just 373 images.

TABLE IV. DIFFERENT METRICS OF VGG16

Metrics	Training	Testing
Accuracy	100%	96.9%
F1-Score	100%	96.87%
Recall	100%	96.84%
Precision	100%	96.89%
Loss	2.8e-05	0.18

2) VGG16: Table IV shows the Accuracy, F1-score, Recall, Precision, and loss of VGG16. Indeed, the accuracy decreased from training to testing with a différence of -3.1%. The same was noticed in the other metrics except for the loss which increased from the value of 2.8e-05 for training to 0.18 for testing.

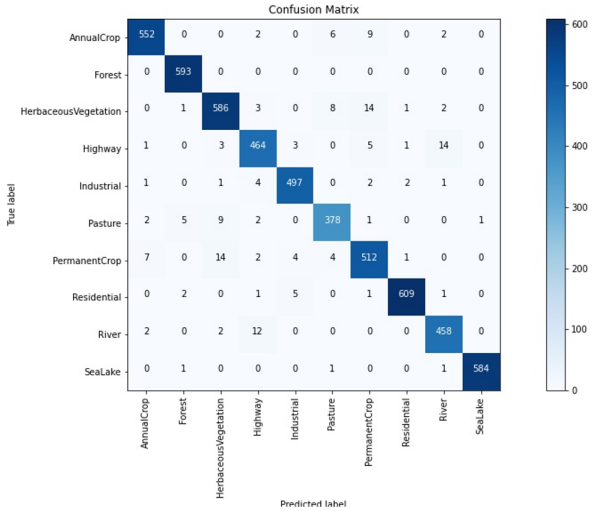

Fig. 5. Confusion matrix of VGG16

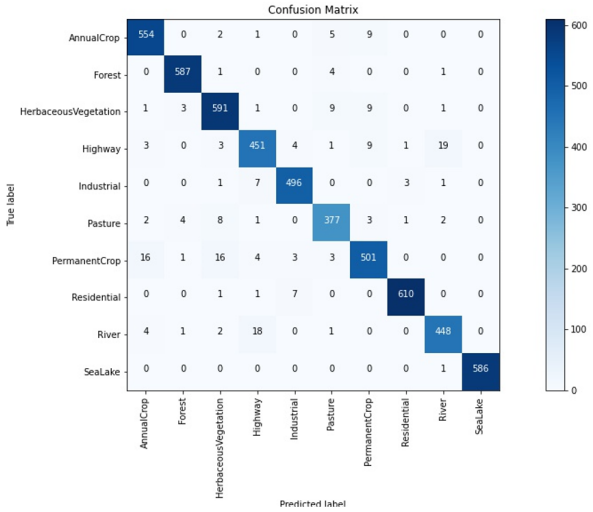

Fig. 6. Confusion matrix of ResNet50

Fig. 5 shows the confusion matrix of the VGG16 model, which illustrates that 609 images were correctly labeled as the "Residential" class, but only 378 images for "Forest" were classified by the model.

TABLE V. DIFFERENT METRICS OF RESNET50

Metrics	Training	Testing
Accuracy	100%	96.31%
F1-Score	100%	96.34%
Recall	100%	96.28%
Precision	100%	96.4%
Loss	1.55e-05	0.17

3) ResNet50: Table V presents the obtained results with the Resnet50 classifier. We observe an increase in the accuracy of the training compared to the testing where its values are equal to 100%, 96.31% respectively, and it is the same for the other metrics. The value of F1-score for the training is equal to 100%, while for the testing it is 96.34%. We also observe that the recall and the precision have increased from the training (100%, 100%, respectively) in comparison with the testing (96.28%, 96.4%, respectively). For the loss function, its value is equal to 1.55 e-05 for training and 0.17 for testing.

As shown in Fig. 6, the most identifiable class is "Residential", whereas the model was capable to predict only 377 images as "Pasture".

TABLE VI. DIFFERENT METRICS OF MOBILENET_V1

Metrics	Training	Testing
Accuracy	99.91%	76.31%
F1-Score	99.89%	76.37%
Recall	99.87%	74.96%
Precision	99.91%	77.88%
Loss	0.01	0.9

4) MobileNet_V1: Table VI illustrates the obtained results of MobileNet_V1. The table shows a remarkable difference between the results of the different metrics for the training data compared to the testing data with values of 99.91%, 99.89%, 99.87%, 99.91% and 0.01 (Accuracy, F1-score, Recall, Precision, and Loss, respectively) for the training and for the testing (76.31%, 76.37%, 74.96%, 77.88% and 0.9). According to these results we can distinguish that the model overfit.

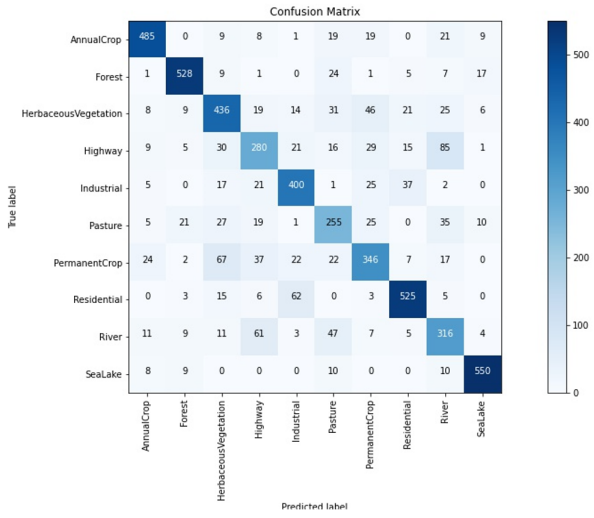

Fig. 7. Confusion matrix of MobileNet_V1

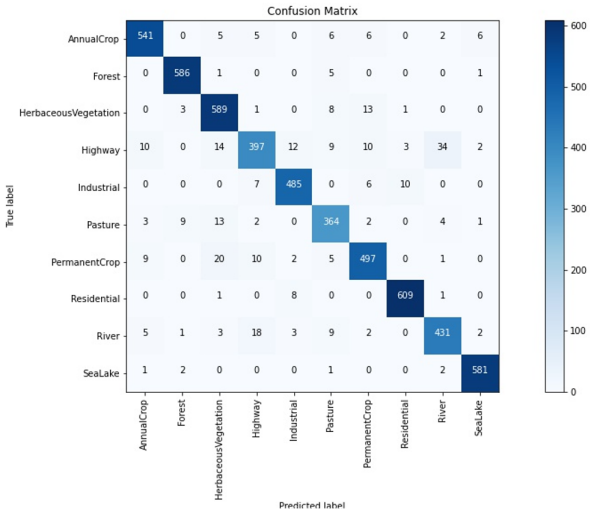

Fig. 8. Confusion matrix of DenseNet201

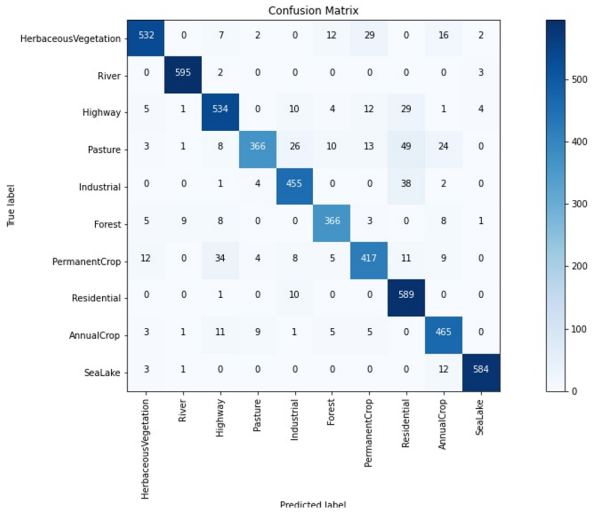

Fig. 9. Confusion matrix of proposed CNN MMN

TABLE VIII. THE RESULTS OF THE PROPOSED CNN ARCHITECTURE MMN WITH DATA AUGMENTATION

Metrics	Training	Testing
Accuracy	91.89%	90.79%
F1-Score	91.89%	90.79%
Recall	90.52%	89.75%
Precision	93.34%	91.91%
Loss	0.23	0.27

The confusion matrix (Fig. 7) indicates that the model can correctly predict 550 images of the "SeaLake" class and just 255 images of the "Pasture" class.

TABLE VII. DIFFERENT METRICS OF DENSENET201

Metrics	Training	Testing
Accuracy	100%	94.07%
F1-Score	100%	94.11%
Recall	100%	93.98%
Precision	100%	94.24%
Loss	8.72e-05	0.25

5) DenseNet201: As shown in Table VII, we can observe that accuracy, F1-score, recall and precision have remarkably high values for training (100%, 100%, 100%, 100%) compared to testing (94.07%, 94.11%, 93.98%, 94.24%), as well as for loss which has decreased from testing (0.25) to training (8.72e-05).

Concerning the confusion matrix (Fig. 8), the model was able to correctly identify 609 images of the class "Residential", however only 364 images were correctly labeled as "Pasture".

D. Results with Data Augmentation

In this section, we present the different results of the trained models, but this time using data augmentation techniques on the same dataset.

1) Proposed Architecture Modified MobileNet_V1 (MMN): According to Table VIII, we can see that the different metrics do not vary remarkably from training to testing. There is a small decrease in the accuracy, the F1-score, the recall, and the precision. However, the loss function was increased from 0.23 in training to 0.27 in testing.

TABLE IX. DIFFERENT METRICS OF VGG16

Metrics	Training	Testing
Accuracy	96.87%	96.01%
F1-Score	96.87%	96.09%
Recall	96.49%	95.71%
Precision	97.26%	96.5%
Loss	0.09	0.12

TABLE X. DIFFERENT METRICS OF RESNET50

Metrics	Training	Testing
Accuracy	99.97%	97.59%
F1-Score	99.97%	97.61%
Recall	99.97%	97.59%
Precision	99.97%	97.63%
Loss	0.001	0.17

From the confusion matrix (Fig. 9), we note that the model correctly recognizes 595 images of the class "River",the model was also able to identify only 366 images of "Forest" and "Pasture".

2) VGG16: Table IX shows the accuracy, f1-score, recall, precision and loss of the VGG16 model on the training data and on the testing data. We notice that for all the metrics, there is an insignificant difference between training and testing.

Concerning the confusion matrix (Fig. 10), the model was able to classify 600 images of "Residential" as well as 367 images of the "Forest" class.

3) ResNet50: As shown in Table X, the different evaluation metrics decreased from training to testing, with values of more than 97% for testing and more than 99% for training. The loss function showed a remarkable increase between the two cases with 0.001 for training data and 0.17 for testing data.

For the confusion matrix (Fig. 11), the most recognized class by ResNet50 is the "SeaLake" class with 599 images,

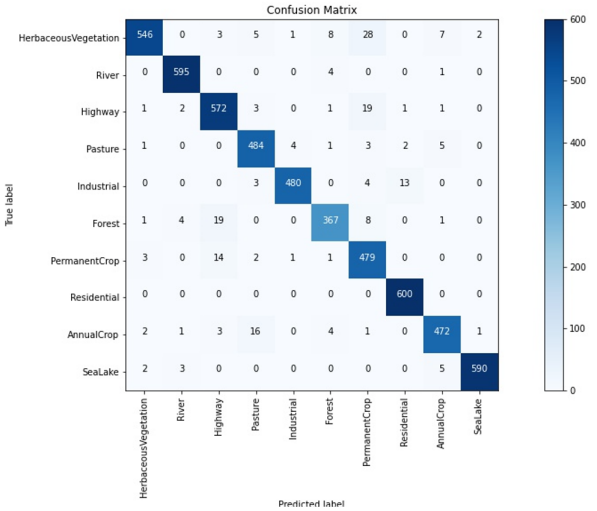

Fig. 10. Confusion matrix of VGG16

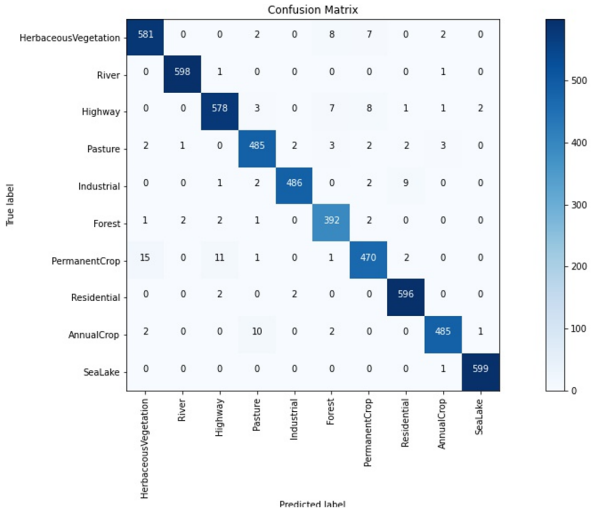

Fig. 11. Confusion matrix of ResNet50

TABLE XI. DIFFERENT METRICS OF MOBILENET_V1

Metrics	Training	Testing
Accuracy	97.18%	96.59%
F1-Score	97.17%	96.64%
Recall	97.03%	96.56%
Precision	97.32%	96.73%
Loss	0.08	0.12

therefore the least classified class is the “Forest” with just 392 images.

4) MobileNet_V1: Table XI represents the evaluation result of MobileNet_V1 architecture. It shows that there is a small difference between training and testing i.e. accuracy, f1-score, recall, precision and loss.

As shown in Fig. 12, the model was able to recognize correctly 596 images of the class “River” and only 375 for the class “Forest”.

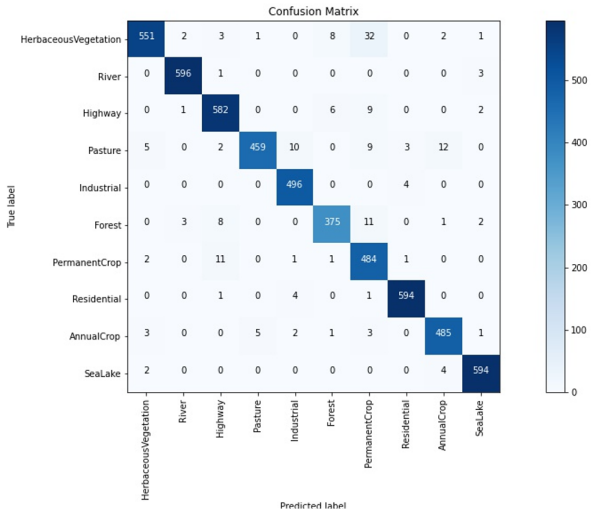

Fig. 12. Confusion matrix of Mobilenet_V1

TABLE XII. DIFFERENT METRICS OF DENSENET201

Metrics	Training	Testing
Accuracy	98.77%	97.24%
F1-Score	98.74%	97.27%
Recall	98.62%	98.18%
Precision	98.87%	97.36%
Loss	0.03	0.09

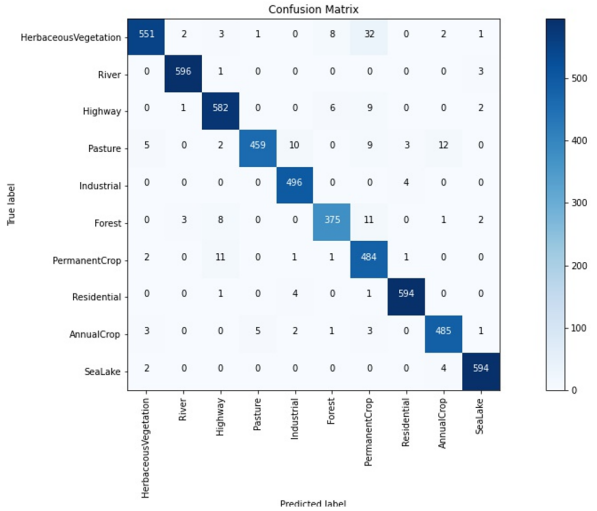

Fig. 13. Confusion matrix of DenseNet201

5) DenseNet201: All metrics have a minimal variation between training and testing, as shown in Table XII, with the exception of loss, which has a small increase between training and testing.

Fig. 13 shows that 593 images were correctly marked as “SeaLake” and “River”, but the model was able to identify just 389 images of “Forest”.

E. Results on Real Images

The satellite image was acquired in the surroundings of Berrechid, Morocco, in May 2022, we took two patches from it to test several classifiers.

Fig. 14. The results of the different models

The Fig. 14 shows the results of the different models on two real images. We notice that for the proposed architecture MMN, the model gave as classification for the first image "Herbaceous Vegetation" and for the second image "Residential". For VGG16 and ResNet50, we got "Highway" for the first image which is clearly wrong classification, and "Industrial" for the second, while MobileNet_V1 has classified the first and second image as "AnnualCrop", which can not be possible for the second one. Finally, for DenseNet201, the model gave an erroneous classification for the first image (SeaLake) and classified the second as "Industrial".

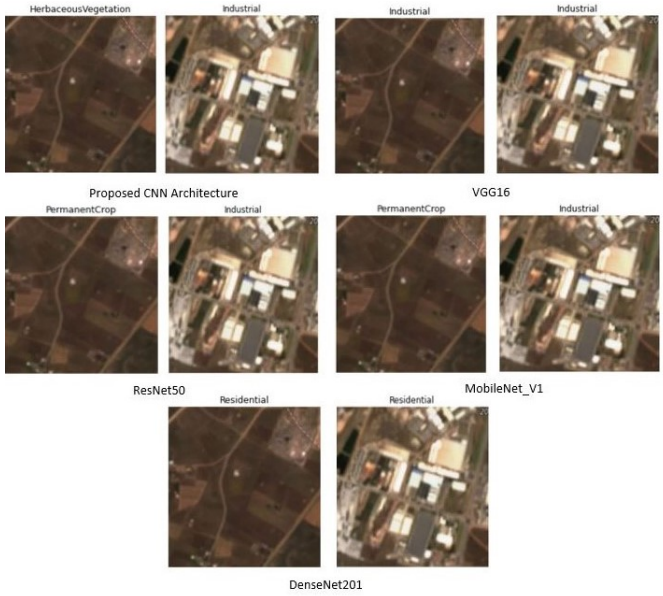

Fig. 15. The results of the different models with data augmentation

The obtained results with data augmentation are shown in the Fig. 15. We can observe that the proposed method identified the first image as "Herbaceous Vegetation" and the second as "Industrial", while VGG16 classified the 2 images as "Industrial" which can't be true for the first image. For ResNet50 and MobileNet_V1, they recognized the first image as "Permanent Crop" and the second as "Industrial". DenseNet201 identified the first and second image as "Residential", it gave wrong classification for the first image.

V. Discussion

In this study, we addressed the multiclass classification of the LULC domain, based on the Eurosat dataset [4]. The training was applied using transfer learning, in order to identify the best architecture. The experimental study proved that the neural networks architectures are very useful for LULC image classification since they have shown high performance.

TABLE XIII. Evaluations Metrics of Different Architectures

Models	Accuracy	F1-Score	Recall	Precision	Loss	parameters number
MMN	94.01%	94.02%	93.9%	94.8%	0.18	6402506
VGG16	96.9%	96.87%	96.84%	96.89%	0.18	15768906
ResNet50	96.31%	96.34%	96.28%	96.4%	0.17	27787658
MobileNet-_V1	76.31%	76.37%	74.96%	77.88%	0.9	3239114
DenseNet-201	94.07%	94.11%	93.98%	94.24%	0.25	22259786

Table XIII compares the performance of each architecture based on the mentioned metrics in Section IV-B, and the number of parameters. From the results, it can be seen that accuracy, F1-score, recall and precision when using the

MobileNet_V1 model are remarkably lower than the other models, with values of 76.31%, 76.37%, 74.96% and 77.88%, respectively, and this is due to the minimum number of parameters (3.2 M). Furthermore, the DenseNet201 and the proposed model showed the same performance except for the loss function, which has a value of 0.18 for the proposed architecture and 0.25 for DenseNet201. This with a too small number of parameters (6.4 M) in comparison with DenseNet201 (18.35 M), It is due to the fact that the extracted features from the previous layer have been added to the next layer. The main advantage of our architecture is that it needs a small number of parameters.

Both the VGG16 and ResNet50 architectures perform well, with 96% accuracy, F1-score, recall, and precision, which is expected considering ResNet50's depth and VGG16's usage of small filters. It can be seen that the MMN architecture is almost as accurate compared to the other architectures while being smaller in terms of parameters.

TABLE XIV. EVALUATIONS METRICS OF DIFFERENT ARCHITECTURES WITH DATA AUGMENTATION

Models	Accuracy	F1-Score	Recall	Precision	Loss	parameters number
MMN	90.79%	90.79%	89.75%	91.91%	0.27	6402506
VGG16	96.01%	96.09%	95.71%	96.5%	0.12	15768906
ResNet50	97.59%	97.61%	97.59%	97.63%	0.17	27787658
MobileNet-_V1	96.59%	96.64%	96.56%	96.73%	0.12	3239114
DenseNet-201	97.24%	97.27%	97.18%	97.36%	0.09	22259786

Table XIV shows the obtained results of the different architectures with the use of data augmentation. Based on the presented results, ResNet50 and DenseNet201 showed good performance with an accuracy of more than 97%. Thus, we notice that data augmentation has improved the results for both architectures. Then we have the MobileNet_V1 and VGG16 with an accuracy of more than 96%. We also have for the loss function, 0.09 for the DenseNet201 model, 0.12 for VGG16 and MobileNet_V1, and an increase for ResNet50 with 0.17. Therefore, we can see that MobileNet_V1 with a lower number of parameters was able to achieve such remarkable results compared to the other models. Generally, this is expected since data augmentation is one of the best techniques for reducing overfitting. Table XIV shows that the MMN architecture was able to achieve an accuracy of 90.79% which is a reduced value compared to other architectures. This is most likely due to the model's limited capacity, which prevents it from learning all of the patterns in the data, or because some architectures are very sensitive to data augmentation.

VI. CONCLUSION AND PERSPECTIVES

In this work, we evaluated the performance of the automated methods used to classify LULC images into 10 classes on the Eurosat dataset using four DL architectures (VGG16, ResNet50, MobileNet_V1 and DenseNet201) and a proposed CNN architecture MMN based on a combination of the two methods MobileNet_V1 and ResNet50. Comparing the results, we found that our architecture proved good performance but performs poorly using data augmentation compared to the other architectures. Therefore, our model needs to be adjusted to get better performance on the dataset with data augmentation. We also found that the performance of the DenseNet201 model and the proposed architecture is equivalent but the proposed model involves fewer parameters, which means it requires less memory consumption. In the experiment, we found that with and without the use of data augmentation, the VGG16 and ResNet50 models perform well (accuracy greater than 96%) compared to the other architectures. Moreover, we noticed that MobileNet_V1 can efficiently classify LULC images in the case of data augmentation, but with the original dataset its results were poor.

The proposed model will be improved by applying parameter tuning and studying the loss function. In addition, the used dataset in the training phase has a high impact on the efficiency of the model, thus it is very important to test the model with another dataset. Since each model has its own method to extract features, which highly influences the classification accuracy, we aim to study deeply the layers of feature extraction in order to improve the proposed model efficiency.

ACKNOWLEDGMENTS

This research is supported by National Institute of Statistics and Applied Economics, Rabat, Morocco.

REFERENCES

[1] X. Jiang, Y. Wang, W. Liu, S. Li, and J. Liu, "Capsnet, cnn, fcn: Comparative performance evaluation for image classification," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 840–848, 2019.

[2] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 558–567.

[3] M. I. Lakhal, H. Çevikalp, S. Escalera, and F. Ofli, "Recurrent neural networks for remote sensing image classification," *IET Computer Vision*, vol. 12, no. 7, pp. 1040–1045, 2018.

[4] 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 Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[5] S. Mubako, O. Belhaj, J. Heyman, W. Hargrove, and C. Reyes, "Monitoring of land use/land-cover changes in the arid transboundary middle rio grande basin using remote sensing," *Remote Sensing*, vol. 10, no. 12, p. 2005, 2018.

[6] C. Zhang, P. A. Harrison, X. Pan, H. Li, I. Sargent, and P. M. Atkinson, "Scale sequence joint deep learning (ss-jdl) for land use and land cover classification," *Remote Sensing of Environment*, vol. 237, p. 111593, 2020.

[7] S. E. Jozdani, B. A. Johnson, and D. Chen, "Comparing deep neural networks, ensemble classifiers, and support vector machine algorithms for object-based urban land use/land cover classification," *Remote Sensing*, vol. 11, no. 14, p. 1713, 2019.

[8] S. Talukdar, P. Singha, S. Mahato, S. Pal, Y.-A. Liou, A. Rahman *et al.*, "Land-use land-cover classification by machine learning classifiers for satellite observations—a review," *Remote Sensing*, vol. 12, no. 7, p. 1135, 2020.

[9] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

[10] 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 sensing of environment*, vol. 221, pp. 173–187, 2019.

[11] R. Nijhawan, D. Joshi, N. Narang, A. Mittal, and A. Mittal, "A futuristic deep learning framework approach for land use-land cover classification using remote sensing imagery," in *Advanced computing and communication technologies*. Springer, 2019, pp. 87–96.

[12] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015.

[13] K. Nogueira, O. A. Penatti, and J. A. Dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[14] O. A. Penatti, K. Nogueira, and J. A. Dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 44–51.

[15] 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 Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[16] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.

[17] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS journal of photogrammetry and remote sensing*, vol. 145, pp. 197–209, 2018.

[18] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[19] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: a learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, 2015, pp. 1–10.

[20] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2889–2901, 2017.

[21] G. Cheng, J. Han, L. Guo, and T. Liu, "Learning coarse-to-fine sparselets for efficient object detection and scene classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1173–1181.

[22] Q. Yuan, H. Shen, T. Li, Z. Li, S. Li, Y. Jiang, H. Xu, W. Tan, Q. Yang, J. Wang *et al.*, "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sensing of Environment*, vol. 241, p. 111716, 2020.

[23] A. Panda, A. Singh, K. Kumar, A. Kumar, A. Swetapadma *et al.*, "Land cover prediction from satellite imagery using machine learning techniques," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2018, pp. 1403–1407.

[24] M. M. Aburas, M. S. S. Ahamad, and N. Q. Omar, "Spatio-temporal simulation and prediction of land-use change using conventional and machine learning models: a review," *Environmental monitoring and assessment*, vol. 191, no. 4, pp. 1–28, 2019.

[25] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and remote sensing magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[26] 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 transactions on geoscience and remote sensing*, vol. 56, no. 5, pp. 2811–2821, 2018.

[27] G. Sumbul and B. Demİr, "A deep multi-attention driven approach for multi-label remote sensing image classification," *IEEE Access*, vol. 8, pp. 95 934–95 946, 2020.

[28] D. Koßmann, T. Wilhelm, and G. A. Fink, "Towards tackling multi-label imbalances in remote sensing imagery," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5782–5789.

[29] A. K. Aksoy, M. Ravanbakhsh, T. Kreuziger, and B. Demir, "A novel uncertainty-aware collaborative learning method for remote sensing image classification under multi-label noise." *CoRR, vol. abs/2105.05496*, 2021.

[30] I. Kakogeorgiou and K. Karantzalos, "Evaluating explainable artificial intelligence methods for multi-label deep learning classification tasks in remote sensing," *International Journal of Applied Earth Observation and Geoinformation*, vol. 103, p. 102520, 2021.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[32] U. Chaudhuri, S. Dey, M. Datcu, B. Banerjee, and A. Bhattacharya, "Interband retrieval and classification using the multilabeled sentinel-2 bigearthnet archive," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9884–9898, 2021.

[33] P. Aggarwal, N. K. Mishra, B. Fatimah, P. Singh, A. Gupta, and S. D. Joshi, "Covid-19 image classification using deep learning: Advances, challenges and opportunities," *Computers in Biology and Medicine*, p. 105350, 2022.

[34] W. Liu, C. Li, M. M. Rahaman, T. Jiang, H. Sun, X. Wu, W. Hu, H. Chen, C. Sun, Y. Yao *et al.*, "Is the aspect ratio of cells important in deep learning? a robust comparison of deep learning methods for multi-scale cytopathology cell image classification: From convolutional neural networks to visual transformers," *Computers in biology and medicine*, vol. 141, p. 105026, 2022.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[38] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv preprint arXiv:1803.01164*, 2018.

[39] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 5, pp. 4340–4354, 2020.

[40] P. C. Pandey, N. Koutsias, G. P. Petropoulos, P. K. Srivastava, and E. Ben Dor, "Land use/land cover in view of earth observation: data sources, input dimensions, and classifiers—a review of the state of the art," *Geocarto International*, vol. 36, no. 9, pp. 957–988, 2021.

[41] P. Kim, "Convolutional neural network," in *MATLAB deep learning*. Springer, 2017, pp. 121–147.

[42] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[44] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 169–175.

[45] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 1011–1014.

[46] K. Cai, X. Miao, W. Wang, H. Pang, Y. Liu, and J. Song, "A modified yolov3 model for fish detection based on mobilenetv1 as backbone," *Aquacultural Engineering*, vol. 91, p. 102117, 2020.

[47] S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through

glass diffusers using densely connected convolutional networks," *Optica*, vol. 5, no. 7, pp. 803–813, 2018.

[48] O. Mohamed, E. A. Khalid, O. Mohammed, and A. Brahim, "Content-based image retrieval using convolutional neural networks," in *First International Conference on Real Time Intelligent Systems*. Springer, 2017, pp. 463–476.

[49] M. Fatourechi, R. K. Ward, S. G. Mason, J. Huggins, A. Schloegl, and G. E. Birch, "Comparison of evaluation metrics in classification applications with imbalanced datasets," in *2008 seventh international conference on machine learning and applications*. IEEE, 2008, pp. 777–782.

APPENDIX

APPENDIX A. TERMS AND THEIR ABBREVIATIONS

LULC: Land Use and Land Cover.
DL: Deep Learning.
CNN: Convolutional Neural Network.
GIS: Geographic Information System.
MMN: Modified MobileNet_V1.