Slum image detection and localization using transfer learning: a case study in Northern Morocco

Tarik El Moudden[1], Rachid Dahmani[2], Mohamed Amnai[1], Abderrahmane Aït Fora[2]
[1]Computer Science Research Laboratory, Ibn Tofail University, Kenitra, Morocco
[2]Geoscience Laboratory, Ibn Tofail University, Kenitra, Morocco

Article Info

Article history:

Received Jun 23, 2022
Revised Sep 13, 2022
Accepted Oct 1, 2022

Keywords:

Convolution neural network
Deep learning
Machine learning
Remote sensing
Slums
Transfer learning

ABSTRACT

Developing countries are faced with social and economic challenges, including the emergence and proliferation of slums. Slum detection and localization methods typically rely on regular topographic surveys or on visual identification of high-resolution spatial satellite images, as well as socio-environmental surveys from land surveys and general population censuses. Yet, they consume so much time and effort. To overcome these problems, this paper exploits well-known seven pretrained models using transfer learning approaches such as MobileNets, InceptionV3, NASNetMobile, Xception, VGG16, EfficientNet, and ResNet50, consecutively, on a smaller dataset of medium-resolution satellite imagery. The accuracies obtained from these experiments, respectively, demonstrate that the top three pretrained models achieve 98.78%, 97.9%, and 97.56%. Besides, MobileNets have the smallest memory sizes of 9.1 Mo and the shortest latency of 17.01 s, which can be implemented as needed. The results show the good performance of the top three pretrained models to be used for detecting and localizing slum housing in northern Morocco.

Corresponding Author:

Tarik El Moudden
Computer Science Research Laboratory, Ibn Tofail University
Residence Adam, App 1 Bloc D Ext 4 and 5, Maghrib Arabi, Kenitra, Morocco
Email: tarik.elmoudden@uit.ac.ma

1. INTRODUCTION

The collected data concerning cities without slums (CSW) is worrying. The program was supposed to eradicate unplanned settlements such as shanty-towns, slums in Morocco during the period 2008 to 2018. the Minister of National Territorial Planning, Urban Planning, Housing and City Policy revealed in 2021 that this program did not achieve its intended goals. In addition, the Minister confirmed on January 27, 2021 that 152,000 families had not been counted [1]. As a result, they had not been added to the dataset of 26 cities that have not been declared cities without slums yet. This kind of settlement area is often filled with high-density small homes, constructed with temporary materials like sheet metal and other recyclable materials [2], and their existence is a violation of human rights as they are deprived of several essential decent living elements such as access to drinking water, sanitation, sufficient living spaces and security.

These issues have become challenges for the government in terms of developing programs that might eradicate all the slums in urban areas using resettlement, rehousing, and restructuring [1]. Sadly, despite all of our administrative agencies' efforts, slum housing is constantly expanding. Some of the time, this was because citizens preferred slum living to bettering their financial situation, rural-urban migration, or just plain poverty. Furthermore, the expanding shanty-towns may have a devastating impact on the health of their citizens and sustainable development, such as the rise in pandemics and air pollution [3].

Over the last 10 years, deep learning, specifically convolution neural networks (CNNs), has become an important tool that can study, analyze, and identify remote sensing (RS) images. Several researchers have built various architectures to improve the computation accuracy of convnets, which can do even better than human capabilities in many real-world applications and studies. The history of victories of convnets started in 2012 with the AlexNet architecture [4], which won the ImageNet [5], ImageNet large scale visual recognition challenge (ILSVRC) challenge with top-1 and top-5 error rates of 37.5% and 17.0%. The Convnets continued to evolve towards VGG16 in 2014 [6], GoogLeNet in 2014 [7], ResNet in 2015 [8], Inceptionv3 in 2015 [9], Xception in 2017 [10], MobileNets in 2017 [11], NASNetMobile in 2018 [12], and EfficientNetB0 in 2019 [13].

In the field of land cover/use classification, including slum housing detection and localizing from satellite imagery, there are several studies that used convnets through different approaches to classify very high resolution (VHR) images. On one hand, many studies have found an overuse of the transfer learning (TL) approach in the field of RS due to small labeled training samples [14]. On the other hand, others escaped from designing and training deep networks from scratch due to their time-consumption, complicated computation process, and uncertain outcomes.

Compared with classification and detection tasks, a new trend called fully convolutional network (FCN) has emerged, which uses mostly an encoder-decoder architecture to produce pixel-by-pixel segmentation prediction. The main objective of FCN design is to segment and map slum housing changes and bondaries more efficiently from VHR images. Consequently, some studies developed several types of FCN, such as: the U-Net architecture [15], TL FCN-AlexNet, FCN-VGG16 and FCN-GoogLeNet networks [16].

Furthermore, when training a pretrained convnets, it is necessary to minimize the loss function and maximize the accuracy. Therefore, several researchers in different domains developed a computational method to optimize a problem. Three of them attract more attention, such as: metaheuristic particle swarm optimizations (PSO) [17], [18], improved variation of bat algorithm (BA) [19] and several types of Tensorflow Keras gradient descent. To our knowledge, most pretrained convnets use variants of gradient descent to change iteratively the weight and bias of each layer to reach the global minima in the field of remote sensing imagery.

The objective of this study is to identify the top pretrained models for automatically locating slum houses in northern Morocco and produce slum maps to study their evolution and change. To do this, we begin by gathering medium-resolution slum patches of the size (448,448,3) using Pro Google Earth, then we perform data preprocessing of the images. Finally, we investigate and apply a variety of pretrained convnets, employing TL approaches. This study is organized: previous work is presented in section 2. Methods and materials are tackled in section 3. Results and discussion are in section 4. Conclusion and recommendations are provided in section 5.

2. PREVIOUS WORK

In this section, we will try to shed light on the previous research related to our study in order to show both the advantages and drawbacks of each and every approach. In so doing, we have structured this section into three parts. The first part deals with machine learning while the second part tackles transfer learning and the last part explains fully CNN.

2.1. Machine learning

Over the last decade, satellite remote sensing has become indispensable in localization and segmentation of urban buildings, which is not expensive compared to ground or airborne sensor acquisitions [20]. Slum mapping-related remote sensing imagery via satellite has become one of the best approaches to begin generating algorithms and models to detect, classify, and delimit the shape of slums [21]. In the past, there was a method called object-based image analysis (OBIA). Still in use, especially in higher resolution imagery in capturing slum settlements, OBIA can have many drawbacks. Two of them will be explained. Firstly, the spectral information provided is insufficient in capturing enough details regarding the clear differentiation between different land patterns in cities. Secondly, the similarity between the roof textures of slums and other buildings can lead to false classification [22].

The logistic regression is applied in hyperspectral image classification by Li *et al.* [23], who included spatial information with spectral information in the hope that it would improve the performances, but the logistic regression still provides poor classification and segmentation performance. The same results are obtained depending on the support vector machines (SVMs) and K-nearest neighbors (KNNs) algorithms [24], which can provide an accuracy between 60.2% and 88%. The SVM [25], a multi-class classifier, was trained on the tiles extracted from geo-referenced raster maps with borders outlining the different slums existing in Kalyan, Dombivli, and Bangalore city, and the classifier performed an overall accuracy of 66% to

93% depending on different tile sizes and vocabulary sizes [26]. The random forest (RF) classifier [27] achieved an overall accuracy of 79% to 86.5% depending on varying texture window sizes, and it proved capable of following the slum patch outline geometries more precisely [28], compared to a linear discriminant classifier.

2.2. Transfer learning

Deep learning is a subfield of artificial intelligence (AI). It learns automatically from multiple levels of data abstraction [29], without having to be told explicitly which features to use and how to extract them. In the domain of image classification, and object detection [30], convnets, as the state of the art in terms of accuracy, have a strong ability to extract simple and complex features that express the image more broadly and learn specific features that are much more efficient. As the depth of convolutional networks goes up to 19 weight layers for large-scale images, the classification accuracy increases. Therefore, the model generalizes well-used discriminative semantic features compared to mid-level approaches [6], [29].

However, if we choose the strategy to train a network from scratch, despite the difficulties in the choice and use of appropriate architecture, we will observe many experimental drawbacks including the fluctuations in the validation accuracy and validation loss curve. This is due to the very small number of samples in the training dataset [31]. Herein, lies the necessity of the TL approaches, in which a model that is trained on a certain source domain DS (ImageNet) to achieve learning task ƮS (classification of 1,000 objects) can be used to make predictions for a target task based on the target domain DT through the optimization of an objective predictive function fT(.) in DT, where DS≠DT, ƮS≠TT , as described by [32].

In military object recognition [33], the authors train special convnets with ImageNet and then transfer the learned features to the small military dataset to recognize 16 military vehicles with high precision. In [22], the Inception-based TL model can detect informal slums with an overall accuracy of 94.2% and 90.2% and kappa of 70%, and 55% is obtained from very high resolution (VHR) and medium resolution (MR) imagery, respectively. Another approach, combining triplet deep metric learning networks (TDMLN) is proposed in [34] to build effective and efficient image retrieval tools. For each network, the pretrained model on the ImageNet dataset: AlexNet, VGG16, and ResNet50, are used to extract features.

Ajami *et al.* [35] take the TL away from pretrained models on ImageNet for quantifying deprivation with a data-driven index of multiple deprivation (DIMD). First, they trained shallow convnets on 1,461 samples to classify “slums” from “formal areas”. Second, they transform the learned features into regression model dealing with 121 samples with known DIMD to predict the degree of deprivation with R-squared of 0.75.

However, in [36], [37], the negative transfer (NT) risk is among the biggest threats to the exploitation and development of TL of pretrained models on huge datasets such as ImageNet. The bad effects of this risk occur when the transfer of knowledge between the source and target domains is low-quality. Moreover, this risk can be addressed through a variety of methods, including training convnets from scratch directly on remote sensing data [38], or in [39], leveraging some similarity between the source and target domains, using data transferability enhancement, model transferability enhancement, and target prediction enhancement.

2.3. Fully convolution neural network

In several studies, for semantic segmentation, we find two different artificial neural network architectures: U-Net and fully convolutional networks (FCN). Pretrained U-net architecture [40] reaches an overall accuracy of over 86% for building segmentation and over 83% for classification, outperforming the widely used RF and OBIA. FCN-VGG19 in [41], on Quickbird to Sentinel-2 (FCN-TL S2) and TerraSAR-X (FCN-TL TX) for slum segmentation in satellite imagery improves significantly the segmentation of intersection-over-union (IoU) to 87.43% for FCN-TL S2, and to 73.02% for FCN-TL TX. In [38], which adapts FCN to the TL for slum mapping, has used two approaches to study the effects of an imbalanced dataset on intersection over union. The first approach utilizes the VGG19 pretrained on ImageNet to adjust the FCN weights. The second one employs the TL from other RS data. The results show the first approach produces higher accuracy measures than the second approach. Furthermore, the performances are dependent on the slum sample proportion and the number of trained images. In the paper [39], Stark *et al.* proposed using TL with fully convolutional using Xception as the backbone network (Xception-FCN), to build large-scale slum maps. The Xception-FCN is trained from scratch on datasets provided by many cities that host various categories of slums. The approach proposed can achieve F1 scores of up to 89%.

3. METHODS AND MATERIALS

In this paper, we have conducted a comparative study to find pretrained models suitable in terms of high accuracy, reduced fluctuation during training, avoiding negative transfer learning, high latency, and low memory model size. To achieve this goal, we first discuss the pretrained model characteristics. Building a slum dataset comes second. Data preprocessing comes in third. The pretrained models’ implementation and software environment come last.

3.1. Pretrained model characteristics

3.1.1. MobileNets

MobileNets, a class of light-weight deep CNNs built on depthwise factorization separable filters, achieve 89.5% Top-5 Accuracy with 4.3 M parameters [11]. This factorization has the effect of drastically reducing latency and model size that can be easily implemented in mobile or executed in vision applications. The MobileNet model is based on depthwise separable convolutions, which is a form of factorized convolutions, factorizing a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. With the computational cost of depthwise and pointwise convolution in Table 1, we obtain the ratio, as in (1). This ratio demonstrates the MobileNet using 3×3 depthwise separable convolutions, ranging from 8 to 9 times less computation than standard convolutions with only a small reduction in accuracy [11].

$$Ratio = \frac{M\times D_G^2\times\left(D_K^2+N\right)}{N\times D_G^2\times D_K^2\times M} = \frac{D_K^2+N}{D_K^2\times N} = \frac{1}{N}+\frac{1}{D_K^2} \tag{1}$$

Table 1. The computational cost of MobileNet model vs standard convolutional

	Depthwise and Pointwise Convolution	Standard Convolutional
Computational cost	$=(D_K\times D_K\times D_G\times D_G\times M)+(D_G\times D_G\times M\times N=M\times D^2{}_G\times(D^2{}_K+N)$	$=N\times D^2{}_G\times D^2{}_G\times M$

3.1.2. EfficientNetBO

To reach better accuracy, most strategies scale up CNNs arbitrarily and often provide unsatisfactory accuracy and efficiency. To address this problem in a systematic and comprehensive way, Google released a new family of EfficientNet in 2019, which is achieving state-of-the-art in the top-1 accuracy on ImageNet. Tan and Le [13], use neural architecture search (NAS) to build an efficient network architecture called EfficientNetB0. It achieves 77.3% accuracy on ImageNet with only 5.3 M parameters and 0.39 B FLOPS, and then scales depth (the number of layers), width (the number of map features), and resolution (image size) using a balanced compound coefficient. The latter was done in order to attain EfficientNet family (EfficientNetB1) to EfficientNetB7) state-of-the-art accuracy on ImageNet compared to other TL models. We are going to use the EfficientNetB0 architecture as it is the least complex and it works on images with less computationally powerful hardware compared to the EfficientNet family.

3.1.3. NASNetMobile

The NAS developed at Google Brain has put a lot of the challenge into the field of finding the best model in terms of accuracy and error rate compared to those of the state-of-the-art human-designed models [12]. For instance, the reduced version of NASNet is called NASNetMobile. It achieves 74.4% top-1 accuracy with 5.3M parameters, which is 3.1% better than the equivalently-sized ones. The NAS approach uses a search strategy based on the reinforcement learning method to optimise architecture configurations chosen from the cell-based search space that specifies the architecture of each cell. In addition to this, it is composed of two types of repeated modules: “normal cells” and “reduction cells”.

3.1.4. Xception

With 22.9 M parameters, the Xception achieves 94.5% Top-5 accuracy. It was proposed by the creator of Keras, Chollet [10]. He replaces Inception modules with depthwise separable convolutions followed by a pointwise convolution (1×1 convolution) that requires fewer parameters and fewer computations. The Xception module has three main parts: the entry flow, the middle flow (which is repeated 8 times), and the exit flow. These three parts use the following layers: batch normalization after convolution and separable convolution layers, max Pooling layers with stride of 2×2 or 1×1, skip connections with ADD operation.

3.1.5. Inceptionv3

The increase in the depth and the width of the neural networks (NNs) has many drawbacks, three of which are to be explained as [9]: i) there are a larger number of parameters that lead to overfitting, ii) uniform increases in the filters of convolutional layers result in expensive computation, and iii) by choosing the smaller kernel size for clusters, it concentrates locally. Conversely, by choosing larger kernels for clusters, NNs becomes more globally distributed.

To overcome these drawbacks, the architecture performs convolution on an input with three different sizes of filters (1×1, 3×3, 5×5) at the same level. Moreover, max pooling is also performed, resulting in a wider rather than deeper network, and then the outputs are concatenated and fed into the next layer. The last architecture might cover the issue of different sizes of filters, and it may do it very inefficiently, leading to a computational blow-up within a few stages. However, we can reduce the parameters of the network by adding an extra 1×1 convolution before the 3×3 and 5×5 convolutions, and an extra 1×1 after the Max Pooling layer. The InceptionV3 architecture was published in the same paper as InceptionV2 in 2015, achieving 93.7% top-5 accuracy with 23.9 M parameters. It has a total of 42 layers, and it has many improvements over the previous Inception architectures, such as: i) the auxiliary classifier is mainly used to reduce the vanishing gradient problem, ii) there is less computational cost compared to the previous Inception version, and iii) it has a low error rate compared with its previous models.

3.1.6. ResNet50

ResNet50 achieves 92.1% Top-5 accuracy with 25.6 M parameters, which is short for residual network. The use of skip connections in the standard NN was the novel idea behind the architecture, which helps to reduce the vanishing gradient problem that occurs when NNs deepen. According to the original ResNet paper [8], it is worth noticing that making the network deeper led to higher classification errors for 56-layer compared to the shallower architecture of 20-layer. In fact, when the network depth increases by simply adding layers, accuracy is saturated and then degrades rapidly. The “identity shortcut connection” that skips almost two layers does not degrade the performance of the network because the weights and bias in the worst scenario can have small or zero values, and the network learns the identity function.

3.1.7. Vgg16

The VGG16 model achieves 90.1% Top-5 accuracy with 138.4 M parameters and supports 16 layers. It is a convnets model proposed by Simonyan and Zisserman [6] from the University of Oxford. The VGG model uses a stack of three 3×3 convolutions instead of a single one, resulting in more non-linearity and discrimination of the model, and it also reduces the number of parameters.

3.2. Slum dataset

In this study, we manually collected (448, 448, 3) settlement slum photographs at medium resolution using Google Earth Pro, which employs satellite imagery, due to a dearth of official slum image dataset, specifically their magnitude, location, limits, and populations. The settlement of slums in Figure 1 serves as an example of how we may visually distinguish the delineation of slum boundaries, which change dynamically in both space and time [42]. Furthermore, this can result in the decreasing or eventual expansion of the slum housing being hidden by vegetation or building camouflage. From satellite imagery of Northern Morocco, we have classified many images into 2,051 slum images. The dataset was further subdivided into training and validation set with each containing both slum and not-slum annotated images.

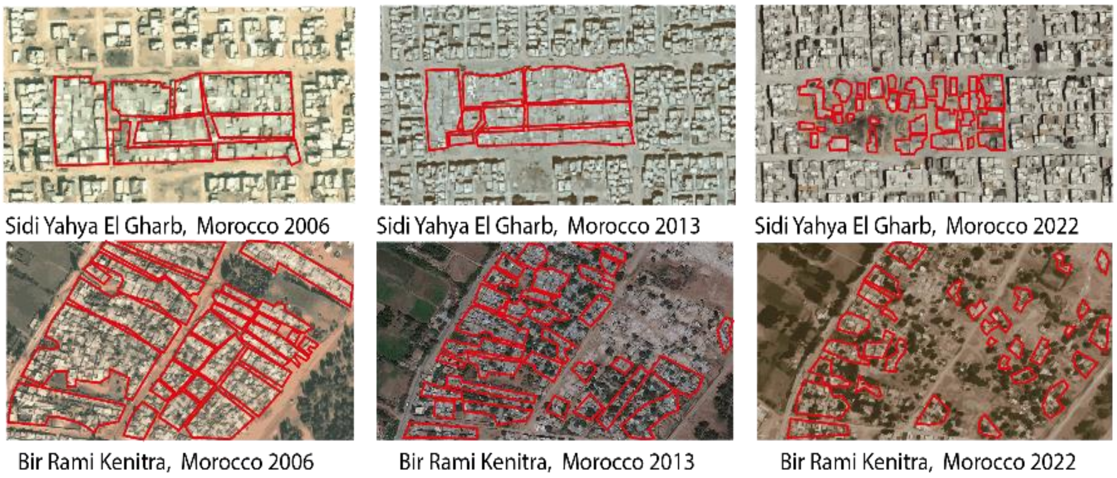

Sidi Yahya El Gharb, Morocco 2006
Sidi Yahya El Gharb, Morocco 2013
Sidi Yahya El Gharb, Morocco 2022
Bir Rami Kenitra, Morocco 2006
Bir Rami Kenitra, Morocco 2013
Bir Rami Kenitra, Morocco 2022

Figure 1. Schematic illustration of two different study areas, “Sidi Yahya El Gharb” and “Bir Rami Kenitra” in Kenitra city, over the years 2006, 2013, and 2022. Slum housing boundaries in red are established manually

As an example, the classification [43] categorizes informal settlements into five classes according to fieldwork surveys and discussions with local experts. We conclude that several of the identified challenges were that slums can visually include various morphologies, environments with little vegetation and trees, and roofs covered with used tires [44]. All of these share certain types of features, such as edges, different reflectance shades of gray and rust (color), smooth texture, and very dense areas.

3.3. Data preprocessing

We have collected 2,051 slum images from satellite imagery of Kenitra, Morocco. Before the training began, we first resized images from (448,448,3) to the size (224, 224, 3) for MobileNets; (299, 299, 3) for InceptionV3, (224, 224, 3) for NASNetMobile, (299, 299, 3) for Xception; (224, 224, 3) for VGG16; (224, 224, 3) for EfficientNetB0; and (224, 224, 3) for ResNet50. Then we shuffle and split the dataset into test samples and training samples with approximately a 1:5 ratio.

3.4. Pretrained models' implementation

TL methods can be categorized into three subcategories: inductive transfer learning, transductive transfer learning, unsupervised transfer learning. Firstly, we take a pretrained model with all previously frozen layers that contain features learned. Secondly, we add trainable fully connected layers on top of the frozen layers as long as we do not exceed the graphics processing unit (GPU) memory. Thirdly, we feed the output of data processing into the new model. Fourthly, we tune hyperparameters such as (learning rates, batch-size, epochs) and train the new model toward our specific classification task, as shown in the Figure 2. Once training begins with the new dataset, the generic weights are kept frozen to evaluate a new set of images, whereas the newly added task-specific layers are allowed to be modified [45].

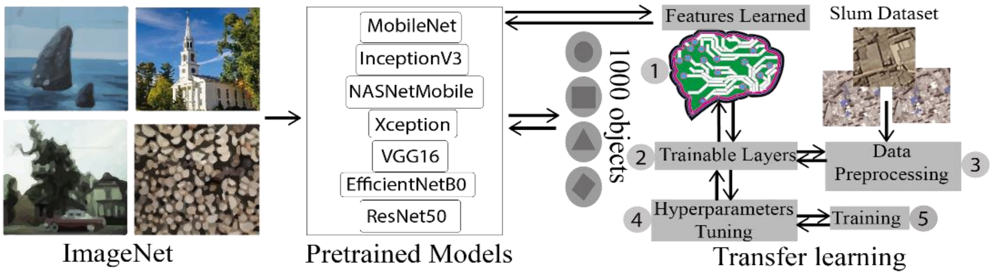


Figure 2. An implementation block diagram of pretrained models describing the process of the TL approach used in this study

3.5. Software environment

The training and evaluation of all models are performed in the workstation with central processing unit (CPU) and graphics processing unit (GPU) of the Intel(R) Core (TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz and NVIDIA GeForce 1650 Ti of 4 Go. The platform was equipped with 64-bit Windows 10. “Anaconda2.1.1”. And “TensorFlow2.4.0”. “cuda_11.0.3_win10_network”. cudnn-11.0-windows-x6”. And “Keras 2.4.0”, which facilitates the implementation of CNNs architectures deep learning models.

3.6. Computer code availability

All code associated with the experiments is available at https://github.com/mouddentarik/Detect-Slums.git. Namely, the transfer learning-based CNNs, prediction, and visualization code. The slum image dataset used in this study can be obtained by contacting tarik.elmoudden@uit.ac.ma.

4. RESULTS AND DISCUSSION

4.1. Pretrained model training

This research evaluates the performance of the pretrained model during training according to many performance metrics such as:

- Overall accuracy $\mathrm{OA} = (\text{number of correct predictions})/(\text{total number of predictions})$
- Precision: is the percentage of correct predictions of a class among all predictions for that *class.its* value is calculated from the four outcomes of the confusion matrix [46],
- Training time: time-consuming during the training phase of pretrained models.

– Fourteen (14) curves: accuracy and loss curves help us to assess qualitatively the fluctuations and the convergence during training.

In this paper, we have trained seven different pretrained models over 70 epochs. Each pretrained uses the Adam optimizer because it is the most preferred optimizer in several studies and converges faster. For the *batch_size* hyperparameter, we stick to the value of 32 in order to not exceed the dedicated GPU memory and avoid the training crash. We have also run identical experiments of each pretrained model 30 times to calculate the variance of the final overall accuracy and loss.

For the smallest memory size MobileNetV2. We add dense layers 1: “layers. Dense(196, activation= ‘ReLU’)”, followed by “layers.Dropout(0.1)” to avoid the overfitting, and finally dense layers 2 “layers.Dense(2, activation=‘softmax’)”, without fear of exceeding the GPU memory. For binary classification (slum or not slum), MobileNetV2 achieved the greatest accuracy of 98.78% OA, 0.004 loss, 96.2% precision, and 4min12 s training time when compared to other pretrained models, as described in Table 2. As illustrated in Figures 3(a) and 3(b) (see in appendix), we find extremely weak oscillations during training, very quick convergence, and the new model reaches the maximum accuracy at epoch 10. For the Xception pretrained model, the best result was obtained using the Adam (*learning_rate*=0.001) optimizer. According to Table 2, Xception achieves: 96.5%±0.49% OA, 0.073±0.004 loss, 100% precision, and 30min2s training time. Thus, it is underlined by the weak oscillations during training, fast convergence, reach maximal accuracy at epoch 50, as shown in Figures 3(c) and 3(d) (see in appendix).

As we can see in Table 2, NASNetMobile and Inceptionv3 pretrained models both achieve approximately the same performance. Their OA, loss, and precision attained approximately 97.8%, 0.05, and 96.1%, consecutively. The same appears to be true for accuracy and loss curves in Figures 3(e) to 3(h) (see in appendix), in spite of the slight difference in the training time, which does not matter. Table 2 shows that the pretrained model VGG16 achieves 97.47%±1.9% of OA, 0.068±0.004 of loss, 100% of precision, and 13 minutes and 57 seconds of training time. This means that VGG16 suffers from the highest variance, which can decrease the reproducibility of the model and can produce unpredictable results during the execution stage, as shown Figures 3(i) and 3(j) (see in appendix). For these reasons, we will exclude it from the pretrained model execution sub-section.

EfficientNet and ResNet50, as shown in Figures 3(k) to 3(n) (see in appendix), suffer from excessively large oscillations during training. And also, they have poor performance for all quantitative metrics, according to Table 2, making them unsuitable models for binary classification. This suggests that their internal architecture is not well suited to finding patterns in our dataset. We shall therefore exclude them from the pretrained model execution sub-section.

Table 2. The quantitative performance comparison of different pretrained models in terms of validation accuracy (%), validation loss (%), precision (%), and training time. (Best model in italic)

Epochs	70	70	70	70	70	70	70
Validation Accuracy	98.78%	68.29%	97.56%	96.5%±0.49%	97.9%±0.57%	82.2%±7.68%	97.47%±1.9%
Validation Loss	0.044±0.003	0.627±0.012	0.051±0.001	0.073±0.004	0.05±0.005	0.395±0.086	0.068±0.014
Precision	≃96.2%	≃ null	≃96.15%	≃100%	≃96.2%	≃100%	≃100%
Training time	4 min 12 s	5 min 50 s	7 min 12 s	30 min 2 s	11 min 34 s	8 min 35 s	13 min 57 s
Model	*Mobile-Nets*	Efficient-Net	*NASNet-Mobile*	*Xception*	*Inceptionv3*	ResNet50	*VGG16*

4.2. Pretrained model execution

Other essential metrics for the analysis of a model in the real-world application are its latency. In this paper, the latency corresponds to the total delay between the blocks “preprocessing the image” and “visualization”, as shown in Figure 4. To measure and compare the pretrained models’ latencies, we decided to pick three pretrained models: MobileNet, NASNetMobile, and Xception. This is firstly because they achieve good performance and, secondly, because they have a low-size file (HDF5 format), which is better suited to hardware with limited computational resources, as shown in Table 3.

In our application, as shown in Figure 4, the images are collected as (4800,2912,3) from Pro Google Earth. Then, the algorithm loop is launched to crop all the specific sizes of (299, 299, 3), or (224, 224, 3) to feed the input of the model. Then, we run loop inferences on each cropped image to get the coordinates of possible slum housing. Last, for visualization, we add the red rectangle to the image directly by exploiting the coordinates. It was shown in Figure 5 that three pre-trained models achieve a good delineation of slum areas. And this allows administrative agencies to survey the spatial distribution of slum housing in northern Morocco. Moreover, the latency of the MobileNet, NASNetMobile, and Xception is 17.01, 19.71, and 23.5 s, respectively, as shown in Table 3.

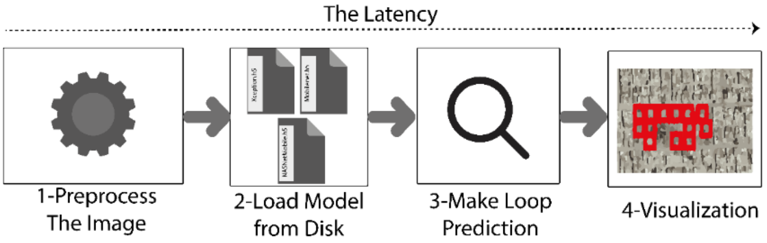


Figure 4. The procedure for loading and executing the pre-trained models: MobileNet, NASNetMobile, and Xception. This procedure started by "preprocess the image" until "visualization"

Table 3. Latency and memory size of MobileNets, NASNetMobile, Xception

Average Latency	17.01s	19.71s	23.5s
Size (MB)	9.1	17.9	79.9
Model	MobileNets	NASNetMobile	Xception

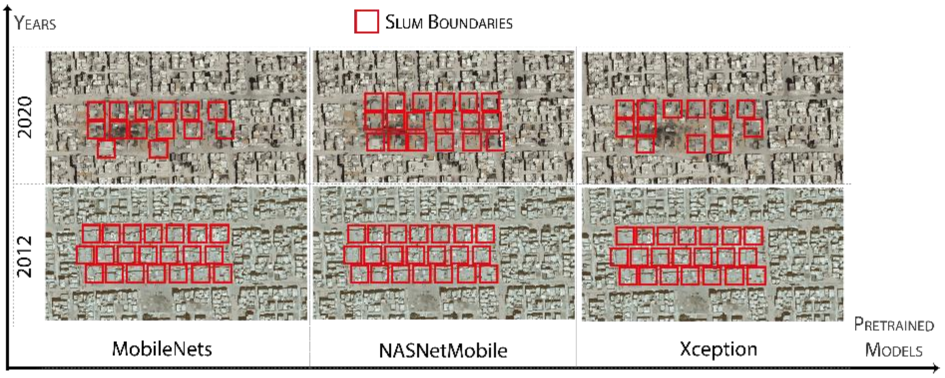


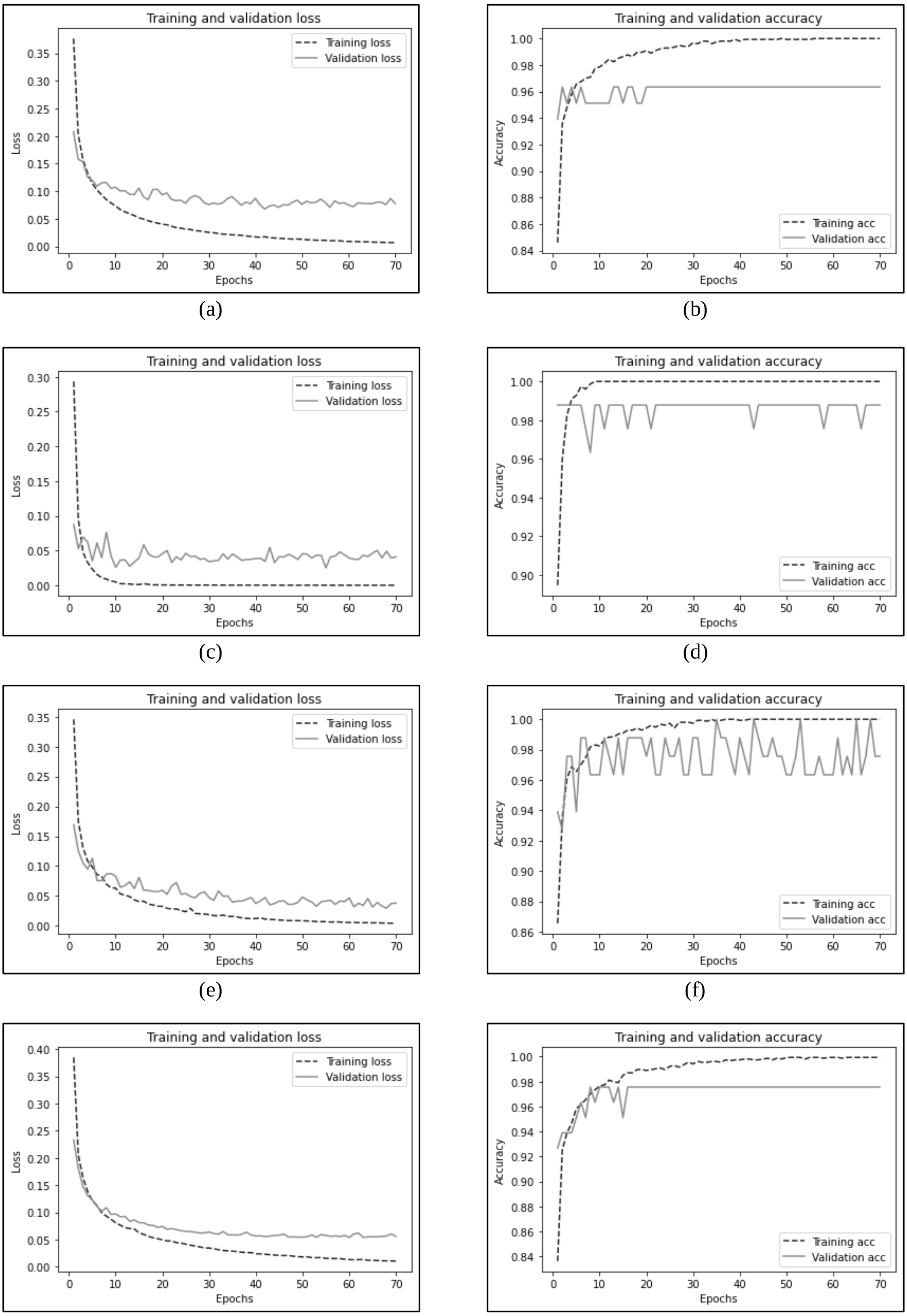
Figure 5. Comparative localizing slum instances results by Xception, MobileNets, and NASNetMobile pretrained models, using the same study area "Sidi Yahya El Gharb" captured by Pro Google Earth satellite. Three images in the first and second row are captured in 2020 and 2012, respectively. Bounding box surround slum housing

5. CONCLUSION AND RECOMMENDATION

This study has investigated, explored, and applied the TL approach to detect and localize slum housing in northern Morocco. The results obtained show that training our slum dataset for the three pretrained models: MobileNet, NASNetMobile and Xception are very promising-in terms of their highest accuracy and precision and lowest latency-can be an effective tool when used in conjunction with other existing programmers to measure and survey the expansion of slum housing in northern Morocco. From seven pretrained assessments, the result shows that MobileNets, NASNetMobile and Xception are more stable, namely with weak oscillation during training using Adam optimizer with a default learning rate of 0.0001 and *batch_size* of 32. In addition to that, these three pretrained models can localize camouflaged slum housing under vegetation, or between urban neighborhoods, with high precision. It should be noted that we can use the MobileNet pretrained model in the first place because it is a confident, accurate, and efficient large-scale detecting and localizing slum housing system, which can be implemented easily on low-cost computational hardware to exploit the RS database.

However, our approach has two major possible limitations. The first is generating a rectangle around the slum housing within roads and vegetation instead of using polygon form to surround the borders of slum housing. The second limitation concerns using resized images from (448,448,3) to (224,224,3) or (299,299,3) which has a negative impact on the quality of images. Consequently, the pretrained models may be influenced by bad inputs and can produce low results. Within these limitations, we recommend two future works. One compares pretrained backbones for semantic segmentation, which extracts the border of slum housing from the background. The other one uses fully convolutional networks, which take arbitrary input size and produce semantic segmentation.

APPENDIX



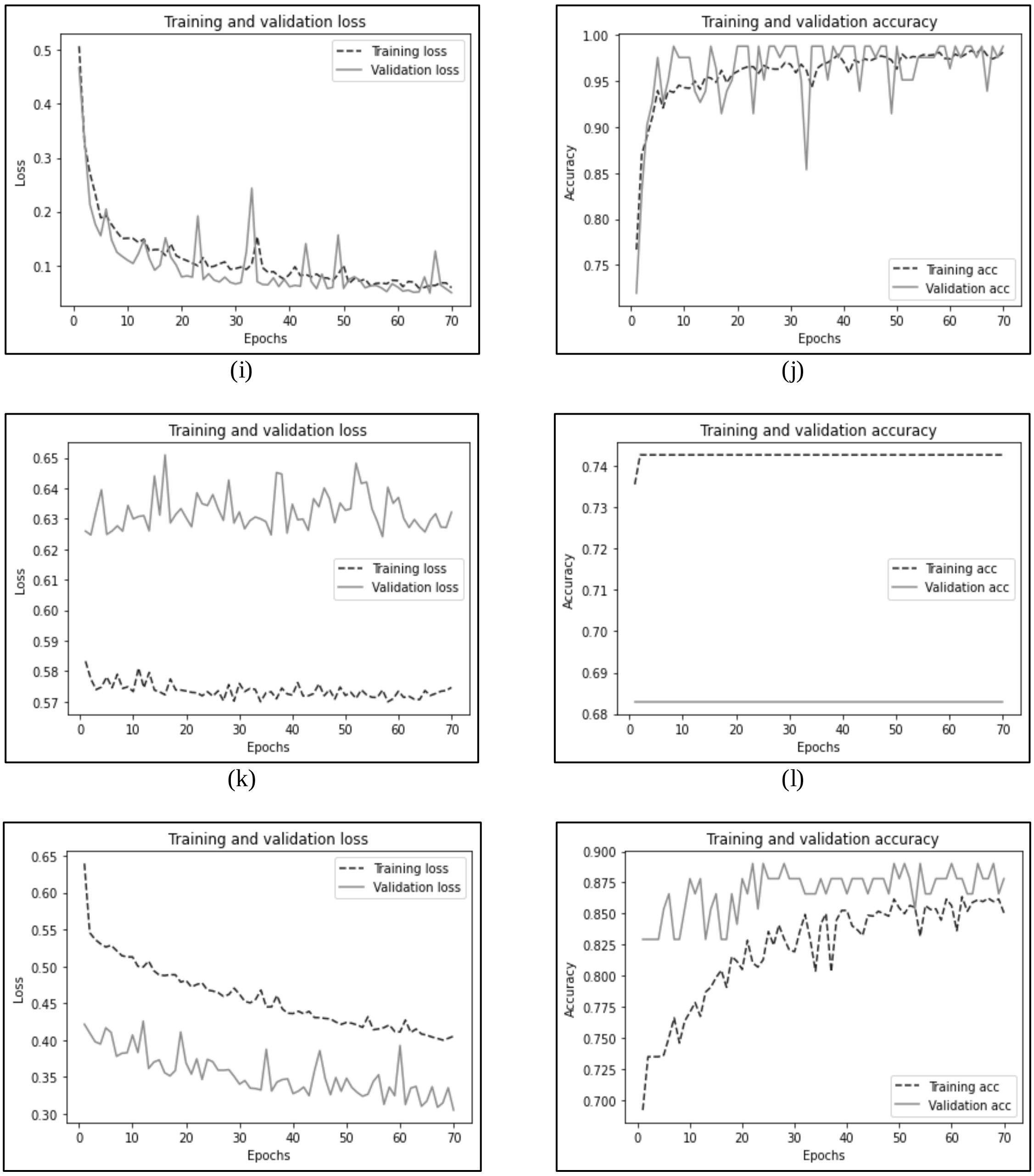
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3. Graphs for training and validation accuracy, loss versus number of epochs: (a) MobileNet [loss], (b) MobileNet [acc], (c) Xception [loss], (d) Xception [acc], (e) Inceptionv3 [loss], (f) Inceptionv3 [acc], (g) NASNetMobile [loss], and (h) NASNetMobile [acc] *(continue)*



(i) (j)

(k) (l)

(m) (n)

Figure 3. Graphs for training and validation accuracy, loss versus number of epochs: (i) EfficientNet [loss], (j) EfficientNet [acc], (k) VGG16 [loss], (l) VGG16 [acc], (m) ResNet50 [loss], and (n) ResNet50 [acc]

REFERENCES

[1] Department of Housing and Urban Policy, “The national program 'cities without slums (VSB)',” Ministry of national territory planning, land planning, housing and city policy of Morocco http://www.mhpv.gov.ma/?page_id=956 (accessed Jan. 01, 2021).

[2] M. Atia, “Refusing a ‘City without Slums’: Moroccan slum dwellers’ nonmovements and the art of presence,” *Cities*, vol. 125, Art. no. 102284, Jun. 2022, doi: 10.1016/j.cities.2019.02.014.

[3] M. Coccia, “How (Un)sustainable environments are related to the diffusion of COVID-19: the relation between coronavirus disease 2019, air pollution, wind resource and energy,” *Sustainability*, vol. 12, no. 22, Nov. 2020, doi: 10.3390/su12229709.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

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

[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[8] 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.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[10] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[11] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, Apr. 2017, [Online]. Available: http://arxiv.org/abs/1704.04861.

[12] B. Zoph, V. Vasudevan, J. Shlens, and Q. V Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[13] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

[14] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *Journal of Applied Remote Sensing*, vol. 11, no. 4, Sep. 2017, doi: 10.1117/1.JRS.11.042609.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[17] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A Systematic Literature Review on Particle Swarm Optimization Techniques for Medical Diseases Detection," *Computational and Mathematical Methods in Medicine*, pp. 1–10, Sep. 2021, doi: 10.1155/2021/5990999.

[18] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 10, pp. 1537–1541, 2016.

[19] W. Haider Bangyal *et al.*, "New modified controlled bat algorithm for numerical optimization problem," *Computers, Materials and Continua*, vol. 70, no. 2, pp. 2241–2259, 2022.

[20] S. Ghaffarian, D. Roy, T. Filatova, and N. Kerle, "Agent-based modelling of post-disaster recovery with remote sensing data," *International Journal of Disaster Risk Reduction*, vol. 60, Jun. 2021, doi: 10.1016/j.ijdrr.2021.102285.

[21] M. Kuffer, K. Pfeffer, and R. Sliuzas, "Slums from space-15 years of slum mapping using remote sensing," *Remote Sensing*, vol. 8, no. 6, May 2016, doi: 10.3390/rs8060455.

[22] D. Verma, A. Jana, and K. Ramamritham, "Transfer learning approach to map urban slums using high and medium resolution satellite imagery," *Habitat International*, vol. 88, Jun. 2019, doi: 10.1016/j.habitatint.2019.04.008.

[23] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, Nov. 2010, doi: 10.1109/TGRS.2010.2060550.

[24] B.-C. Kuo, J.-M. Yang, T.-W. Sheu, and S.-W. Yang, "Kernel-based KNN and gaussian classifiers for hyperspectral image classification," in *IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, pp. 1006–1008, doi: 10.1109/IGARSS.2008.4779167.

[25] R. Dahmani, A. A. Fora, and A. Sbihi, "Extracting slums from high-resolution satellite images," *International Journal of Engineering Research and Development*, vol. 10, no. 9, pp. 1–10, 2014.

[26] E. Ranguelova, B. Weel, D. Roy, M. Kuffer, K. Pfeffer, and M. Lees, "Image based classification of slums, built-up and non-built-up areas in Kalyan and Bangalore, India," *European Journal of Remote Sensing*, vol. 52, pp. 40–61, Mar. 2019, doi: 10.1080/22797254.2018.1535838.

[27] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] M. Wurm, H. Taubenböck, M. Weigand, and A. Schmitt, "Slum mapping in polarimetric SAR data using spatial features," *Remote Sensing of Environment*, vol. 194, pp. 190–204, Jun. 2017, doi: 10.1016/j.rse.2017.03.030.

[29] X. X. Zhu *et al.*, "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, Dec. 2017, doi: 10.1109/MGRS.2017.2762307.

[30] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal Processing: Image Communication*, vol. 47, pp. 482–489, Sep. 2016, doi: 10.1016/j.image.2016.05.007.

[31] R. S. Andersen, A. Peimankar, and S. Puthusserypady, "A deep learning approach for real-time detection of atrial fibrillation," *Expert Systems with Applications*, vol. 115, pp. 465–473, Jan. 2019, doi: 10.1016/j.eswa.2018.08.011.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[33] Z. Yang *et al.*, "Deep transfer learning for military object recognition under small training set condition," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6469–6478, Oct. 2019, doi: 10.1007/s00521-018-3468-3.

[34] R. Cao *et al.*, "Enhancing remote sensing image retrieval using a triplet deep metric learning network," *International Journal of Remote Sensing*, vol. 41, no. 2, pp. 740–751, Jan. 2020, doi: 10.1080/2150704X.2019.1647368.

[35] A. Ajami, M. Kuffer, C. Persello, and K. Pfeffer, "Identifying a slums' degree of deprivation from VHR images using convolutional neural networks," *Remote Sensing*, vol. 11, no. 11, May 2019, doi: 10.3390/rs11111282.

[36] Z. Wang, Z. Dai, B. Poczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11293–11302.

[37] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning ICANN 2018*, 2018, pp. 270–279.

[38] T. Stark, M. Wurm, H. Taubenbock, and X. X. Zhu, "Slum mapping in imbalanced remote sensing datasets using transfer learned deep features," in *2019 Joint Urban Remote Sensing Event (JURSE)*, May 2019, pp. 1–4, doi: 10.1109/JURSE.2019.8808965.

[39] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenbock, "Satellite-based mapping of urban poverty with transfer-learned slum morphologies," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5251–5263, 2020, doi: 10.1109/JSTARS.2020.3018862.

[40] Z. Pan, J. Xu, Y. Guo, Y. Hu, and G. Wang, "Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net," *Remote Sensing*, vol. 12, no. 10, May 2020, doi: 10.3390/rs12101574.

[41] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, "Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 59–69, Apr. 2019, doi: 10.1016/j.isprsjprs.2019.02.006.

[42] J. E. Patino and J. C. Duque, “A review of regional science applications of satellite remote sensing in urban settings,” *Computers, Environment and Urban Systems*, vol. 37, pp. 1–17, Jan. 2013, doi: 10.1016/j.compenvurbsys.2012.06.003.

[43] M. Kuffer, K. Pfeffer, R. Sliuzas, I. Baud, and M. Maarseveen, “Capturing the diversity of deprived areas with image-based features: the case of Mumbai,” *Remote Sensing*, vol. 9, no. 4, Apr. 2017, doi: 10.3390/rs9040384.

[44] R. Sliuzas, M. Kuffer, and I. Masser, “The spatial and temporal nature of urban objects,” in *Remote Sensing of Urban and Suburban Areas*, 2010, pp. 67–84.

[45] A. Koul, S. Ganju, and M. Kasam, *Practical deep learning for cloud, mobile, and edge: real-world AI and computer-vision projects using Python, Keras and TensorFlow*. O’Reilly Media, 2019.

[46] G. M. Foody, “Status of land cover classification accuracy assessment,” *Remote Sensing of Environment*, vol. 80, no. 1, pp. 185–201, Apr. 2002, doi: 10.1016/S0034-4257(01)00295-4.

BIOGRAPHIES OF AUTHORS

Tarik EL Moudden received his bachelor’s degree, in electronic and master’s degree, in computer science and telecommunications from Ibn Tofail University, Kenitra city. His research interests include artificial intelligence (AI), and data analysis systems. He can be contacted at tarik.elmoudden@uit.ac.ma.

Rachid Dahmani master’s degree in computer science and telecommunications in the field of communication system and information processing. he has a degree in topography and photogrammetry. Currently he holds the position of head of topography and extension service at the urban agency of Kenitra-Sidi Kacem-Sidi Slimane. Currently his area of research is the clearance of slums through conventional methods based on restructuring work and related topographic surveys. He can be contacted at rachid.dahmani1@uit.ac.ma.

Mohamed Amnai received his bachelor’s degree in 2000, in IEEA (Computers, Electronics, Electrical and Automation) from Molay Ismail’s University, Errachidia city. Then, he obtained his master’s degree in 2007, from Ibn Tofail University, Kenitra city. In 2011, he received his Ph.D. on telecommunications and computer science, from Ibn Tofail University in Kenitra city, Morocco. Since March 2014 he is a professor, he has been professor at National School of Applied Sciences Khouribga, Settat University, Morocco. And now he is professor in computer science research laboratory, Ibn Tofail University, Kenitra, Morocco. He can be contacted at mohamed.amnai@uit.ac.ma.

Abderrahmane Aït Fora holds a Ph.D. in remote sensing between 1995 and 1991 at the University of Sherbrooke. Doctor in quaternary geology between 1983-1986 at the University of Bordeaux 1. He is currently interested in the fields of the fight against desertification, remote sensing, and detection of soil erosion, also to the Mapping of the State of Vegetation Cover in the North of Ivory Coast from Satellite Images. He can be contacted at abderrahman.aitfora@uit.ac.ma.