

Quantifying Uncertainty in Slum Detection: Advancing Transfer Learning With Limited Data in Noisy Urban Environments

Thomas Stark¹, Michael Wurm¹, Xiao Xiang Zhu¹, *Fellow, IEEE*, and Hannes Taubenböck¹

Abstract—In the intricate landscape of mapping urban slum dynamics, the significance of robust and efficient techniques is often underestimated and remains absent in many studies. This not only hampers the comprehensiveness of research but also undermines potential solutions that could be pivotal for addressing the complex challenges faced by these settlements. With this ethos in mind, we prioritize efficient methods to detect the complex urban morphologies of slum settlements. Leveraging transfer learning with minimal samples and estimating the probability of predictions for slum settlements, we uncover previously obscured patterns in urban structures. By using Monte Carlo dropout, we not only enhance classification performance in noisy datasets and ambiguous feature spaces but also gauge the uncertainty of our predictions. This offers deeper insights into the model’s confidence in distinguishing slums, especially in scenarios where slums share characteristics with formal areas. Despite the inherent complexities, our custom CNN STnet stands out, delivering performance on par with renowned models like ResNet50 and Xception but with notably superior efficiency—faster training and inference, particularly with limited training samples. Combining Monte Carlo dropout, class-weighted loss function, and class-balanced transfer learning, we offer an efficient method to tackle the challenging task of classifying intricate urban patterns amidst noisy datasets. Our approach not only enhances artificial intelligence model training in noisy datasets but also advances our comprehension of slum dynamics, especially as these uncertainties shed light on the intricate intraurban variabilities of slum settlements.

Index Terms—Imbalanced dataset, learning from few samples, noisy dataset, slum mapping, transfer learning, uncertainty estimation.

I. INTRODUCTION

THE criticality of data in artificial intelligence (AI), particularly in deep learning model development, are

Manuscript received 15 October 2023; revised 11 December 2023 and 18 January 2024; accepted 25 January 2024. Date of publication 29 January 2024; date of current version 15 February 2024. The work of X. Zhu is supported by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab “AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond” under Grant 01DD20001. This work was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program under Grant 714087-So2Sat. (*Corresponding author: Hannes Taubenböck.*)

Thomas Stark and Xiao Xiang Zhu are with the Chair of Data Science in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: t.stark@tum.de; xiaoxiang.zhu@tum.de).

Michael Wurm and Hannes Taubenböck are with German Remote Sensing Data Center, German Aerospace Center, 82234 Weßling, Germany (e-mail: michael.wurm@dlr.de; hannes.taubenboeck@dlr.de).

Digital Object Identifier 10.1109/JSTARS.2024.3359636

well-documented [1], [2], [3]. Quality datasets, free from biases and errors, are essential for creating algorithms that are generalizable and trustworthy for decision-making processes [4], [5], [6]. However, the prevalence of biases and inaccuracies in training datasets necessitates either thorough curation or specialized methods to handle these challenges. The advancement of AI architectures has been significant in addressing issues like imbalanced and noisy datasets, or classifying within fuzzy feature spaces [7], [8], [9]. These improvements are pivotal for handling the complexities and unpredictability of real-world scenarios, underscoring the importance of data quality in AI workflows for accurate and meaningful outcomes.

By leveraging AI, researchers can uncover hidden connections and gain a deeper understanding of complex phenomena, leading to more insightful studies and breakthrough discoveries. The constant evolution and improvement of AI architectures, especially in dealing with challenging datasets marked by imbalanced and noisy datasets [7], [8], or classifying within fuzzy feature spaces [9], has empowered researchers to handle diverse and unpredictable real-world scenarios effectively. As AI continues to progress, it brings the promise of more comprehensive and accurate solutions for the complexities of our dynamic world.

One area where AI has shown promising results is in remote sensing, particularly when it comes to understanding urban environments [10], [11]. This technology has been used to gather vast amounts of insightful data on cities, including information about population density [12], land use [13], or transportation patterns [14]. This also includes detecting urban poverty, where researchers and policymakers can gain valuable insights on locations of slum settlements. The utilization of high-resolution remote sensing imagery played a pivotal role in the comprehensive mapping of slums within the dynamic cityscape of Mumbai, as highlighted in [15]. Similarly, the city of Accra witnessed the integration of remote sensing data in conjunction with income data, facilitating an insightful mapping of poverty patterns, as seen in [16]. Furthermore, Kuffer et al. [17] conducted an intricate examination of the multifaceted factors that contribute to the enduring presence of slums, shedding light on their persistence within urban landscapes. Satellite imagery, population data, and economic indicators can help to recognize poverty patterns and map poverty levels to identify needy areas, enabling more focused and effective poverty reduction activities [18], [19], [20], [21].



Fig. 1. Dense and low-rise areas shown with a black outline for the city of Nairobi [26]. Google Street View imagery is used to show that only some parts of the dense areas can also be considered a slum settlement highlighting the challenge of slum mapping.

Detecting urban poverty from remote sensing data is very challenging, due to data availability and the many different morphological features that can occur in slum settlements [22]. The issue of data availability is twofold: While some data exist for large and often studied areas [23], [24], for many cities of the Global South there are still very few data on slum settlements following a coherent and reproducible approach. The data that exist on slum settlements is often outdated, incomplete, and based on heterogeneous approaches on its definition regarding the morphology of slum settlements [24]. The second major challenge to detect slum settlements is the nature of its noisy feature space. Despite the fact that a typical morphological slum can be characterized by its high building density, small and complex street layouts, low-rise and small building structures, and use of a wide variety of construction materials, in reality slum settlements sometimes share just parts of these features [22]. Moreover, as indicated in [25], the delineation of slums is subject to variability owing to differing opinions on what constitutes a slum. Recognizing this, the data used for training an AI to classify slum settlements needs to diligently harmonized into a unified dataset to enhance a study's reliability, given that such variability in slum definitions could markedly affect the results. This subjectivity poses a challenge in classifying urban poverty. This impact can make it difficult to distinguish between a slum settlement and a formal built-up region. This challenge is depicted in Fig. 1 where the results from [26] show predictions of the local climate zone class seven, which is described as dense-low-rise buildings and shows two areas within the city of Nairobi, Kenya. While both highlighted areas display dense and low-rise building structures, only some parts of one highlighted area can be described as a slum upon having a closer look using Google Street View imagery. Thus, classifying a settlement as a slum cannot be solely determined by the previously mentioned features. Conversely, just because a settlement has a low-rise and dense structure does not automatically make it a slum. Similarly, the absence of density in a settlement does not guarantee that it cannot be classified as a slum. In other words, the combination of multiple morphological characteristics is a detrimental criterion

for determining whether a settlement is a slum or not. While other factors, like plumbing and access to basic services need to be considered in evaluating the status of a settlement as well, these are not derivable from high-resolution remote sensing data. Thus, with the described noisiness of the dataset in mind, for the purpose of this research and considering the limitations in acquiring actual real ground-truth data, we rely here on the typical morphological appearance of slum settlements.

In our study, we focus on addressing two primary challenges: limited data availability and noisy datasets in the context of slum mapping using remote sensing data. Our main goal is to develop an efficient method for detecting slums with limited training samples, and to estimate the uncertainty in these predictions. To this end, we employ a transfer-learning approach, leveraging a large, imbalanced dataset to effectively train toward a smaller, balanced dataset. This method ensures that only a few samples are needed for successful slum detection. To tackle the issue of noisy datasets, we utilize Monte Carlo dropout. This technique allows us to approximate the uncertainty associated with predicting slum settlements, providing a more robust and reliable analysis. In addition, we introduce a custom convolutional neural network (CNN), the slum transfer network (STnet), specifically designed for high-resolution remote sensing data. STnet is engineered not only to enhance the training efficiency with a limited number of samples but also to offer significant improvements in processing time compared to standard CNN models. Our research aims to demonstrate the effectiveness of STnet in accurately detecting slums in various urban environments, thereby contributing to the broader field of urban studies and remote sensing.

II. RELATED WORK

A. Detecting Urban Poverty Using Remote Sensing

Traditional machine learning approaches have already made significant contributions to the detection of urban poverty by enabling the analysis of large datasets [24]. These approaches

have proven to be invaluable in providing researchers and policy-makers with the necessary tools to gain a deeper understanding of poverty patterns within urban areas. By employing various machine learning algorithms, such as classification and regression models, researchers can process and analyze extensive datasets containing socioeconomic and spatial information [19].

A specific area where traditional machine learning has shown promise is in the application of remote sensing data for larger scale urban poverty detection [27], [28]. In the context of poverty detection, remote sensing data provide valuable information about the morphological patterns of slum settlements. This data can include features such as building density, land cover classification, and infrastructure characteristics [29], [30].

While traditional machine learning approaches have been effective in urban poverty detection, recent advancements in AI have further enhanced our ability to identify poverty using innovative techniques. AI, including deep learning models, has demonstrated remarkable capabilities in analyzing satellite imagery for poverty detection [15], [27], [31], [32], [33]. Deep learning algorithms, characterized by their ability to learn hierarchical representations of data, can automatically extract intricate visual features from satellite images, capturing subtle patterns that may indicate poverty.

However, despite these advancements, there is still a need for larger scale applications of poverty detection using AI. Most existing studies in this field are often limited to specific areas of interest within the same geographical region. To fully harness the potential of AI in urban poverty detection, it is essential to expand research efforts to encompass a broader range of urban environments worldwide. By doing so, it is intended to unlock the true power of AI in addressing the complex challenges associated with urban poverty on a global scale.

B. Training on Imbalanced Datasets

Studies revealed that slum morphologies in general consist of a small share of the built-up environment in cities, and in particular, mapping information is only scarcely if at all available [34]. Dealing with imbalanced datasets in deep learning involves several approaches that can help mitigate the issue of class imbalance. Some common methods include cost-sensitive learning, as seen in [35], which adjusts misclassification costs, favoring the minority class and improving overall performance on imbalanced datasets. Synthetic data generation increases the minority class representation by creating artificial samples, achieving a more balanced dataset and enhancing predictive accuracy [36], [37]. Using curriculum learning gradually exposes the algorithm to challenging examples, minimizing biases toward the majority class [38], [39].

Another simple approach is resampling the dataset by either oversampling the minority class or undersampling the majority class. Oversampling involves replicating or generating new instances from the minority class to balance the dataset. Undersampling reduces the majority class to match the minority class. Both approaches help achieve a more balanced class distribution and improve AI model performance [33], [40], [41]. The choice

between them depends on the dataset and learning algorithm used.

Furthermore, class weight adjustment is a technique in AI used to tackle imbalanced datasets. By assigning higher weights to the minority class during training, the model places greater emphasis on learning from the minority class. This helps to address the issue of class imbalance and ensures that the model pays more attention to the minority class, improving its ability to correctly classify instances from that class. By adjusting the class weights, the model becomes more sensitive to the minority class and achieves a better balance in handling imbalanced datasets [42], [43].

It is important to carefully evaluate the performance of the model after implementing these methods to ensure that the imbalance has been effectively addressed without negatively impacting the overall performance. In this work, we direct our attention toward a class-weighted loss function for pretraining and for transfer learning an undersampling method. Both present a straightforward and efficient workflow that can be effortlessly replicated. By choosing to focus on these specific methods, we aim to harness their advantage and capitalize on their ease of implementation.

C. Transfer Learning From Few Samples

Transfer learning a CNN involves adjusting the weights of an already trained model to fit the specific task or dataset in the target domain. This is achieved by pretraining a model and retraining it with a smaller learning rate on a related classification task for the target domain [44]. The benefits of transfer learning a CNN include: faster training times as the model has already learned useful features from the pretraining data [45], [46], improved performance on the target task as compared to training a model from scratch, and the ability to leverage the knowledge gained by the pretrained model on a large dataset to improve the performance on a smaller dataset [47], [48].

When it comes to transfer learning with few samples, the situation is similar to few-shot learning techniques. However, in transfer learning, the focus is not solely on handling a few labeled examples of a new task. Instead, transfer learning aims to exploit the knowledge learned from a source task with sufficient labeled data and apply that knowledge to a target task with limited labeled data. Whereas in few-shot learning, the model is trained to learn from none or very few labeled samples. In [9], a few-shot learning technique from [49] was used in order detect complex morphologies representing poor areas within the urban environment, the authors found out that the technique works very well when only a hand-full of samples are available. Other approaches have been using self-supervised embedding optimization for adaptive generalization in urban settings [50] or using prototypical networks for urban damage detection after natural hazards [51].

D. Bayesian Uncertainty Estimation

In deep learning, Bayesian uncertainty refers to the incorporation of probabilistic inference into neural networks and can be categorized into two domains: epistemic and aleatoric.

The former, epistemic uncertainty, pertains to the uncertainty associated with the model parameters or weights, while the latter, aleatoric uncertainty, is commonly associated with data uncertainty.

Variational inference, which models the network's weights as probability distributions and employs optimization techniques to approximate them [52], [53], and Bayesian neural networks, which treat the network's parameters as random variables and infer posterior distributions [54], [55], offer valuable insights into model uncertainty. These approaches can significantly assist in improving the trustworthiness of deep learning methods in remote sensing tasks [56], [57], [58].

Monte Carlo dropout is another method used for uncertainty estimation in predictive models. It leverages dropout to approximate Bayesian inference for deep neural networks by performing multiple forward passes with dropout during inference [59]. Each pass generates different predictions, allowing for the calculation of prediction variance and capturing the inherent epistemic uncertainty in the model's output. It can also be used to prevent overfitting [60]. This technique has been applied successfully in various domains, such as computer vision [61], [62], natural language processing [63], and healthcare [64].

One of the key benefits of Monte Carlo dropout is its potential to enhance prediction interpretability [65]. By generating multiple predictions with dropout, the method provides a probabilistic distribution of possible outcomes, enabling a more comprehensive understanding of the model's uncertainty. This distribution can be visualized and analyzed to gain insights into the factors influencing the model's decisions. Monte Carlo dropout has found applications in a wide range of tasks. Uncertainty estimation helps to identify ambiguous regions in image classification tasks, or it can guide the system to seek clarification or avoid providing incorrect or misleading information. Moreover, Monte Carlo dropout has been utilized to understand the level of confidence in the model's predictions and assisting in making informed decisions [66].

III. METHODOLOGY

A. Convolutional Neural Networks

ResNet-50 [67] and Xception [68] are two widely acclaimed and standard CNNs that find extensive usage in various scientific domains, including remote sensing image classification tasks. ResNet-50, short for residual network with 50 layers, revolutionized the field of deep learning by introducing residual connections that mitigate the vanishing gradient problem and enable the training of extremely deep networks. This architecture facilitates the construction of deeper models, leading to improved accuracy in image classification tasks. On the other hand, Xception, an extension of the Inception architecture, takes the concept of depth-wise separable convolutions to an extreme level. It separates the spatial and channel-wise convolutions, reducing the computational cost significantly while maintaining high performance.

In our study we use both, ResNet-50 and Xception in order to introduce our Slum Transfer network (STnet), a custom CNN specifically designed to excel in processing high-resolution

remote sensing imagery. The STnet is a heavily customized Xception network [68] and a simplified schematic can be seen in Fig. 2. The entry flow consists of five convolution combinations using residual skip connections. In order to capture a larger area when using high-resolution remote sensing imagery, the first two 2-D convolutions use large 9x9 kernels. In the middle flow, feature pyramid pooling is used to provide a unified framework to extract features at different scales. Finally, the classification flow is composed of two linear functions. Throughout the whole STnet, a combination of batch normalization and dropout layers afterwards are used. In total, STnet has 22 layers and 3.3 million trainable parameters.

B. Transfer Learning

The learning strategy employed in this procedure can be divided into two distinct phases. In the initial phase, the STnet undergoes pretraining on a class-imbalanced dataset denoted as $\mathcal{D}_{\text{base}}$. To address the class imbalance during this stage, we employ a weighted loss, as illustrated in (1), to give due importance to underrepresented classes and make the most of the available data. Subsequently, the STnet is transfer learned using an additional dataset, referred to as $\mathcal{D}_{\text{loocv}}^{\text{bal}}$. However, one of the classes in $\mathcal{D}_{\text{base}}$ is significantly imbalanced compared to the others, while in $\mathcal{D}_{\text{loocv}}^{\text{bal}}$, a class balanced dataset is created using undersampling. $\mathcal{D}_{\text{loocv}}^{\text{bal}}$ is designed to be class balanced, meaning it contains an equal number of images from all classes. By ensuring that each class is represented equally in $\mathcal{D}_{\text{loocv}}^{\text{bal}}$, we mitigate the bias toward the imbalanced class from $\mathcal{D}_{\text{base}}$. This balanced dataset allows for a fair and unbiased transfer-learning process, as each class contributes equally to the training of the new classifier.

During pretraining and transfer learning, we use a class weighted cross entropy loss L as seen in (1) where w_i is the weight for each class, scaled by the inverted count of the class occurrence

$$L(x, c, w) = - \sum_i w_i \cdot y'_i \cdot \log \left(\frac{\exp(x_i)}{\sum_j \exp(x_j)} \right)$$

where

- w_i : weight for class i ;
- y'_i : target distribution after label smoothing for class i ;
- x_i : logit for class i .

(1)

During transfer learning, the complete CNN remains trainable, and no layers are frozen. This means that all the layers of the pretrained CNN, are trained using the $\mathcal{D}_{\text{loocv}}$ dataset. By keeping all layers trainable, the CNN can adapt its learned features to the new dataset while still benefiting from the knowledge gained on the base dataset. This approach allows the CNN to capture task-specific features from $\mathcal{D}_{\text{loocv}}^{\text{bal}}$ while retaining the general knowledge acquired from $\mathcal{D}_{\text{base}}$.

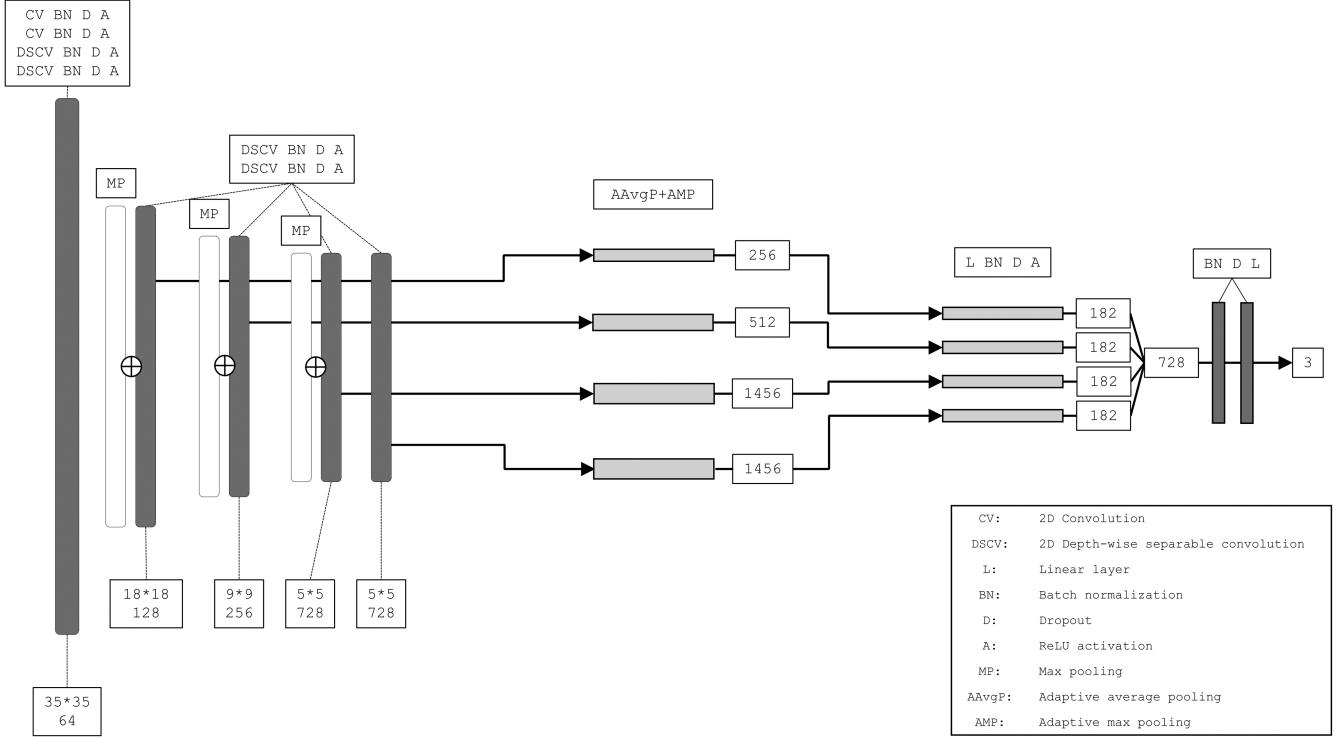


Fig. 2. Simplified schematic of the STnet architecture, comprising five convolutional variants in the entry -flow, succeeded by feature pyramid pooling layers and a classification- flow in the end. This light-weight architecture encompasses 3.3 million trainable parameters.

C. Monte Carlo Dropout for Uncertainty Estimation

In our classification setting, we have a dataset $D(X, Y)$, where $X = x_1, x_2, \dots, x_n$ represents the records of input images and $Y = y_1, y_2, \dots, y_n$ denotes the corresponding reference labels. We employ our STnet model to predict new outputs \bar{y} from new data \hat{x} . The model's predictions rely on a set of weights, and the task at hand involves finding the optimal set of these weights through an optimization problem

$$\bar{y} = \frac{1}{T} \sum_{t=1}^T y^{(t)}$$

where

\bar{y} : Averaged prediction over Monte Carlo runs;

T : Total number of Monte Carlo runs;

$y^{(t)}$: Prediction for the t th forward pass. (2)

To incorporate the Monte Carlo dropout technique as seen in (2), we use a probability $p = 0.3$ for each dropout layer, and for each model in all our experiments. This decision was informed by the preliminary test with $p = 0.1$, $p = 0.3$, and $p = 0.5$, where $p = 0.3$ offered the most effective balance between the Monte Carlo probabilities and the accuracies of the models.

During the forward pass, a unit is dropped and set to zero if its corresponding binary variable is zero. By utilizing Monte Carlo dropout, we aim to model the distribution, and subsequently, the predictive posterior distribution of \bar{y} . Notably, we can achieve

this by training the neural network as if it were a typical network, with the inclusion of dropout layers after each layer with weight parameters and performing T predictions.

In summary, unlike the conventional classification setting where a single prediction $y^{(t)}$ is obtained, the Monte Carlo dropout technique allows us to model a predictive distribution. This approach entails training the network with dropout layers and making multiple predictions, resembling the training process of a standard neural network with slight modifications.

IV. DATA AND EXPERIMENTAL SETUP

A. Dataset

The remote sensing data used in this study were acquired using PlanetScope satellites during 2021. In total, 8-bit RGB data were used and all scenes were resampled to 3-m resolution per pixel. Data from eight cities of the Global South were collected including Cape Town, Caracas, Lagos, Medellin, Mumbai, Nairobi, Rio de Janeiro, and Sao Paulo. The division of the remote sensing data into $88 * 88$ pixel patches (equivalent to $264 \text{ m} * 264 \text{ m}$) was methodically chosen based on empirical evidence from previous studies in the domain of learning with few samples, which demonstrated the efficacy of this specific patch size [9], [49].

Our dataset consists of three target classes: zero background, one formal built-up areas, and two slums. The formal built-up areas were derived by using data from the LCZ42 dataset [26]. Reference data for the slum settlements were created by mapping polygons from experts in the field of remote

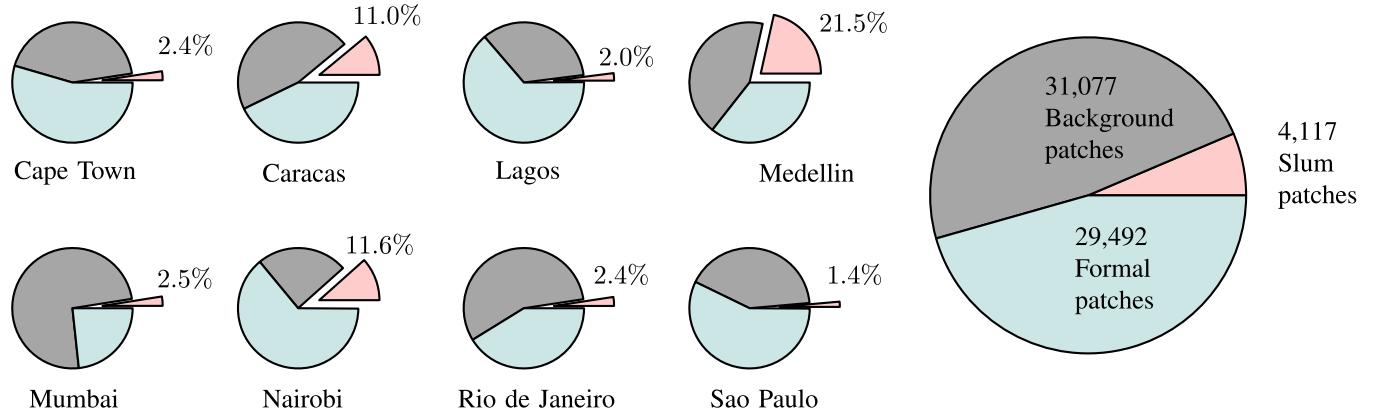


Fig. 3. Class distribution in eight cities and combined distribution. The figure displays nine pie charts depicting the class distribution in eight cities, with the slum sample proportion highlighted for each city. The final pie chart showcases the combined distribution, illustrating the overall class proportions across all cities.

sensing and urban poverty on the basis of up to date aerial imagery using Google Earth. To ensure data consistency for the reference data gathered from all sources, all polygons were checked and if necessary adjusted by the authors.

Each image patch used for training and testing the AI model has a dimension of $88 * 88 * 3$, with each label patch is $88 * 88 * n_{cl}$, where $n_{cl} = 3$ for the three classes used. If a reference patch contains at least 25% pixels of slum settlements, it is considered toward the slum class, patches with less than 25% but containing at least one slum pixel are discarded during training the model. For all other samples, the class with the highest pixel tally is considered as the main class. In total, 64 686 samples are available in the $\mathcal{D}_{\text{original}}$ dataset used for training and testing our approach as seen in Fig. 3.

B. Data Sampling

We define $\mathcal{D}_{\text{original}}$ in (3) as the set of ordered pairs, where each pair consists of an image X and CL as its corresponding city's location. X_n is the n th image in the set and CL_i as the location of the n th image, where i ranges between the city's location ID from 1 to 8. This dataset contains a wide range of diverse samples, encompassing various morphologies of urban patterns relevant to our topic.

For all experiments, we use a leave-one-out cross-validation approach, which is instrumental in ensuring comprehensive model evaluation and robustness across diverse urban environments, reflecting the variability in slum morphologies. This method also effectively mitigates the risk of overfitting, ensuring the model's adaptability and generalizability to different geographical contexts, crucial for the real-world application of urban poverty analysis and slum mapping. The image patches from seven of the eight cities are used for training and validation, while the remaining city's dataset is used for testing and transfer-learning. This process is repeated for all eight cities creating eight pretrained models to use for the test datasets. We partitioned $\mathcal{D}_{\text{original}}$ into two distinct datasets as seen in (4). The first subset, named $\mathcal{D}_{\text{base}}$, was employed for pretraining the STnet. $\mathcal{D}_{\text{base}}$ served as the foundation for training the initial weights

and learning representations, the dataset always consist of seven cities of the dataset as seen in (5). The second subset, called $\mathcal{D}_{\text{loocv}}$ in (6), was dedicated to the transfer-learning phase. By using a leave-one-out cross-validation dataset $\mathcal{D}_{\text{loocv}}$, we were able to refine and optimize the STnet's performance, ensuring its adaptability and robustness. Overall, the division of $\mathcal{D}_{\text{original}}$ into $\mathcal{D}_{\text{base}}$ and $\mathcal{D}_{\text{loocv}}$ played a crucial role in our research, enabling us to achieve accurate and reliable results.

During the transfer-learning phase, the dataset $\mathcal{D}_{\text{loocv}}$ is turned into a class balanced dataset $\mathcal{D}_{\text{loocv}}^{\text{bal}}$, using undersampling of the majority class. In (7) X_n is the n th image in the dataset with its corresponding label Y_n . We count the occurrence of all classes c and randomly sample j patches used for transfer learning.

$$\mathcal{D}_{\text{original}} = \{(X_1, \text{CL}_1), (X_2, \text{CL}_2), \dots, (X_n, \text{CL}_i)\} \quad (3)$$

$$\mathcal{D}_{\text{original}} = \mathcal{D}_{\text{base}} \cup \mathcal{D}_{\text{loocv}} \quad (4)$$

$$\mathcal{D}_{\text{base}} = \{(X_1, \text{CL}_i), \dots, (X_n, \text{CL}_i)\} \in \mathcal{D}_{\text{original}} \quad (5)$$

$$\text{CL}_i \neq \text{loocv}$$

$$\begin{aligned} \mathcal{D}_{\text{loocv}} &= \{(X_1, \text{CL}_i), \dots, (X_n, \text{CL}_i)\} \in \mathcal{D}_{\text{original}} \\ \text{CL}_i &= \text{loocv} \end{aligned} \quad (6)$$

$$\begin{aligned} \mathcal{D}_{\text{loocv}}^{\text{bal}} &= \left\{ (X_n, Y_n) \mid \text{count}(Y_n \in \mathcal{D}_{\text{loocv}}, Y_n = c_1) = j, \right. \\ &\quad \text{count}(Y_n \in \mathcal{D}_{\text{loocv}}, Y_n = c_2) = j, \\ &\quad \left. \text{count}(Y_n \in \mathcal{D}_{\text{loocv}}, Y_n = c_3) = j \right\}. \end{aligned} \quad (7)$$

To evaluate the number of image patches required for transfer learning, we examine 1, 5, 10, 25, 50, and 100 image samples per class. For each experiment, we randomly select these samples per class from $\mathcal{D}_{\text{loocv}}$, and use the remaining city, not included in the training dataset, for transfer learning. The samples chosen for transfer learning are subsequently eliminated from the test dataset. This process ensures that our experiments avoid bias and accurately reflect the model's capability to generalize

from limited data. In order to address the effects of randomly choosing image samples, we averaged the outcomes of five differently seeded experiments and report the standard deviations in our results, highlighting the impact of sample selections on the model's performance. In order to guarantee that there are sufficient samples of each class, particularly the slum class, 100 samples were the maximum number of samples required to verify our transfer-learning strategy.

C. Experimental Setup

To examine the impact of transfer learning on noisy datasets, we follow the setup outlined as follows. In all experiments, we warm up the optimizer for three epochs with a learning rate of $1e - 8$. For pretraining, we use a learning rate of $1e - 3$ and for transfer learning $1e - 4$. All experiments use an Adam optimizer and weighted soft cross entropy loss. In addition, a batch size of 16 is used for training. To tackle both, dataset noise and model prediction uncertainty, we employ Monte Carlo dropout. This technique involves obtaining an average of 25 outputs from the model's predictions. We compute the average of the raw logits produced by the model and calculate the corresponding entropy value in order to compare the level of uncertainty of the prediction.

In our evaluation framework, it is important to note that while our models were trained on three classes to effectively manage class (im-)balance, the accuracy metrics reported specifically pertain to the slum class. This focused approach is due to our primary interest in slum mapping. Classes representing background and urban/formal built-up areas are not included in the accuracy assessment. Therefore, in assessing performance, we use three commonly used metrics for image classification problems, namely the F1-score, precision, and recall, as our primary metrics to gauge the effectiveness of our models in accurately identifying slum areas. To further compare the efficiency of different models, we analyze the training time required for each. In addition, we assess the influence of Monte Carlo steps on our results, examining how variations in this parameter impact the models' stability and inference time. By integrating both performance metrics and computational efficiency measures, we ensure a thorough evaluation that guides our decision making process and optimizes the overall quality of our outcomes.

A fundamental challenge in the context of transfer learning is the variability in model performance when using a limited number of samples. This variability arises due to the random selection of training samples, leading to potential sample selection bias. To obtain a comprehensive understanding of model performance and address the issues arising from outlier data training, it is imperative to employ a rigorous approach. Specifically, we conduct five seeded runs to effectively assess the models' capabilities. By averaging the results obtained from these diverse seeded runs, we obtain a robust estimation of model performance, which allows for a more accurate representation of sample selection bias. This approach aids in reducing the impact of random fluctuations, providing a clearer picture of the model's general performance across varying training data subsets.

TABLE I
RESULTS FOR EIGHT CITIES COMPARING DIFFERENT NUMBER OF SAMPLES USED FOR TRANSFER LEARNING THE STNET, INCLUDING THE STANDARD DEVIATION FOR FIVE SEDED RUNS

| Samples | F1 | Precision | Recall |
|-----------|------------------------------------|------------------------------------|------------------------------------|
| Inference | $0.7201 \pm .11$ | $0.6941 \pm .10$ | $0.7788 \pm .19$ |
| 1 | $0.7324 \pm .09$ | $0.7312 \pm .11$ | $0.7706 \pm .17$ |
| 5 | $0.7806 \pm .05$ | $0.7626 \pm .06$ | $0.8091 \pm .10$ |
| 10 | $0.8082 \pm .05$ | $0.7830 \pm .06$ | $0.8500 \pm .09$ |
| 25 | $0.8358 \pm .04$ | $0.8250 \pm .05$ | $0.8541 \pm .08$ |
| 50 | $0.8432 \pm .05$ | $0.8558 \pm .07$ | $0.8447 \pm .10$ |
| 100 | $0.8624 \pm .05$ | $0.8718 \pm .06$ | $0.8638 \pm .10$ |

V. RESULTS

A. Transfer-Learning Results

The results of the transfer-learned STnet reveal an empirical relationship between the number of samples per class used for transfer-learning and the corresponding F1-score as seen in Table I. Notably, an increase in the number of samples yielded improved F1-scores. However, it is noteworthy that even with just a single sample per class, the model achieved commendable F1-scores of 73.24%. Nevertheless, after 50 samples, the F1-score seems to plateau, suggesting an upper limit of high 80% F1-score for this classification task. These findings indicate the potential for achieving favorable F1-score with STnet, even when training data are scarce. The highest F1-score of 86.24% was achieved when using 100 samples per class for transfer-learning.

In addition, when examining the precision and recall values in Table I of the transfer-learned STnet, notable patterns emerge. While the precision values increases more drastically as the number of samples for transfer learning increases, the recall values, however, only steadily increases. These results underscore the effectiveness of the transfer-learning approach in refining the model's precision and recall, leading to improved overall performance and indicating the potential of STnet in applications with limited training data.

Fig. 4 depicts the F1-scores for eight cities and the corresponding number of samples used to transfer learn our STnet model. The general trend observed in the figure indicates that as the number of samples per class used for transfer learning increases, the F1-scores also increase. The experiment was conducted five times, with each transfer-learning approach utilizing different random samples. The error band in Fig. 4 from the five runs uses confidence intervals of 95% to draw around estimated values.

In Cape Town (93.60%), Caracas (90.62%), and Medellin (91.09%), we achieve the highest F1-scores, when using 100 samples for transfer learning. But it needs to be noted that in Medellin and Caracas, we already achieve high accuracies using simple inference of over 82.10%. We also observe a decrease in F1-score when only one sample per class is used for transfer learning, in Caracas, Medellin, Lagos, and Mumbai, indicating a more challenging setting for transfer learning. But even in

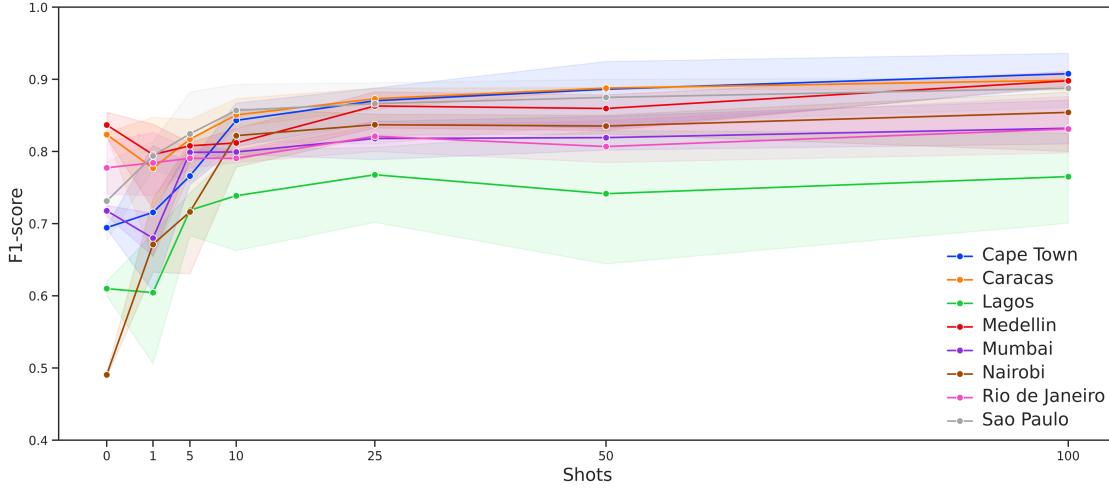


Fig. 4. F1-score for eight cities using a variety of number of samples to transfer learn the STnet pretrained using a transfer-learning approach.

TABLE II
COMPARISON OF F1-SCORES FOR STNET, XCEPTION, AND RESNET50,
AVERAGED OVER FIVE DIFFERENTLY SEEDED RUNS SHOWN WITH ITS
STANDARD DEVIATIONS

| Samples | F1-score | | |
|-----------|---------------------|--------------|---------------------|
| | STnet | Xception | ResNet50 |
| Inference | 0.7201 ± .11 | 0.6724 ± .16 | 0.7150 ± .09 |
| 1 | 0.7324 ± .09 | 0.7309 ± .08 | 0.7010 ± .12 |
| 5 | 0.7806 ± .05 | 0.7804 ± .05 | 0.7811 ± .05 |
| 10 | 0.8082 ± .05 | 0.8031 ± .06 | 0.7941 ± .04 |
| 25 | 0.8358 ± .04 | 0.8380 ± .06 | 0.8553 ± .04 |
| 50 | 0.8432 ± .05 | 0.8502 ± .04 | 0.8709 ± .04 |
| 100 | 0.8624 ± .05 | 0.8775 ± .05 | 0.8957 ± .02 |

All models employed 25 Monte Carlo iterations.

Lagos and Mumbai, using only five samples per class results in a major improvement of roughly 10% in F1-score compared to simple inference.

B. Comparing Various CNNs

In Table II, we conduct a comprehensive comparative analysis of the F1-scores for three distinct CNNs: STnet, Xception, and ResNet50. This comparison spans a range of scenarios in transfer learning, starting from simple inference results to the use of 1–100 image samples in transfer-learning processes. Each model's F1-score is calculated as an average across five independently seeded runs, and we provide the standard deviations to illustrate the variability in performance. In addition, all models were subjected to 25 Monte Carlo iterations to ensure consistency in our evaluation methodology. Our analysis reveals that STnet, despite having a considerably lower parameter count of only 3.3 million, achieves performance metrics that are comparable to those of Xception and ResNet-50, which are significantly more parameter intensive. Notably, in scenarios where only a limited

TABLE III
COMPARING DIFFERENT CNN ARCHITECTURES TO EACH OTHER BASED ON
THEIR SIZE AND TRAINING TIME

| | Parameters | Training time | |
|--|------------|---------------|-------------------|
| | | per step | Total time |
| STnet | 3.3m | 33.25it/sec. | 56:34 36 epochs |
| Xception | 20.8m | 21.28it/sec. | 1:22:45 24 epochs |
| ResNet50 | 23.5m | 17.85it/sec. | 1:15:37 20 epochs |
| All models employed 25 Monte Carlo iterations. | | | |

number of samples are employed for transfer learning, STnet demonstrates superior performance, outscoring both Xception and ResNet-50.

Table III provides a detailed comparison of the training times for each step and the total time required to achieve the best validation metric for the three models. Although the overall F1-scores of these CNNs are relatively similar, a significant difference is observed in their training durations. This discrepancy is largely attributed to STnet's more streamlined architecture, which makes it considerably lighter and faster in processing compared to Xception and ResNet-50, as evidenced in Table III. Notably, STnet not only demonstrates faster processing times but also requires less total time to attain the optimal validation metric. However, it is important to note that despite its shorter overall training duration, STnet demands more epochs to reach the best model fitness, in contrast to Xception and ResNet-50. This aspect highlights the efficiency of STnet in terms of time management.

C. Comparing Monte Carlo Dropout Rates

In Table IV, we investigate the impact of varying the number of Monte Carlo dropout test runs on our STnet model. The performance of the model is evaluated using inference time, F1-score, and finally, the entropy value, which is a measure of uncertainty or randomness within the predicted distributions

TABLE IV
COMPARISON OF STNET'S INFERENCE TIME, F1-SCORE, AND ENTROPY
ACROSS DIFFERENT MONTE CARLO DROPOUT ITERATIONS, TRAINED ON 100
SAMPLES PER CLASS

| Monte Carlo iterations | Inference time | F1-score | Entropy |
|------------------------|---------------------------|----------|---------|
| 1 | 31.59 it/sec. 2:11 min | 0.8423 | — |
| 5 | 20.43 it/sec. 3:20 min | 0.8515 | 0.7845 |
| 25 | 5.68 it/sec. 12:02 min | 0.8624 | 0.7832 |
| 50 | 3.10 it/sec. 22:17 min | 0.8679 | 0.7810 |

generated by the Monte Carlo dropout technique. Specifically, we compare the results obtained from using 1, 5, 25, and 50 Monte Carlo dropout test runs. In [69], 50 iterations are mentioned when using Monte Carlo dropout, but they only used a dropout layer in the last layer of their CNN. Since STnet uses dropout throughout its complete architecture, we test iterations of up to 50.

Our findings reveal that increasing the number of Monte Carlo dropout test runs leads to a slight improvement in the F1-score. Furthermore, we observe a slight decrease in the entropy values as the number of Monte Carlo dropout test runs increases, which implies that the predictions become more focused and certain as more test runs are performed. Significant observations were made regarding the inference time when increasing the number of Monte Carlo dropout iterations. The results indicate a substantial increase in inference time, with a 275% rise observed when transitioning from five Monte Carlo dropout iterations to 25, followed by an additional 84% increase when reaching 50 iterations. Despite the availability of insightful uncertainty measurements with just five iterations, the experiments conducted in this study employed 25 iterations as the preferred configuration for analysis.

VI. DISCUSSION

A. Uncertainty of Slums

Fig. 5 shows the results obtained for all cities using the transfer-learned STnet with 100 samples per class. The incorporation of Monte Carlo dropout as a method for uncertainty estimation unveils a significant advantage. It allows us to discern the STnet's level of certainty in predicting the location of slum settlements and identifies cases where its predictions are inconclusive. This not only provides crucial insights into the decision-making process of the STnet but also sheds light on the inherent challenges associated with the classification of slum areas.

The analysis demonstrates that the STnet exhibits elevated confidence in predicting the presence of typical slum

settlements, characterized by typical morphologic slum features, including high density, heterogeneous building patterns, and irregular road shapes. This pattern is evident in cities that achieve high F1-scores, namely Cape Town, Caracas, Medellin, and Mumbai.

In addressing the challenges faced in slum classification within specific cities, it is observed that Lagos presents notable difficulties with underclassification of slums. Conversely, in Nairobi, Rio de Janeiro, and Sao Paulo, the primary challenge lies in overclassification. These issues are largely due to two key factors. The first factor is the absence of distinct morphological features typically found in slum settlements, which are otherwise noticeable in cities like Caracas and Medellin. The second factor contributing to these classification challenges is the presence of formal settlement structures that share similarities with slum areas in terms of density and low-rise characteristics. This overlap in physical attributes complicates the task of clearly differentiating between formal and slum classes in these urban environments.

This highlights the complexity of slum classification due to local morphologic specifics in relation to the surrounding built-up morphologies as well as it emphasizes the importance of taking into account differences in morphological characteristics present within slums. Moreover, in fringe regions, where slum settlements are intertwined with urban formal settlements, vegetation areas, or both, higher uncertainties are observed.

In regards to assessing the uncertainty of slums and their prediction, evaluating the chosen dropout value during training and Monte Carlo inference becomes a crucial aspect of our methodology. The decision to implement a 30% dropout rate was a strategic one, aimed at striking an optimal balance in our models. This rate was selected after observing effects of different dropout rates in preliminary tests. At a lower 10% dropout rate, we noticed less variability in uncertainties, but this did not significantly enhance the models' accuracies. On the other hand, a higher dropout rate of 50% adversely impacted the models' accuracies, suggesting a potential overadjustment in the learning process.

This understanding of the impacts of varying dropout rates was important in optimizing the performance of our models. By settling on a 30% dropout rate, we managed to maintain a balance where the accuracy of the models was not overly compromised, nor was the effectiveness of the Monte Carlo estimation diluted. This decision was crucial in ensuring that our models remained robust and efficient in predicting slum areas.

It is crucial to acknowledge the influence of the inherently noisy dataset on our results. Although we have unified the dataset into a coherent representation of slums, as detailed earlier in this article, the intra- and interurban variability inherently introduces a significant level of noise. This variability means there is a wide range of slum characteristics to learn and predict. However, this diversity also serves as a key advantage of employing the Monte Carlo Dropout method. By using this technique, we can observe the effects of this variability in the probabilities, which is also evident in the maps presented in Fig. 6.

Furthermore, it is important to consider how the application of our models to different definitions of morphological slums

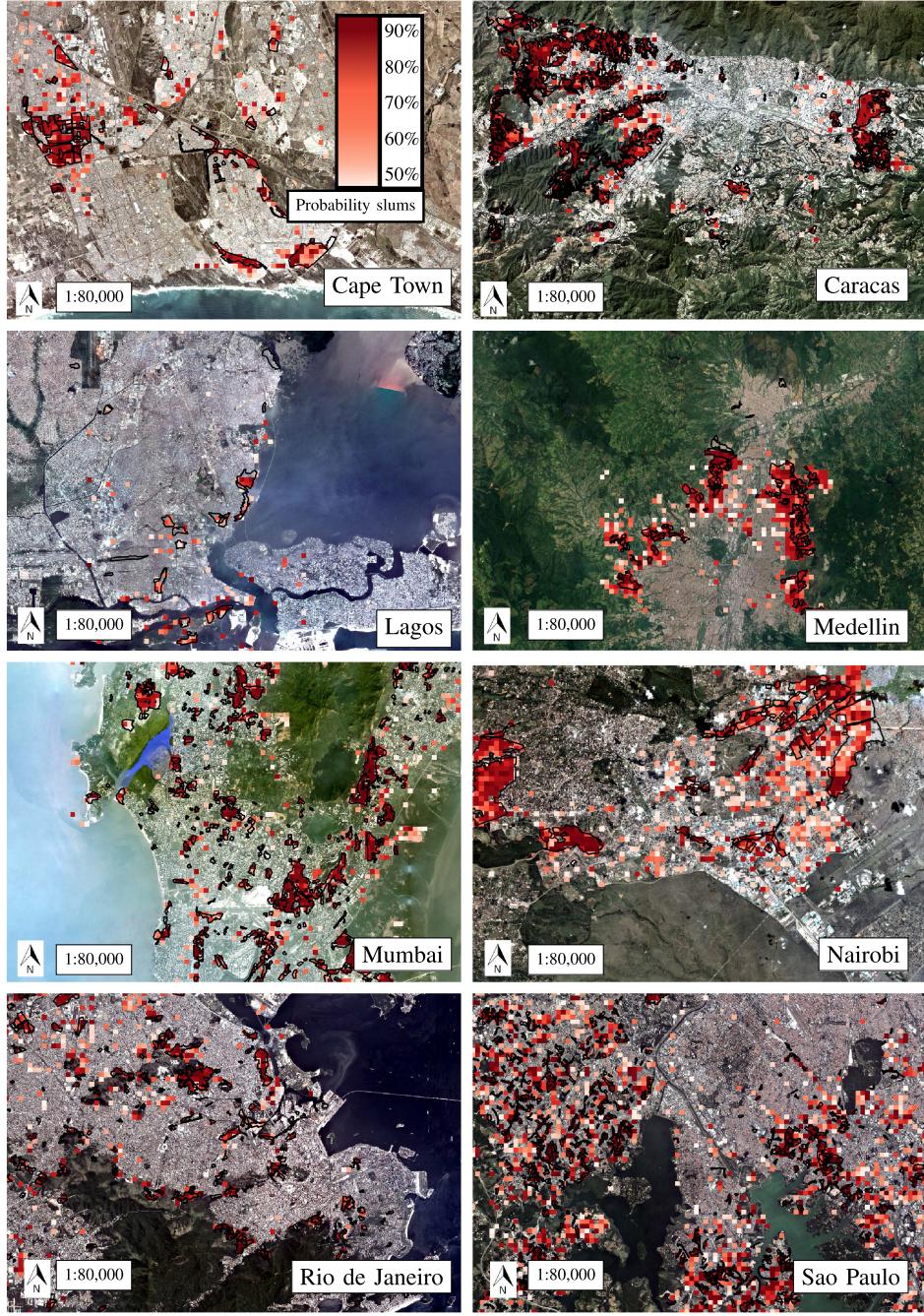


Fig. 5. Results for all eight cities using the transfer-learned STnet trained on 100 samples per class. All results are in the same scale of 1:80 000 and use the same color bar for the probability value of the slum class. Black outlines are used for the reference slum polygons.

could impact the results. Slums can vary greatly in their physical characteristics, spatial distributions, and overall appearances from one urban area to another. If our models were applied to slum areas with different morphological characteristics than those on which they were trained, this could potentially lead to variations in predictive accuracy and uncertainty estimations. Such a transfer would require careful consideration and possibly adjustments to the model to account for these differences. This aspect underscores the importance of context and adaptability in model application, especially in diverse urban environments.

B. Transfer Learning With Few Samples

In Fig. 6, we present the results obtained for the STnet within a similar area of interest as depicted in Fig. 1. To provide additional clarity, we have outlined the slum reference polygons with a black border. Furthermore, we present the slum probability results obtained from the five different training techniques using the same red colorbar. These results shed light on the model's performance in identifying slum settlements. All images (a)–(f) within this figure are consistently displayed at a scale

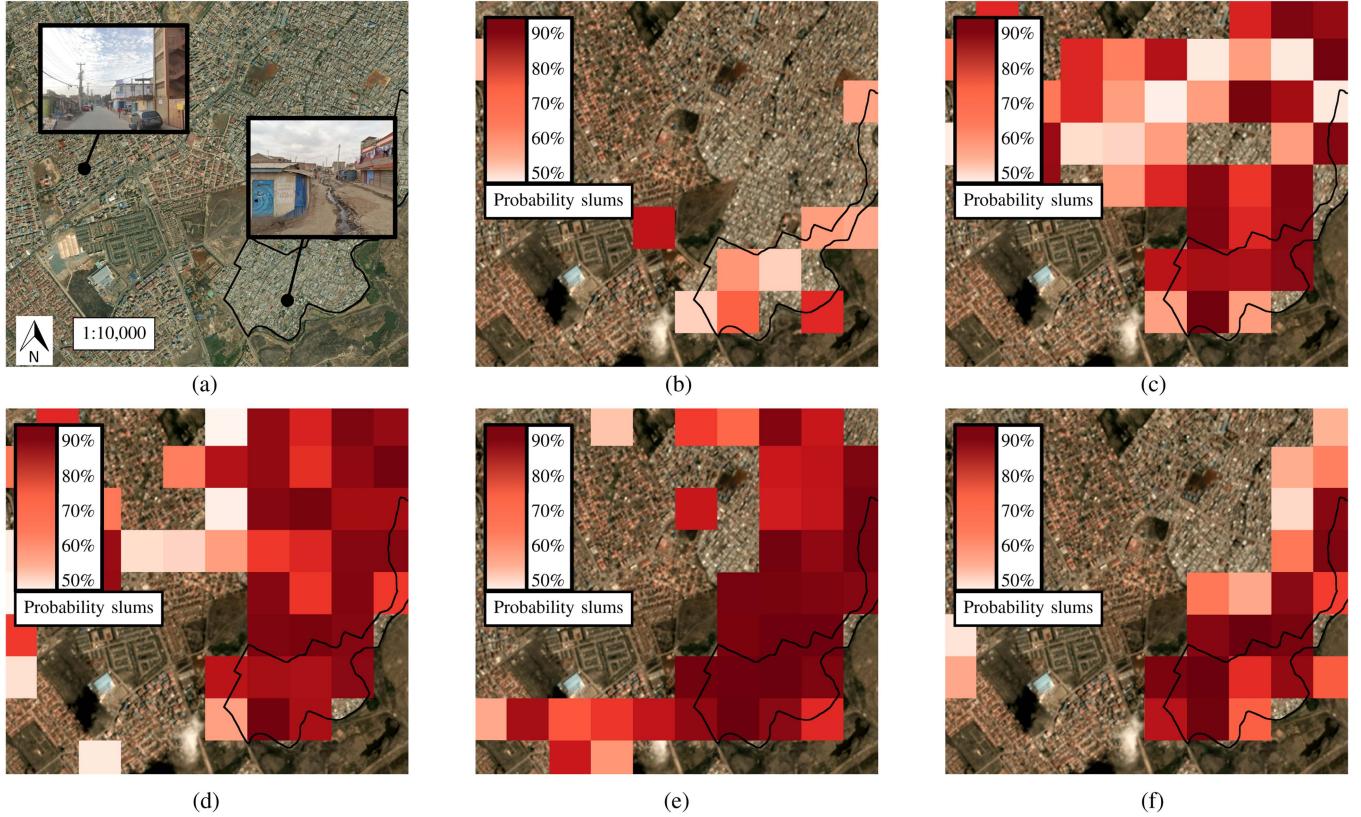


Fig. 6. Results for the STNet in a comparable area of interest, as depicted in Fig. 1. All images (a)–(f) are presented in a consistent scale of 1:10 000. Image (a) showcases a very high-resolution Google satellite imagery of the identical point of interest shown in Fig. 1. Images (b)–(f) exhibit the outcomes obtained using the STNet, with variations from no transfer learning (b) to transfer learning from 1 to 50 samples per class (c)–(f). (a) Google Satellite basemap. (b) No transfer learning. (c) One sample. (d) Five samples. (e) Ten samples. (f) Fifty samples.

of 1:10 000. Fig. 6(a) offers a detailed view, featuring very high-resolution Google satellite imagery of the exact point of interest showcased in Fig. 1. The subsequent images, Fig. 6(b) through 6(f), illustrate the diverse outcomes achieved through the utilization of the STnet. Fig. 6(b) presents results obtained without the application of transfer learning, while images Fig. 6(c) through 6(f) demonstrate the progressive impact of transfer learning with 1 to 50 samples, highlighting the evolution of performance and insights gained through this process. The variation in results for Nairobi, transitioning from utilizing 50 samples per class to 100 samples per class for transfer learning, exhibits negligible differences in both accuracy metrics and visual outcomes. Consequently, we conclude the figure at the 50 sample mark, as further iterations do not yield significant improvements in performance or visual representation. By leveraging transfer learning, we aim to improve the model's ability to recognize and understand the unique features of Nairobi's urban landscape.

From Fig. 4, we find a comprehensive overview of the STnet's performance metrics for the entire city of Nairobi. Specifically, we evaluate the model's F1-score. When employing simple inference without transfer learning, the F1-score achieved was as low as 49.06%, indicative of an initial struggle to map the slums of Nairobi. While Fig. 6(b), initially presents promising results with minimal overclassification tendencies, it is essential to

consider the broader context. The depicted area represents only a small portion of the dataset. What is particularly noteworthy is the relatively low confidence values associated with these predictions. This underscores the significance of considering local context, which becomes evident that the models using transfer learning displays higher confidence in its classifications, emphasizing the value of leveraging transfer learning to enhance the classification accuracy and contextual understanding.

However, as we incorporate one sample per class for transfer learning, we observe a notable improvement, with the F1-score rising to 66.78%. This demonstrates the efficiency of using a limited number of labeled samples to enhance the model's understanding of Nairobi's unique morphologic characteristics. In Fig. 6(c), we observe a significant increase in the F1-score for the entire city of Nairobi. However, this improvement is accompanied by a notable issue of overclassification in the area of interest. In addition, there is an evident rise in the overconfidence levels of the predictions, highlighting a disparity between quantitative scores and visual accuracy. From Fig. 6(c) to 6(e), there is a noticeable progression in the F1-score. However, it is not until Fig. 6(f), when a sufficient number of samples are utilized for transfer learning, that the visual outcomes demonstrate considerable improvement. In this instance, the results are promising, exhibiting only minor instances of over- and underclassification.

As discussed in Section I, low-rise and dense settlement structures do not necessarily equate to slum settlements. The region depicted in Fig. 6 exemplifies this challenge, containing only one slum settlement amidst several dense formal settlements. This blend of characteristics intensifies the difficulty of accurate classification. Furthermore, these findings hold broader implications, suggesting that our results are highly generalizable to other cities with similar fuzzy feature spaces between formal, low-rise dense settlements and slum settlements. Cities like Lagos, Rio de Janeiro, and São Paulo, known for their similar morphological appearances of slums, can especially benefit from these insights, as they present comparable classification challenges.

VII. CONCLUSION

Through the integration of Monte Carlo dropout, we gained valuable insights into the uncertainties in our predictions, allowing us to identify areas where our AI model is more or less certain in its slum classification. The presence of multiple typical slum morphologies led to higher certainty in the model's predictions. However, challenges arose when slums shared features with formal areas, which made the classification task more complex. Despite this, the application of Monte Carlo dropout proved to be effective, especially when dealing with noisy datasets and fuzzy feature spaces, which typically pose significant challenges for any classification tasks.

Moreover, we introduced our custom CNN STnet, which demonstrated comparable results to renowned models like ResNet50 and Xception while offering significantly reduced processing time. We have successfully attained an elevated F1-score of 86.24%, a performance that can be deemed remarkable in the context of slum mapping, where we address intricate urban patterns and challenges. Particularly noteworthy was its performance when trained on limited samples, making it an ideal choice for scenarios with fewer available training data. We were able to outscore both Xception and ResNet50 when using ten or fewer samples per class for transfer learning. By combining Monte Carlo dropout, a class-weighted loss function for pretraining, and class-balanced transfer learning, we presented a simple yet efficient approach for accurately classifying challenging urban patterns in noisy and imbalanced datasets. Our approach not only addressed the uncertainties in slum classification but also tackled the inherent complexities of working with real-world data, which often lacks perfect labels and may exhibit imbalances across classes. In summary, our research provides a valuable contribution to the field of urban pattern classification and demonstrates the importance of considering uncertainties in AI models for more accurate and robust predictions. The proposed framework opens avenues for future research in improving the understanding of slum settlements and urban planning, ultimately leading to more effective and targeted interventions in urban development and poverty alleviation.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

DATA AVAILABILITY

The implementation of all steps in order to reproduce and recreate these results are available through GitHub starkt/UncertaintySlumDetection. The slum data, given its sensitive ethical nature, will be shared exclusively upon reasonable request. Regrettably, the PlanetScope remote sensing data cannot be shared due to copyright restrictions. However, GitHub repository offers resampled Sentinel-2 RGB imagery, serving as a representative example for many of our results.

AUTHORS' CONTRIBUTIONS

T. Stark, M. Wurm, H. Taubenböck, and X. X. Zhu designed the scope of the study. T. Stark was responsible curating the dataset; X. X. Zhu and team prepared the PlanetScope remote sensing data; T. Stark was responsible for methods and experiments. T. Stark took the lead in writing the manuscript, and M. Wurm, H. Taubenböck, and X. X. Zhu reviewed the manuscript.

REFERENCES

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [2] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Berlin, Germany: Springer, 2019.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [4] T. Gebru et al., "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, pp. 86–92, 2021.
- [5] C. M. Gevaert, M. Carman, B. Rosman, Y. Georgiadou, and R. Soden, "Fairness and accountability of AI in disaster risk management: Opportunities and challenges," *Patterns*, vol. 2, no. 11, 2021, Art. no. 100363.
- [6] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1521–1528.
- [7] B. Kellenberger, D. Marcos, and D. Tuia, "Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning," *Remote Sens. Environ.*, vol. 216, pp. 139–153, 2018.
- [8] Q. Li, Y. Chen, and P. Ghamisi, "Complementary learning-based scene classification of remote sensing images with noisy labels," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8021105.
- [9] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenböck, "Detecting challenging urban environments using a few-shot meta-learning approach," in *Proc. Joint Urban Remote Sens. Event*, 2023, pp. 1–4.
- [10] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
- [11] M. Schmitt, S. A. Ahmadi, and R. Hänsch, "There is no data like more data—Current status of machine learning datasets in remote sensing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 1206–1209.
- [12] C. Geiß, J. Maier, E. So, and Y. Zhu, "LSTM models for spatiotemporal extrapolation of population data," in *Proc. IEEE Joint Urban Remote Sens. Event*, 2023, pp. 1–4.
- [13] C. Qiu, X. Tong, M. Schmitt, B. Bechtel, and X. X. Zhu, "Multilevel feature fusion-based CNN for local climate zone classification from Sentinel-2 images: Benchmark results on the So2Sat LCZ42 dataset," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 2793–2806, 2020.
- [14] D. Stiller et al., "Spatial parameters for transportation," *J. Transport Land Use*, vol. 14, no. 1, pp. 777–803, 2021.
- [15] T. Fisher et al., "Uncertainty-aware interpretable deep learning for slum mapping and monitoring," *Remote Sens.*, vol. 14, no. 13, 2022, Art. no. 3072.
- [16] R. Engstrom, D. Pavelesku, T. Tanaka, and A. Wambile, "Mapping poverty and slums using multiple methodologies in Accra, Ghana," in *Proc. Joint Urban Remote Sens. Event*, 2019, pp. 1–4.
- [17] M. Kuffer et al., "The scope of Earth-observation to improve the consistency of the SDG slum indicator," *ISPRS Int. J. Geo- Inf.*, vol. 7, no. 11, 2018, Art. no. 428.

- [18] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, 2016.
- [19] M. Wurm and H. Taubenböck, "Detecting social groups from space—Assessment of remote sensing-based mapped morphological slums using income data," *Remote Sens. Lett.*, vol. 9, no. 1, pp. 41–50, 2018.
- [20] H. Taubenböck and N. Kraff, "The physical face of slums: A structural comparison of slums in Mumbai, India, based on remotely sensed data," *J. Housing Built Environ.*, vol. 29, no. 1, pp. 15–38, 2014.
- [21] C. Mood and J. O. Jonsson, "The social consequences of poverty: An empirical test on longitudinal data," *Social Indicators Res.*, vol. 127, pp. 633–652, 2016.
- [22] H. Taubenböck, N. J. Kraff, and M. Wurm, "The morphology of the arrival city—A global categorization based on literature surveys and remotely sensed data," *Appl. Geogr.*, vol. 92, pp. 150–167, 2018.
- [23] O. Gruebner et al., "Mapping the slums of Dhaka from 2006 to 2010," *Dataset Papers Sci.*, vol. 2014, p. 172182, 2014, doi: [10.1155/2014/172182](https://doi.org/10.1155/2014/172182).
- [24] M. Kuffer, K. Pfeffer, and R. Sliuzas, "Slums from space—15 years of slum mapping using remote sensing," *Remote Sens.*, vol. 8, no. 6, 2016, Art. no. 455.
- [25] N. J. Kraff, M. Wurm, and H. Taubenböck, "Uncertainties of human perception in visual image interpretation in complex urban environments," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4229–4241, 2020.
- [26] X. Zhu, C. Qiu, J. Hu, Y. Shi, Y. Wang, and M. Schmitt, "NEW: So2Sat LCZ42," 2019.
- [27] M. Wurm, H. Taubenböck, M. Weigand, and A. Schmitt, "Slum mapping in polarimetric SAR data using spatial features," *Remote Sens. Environ.*, vol. 194, pp. 190–204, 2017.
- [28] R. Engstrom, J. S. Hersh, and D. L. Newhouse, "Poverty from space: Using high-resolution satellite imagery for estimating economic well-being," World Bank Policy Research Working Paper 8284, 2017.
- [29] R. Prabhu and B. Parvathavarthini, "An enhanced approach for informal settlement extraction from optical data using morphological profile-guided filters: A case study of Madurai city," *Int. J. Remote Sens.*, vol. 42, no. 17, pp. 6688–6705, 2021.
- [30] N. Mudau and P. Mhangara, "Investigation of informal settlement indicators in a densely populated area using very high spatial resolution satellite imagery," *Sustainability*, vol. 13, no. 9, 2021, Art. no. 4735.
- [31] C. Persello and A. Stein, "Deep fully convolutional networks for the detection of informal settlements in VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2325–2329, Dec. 2017.
- [32] D. Verma, A. Jana, and K. Ramamritham, "Transfer learning approach to map urban slums using high and medium resolution satellite imagery," *Habitat Int.*, vol. 88, 2019, Art. no. 101981.
- [33] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenböck, "Satellite-based mapping of urban poverty with transfer-learned slum morphologies," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5251–5263, 2020.
- [34] J. Friesen, H. Taubenböck, M. Wurm, and P. F. Pelz, "The similar size of slums," *Habitat Int.*, vol. 73, pp. 79–88, 2018.
- [35] S. Park, J. Lim, Y. Jeon, and J. Y. Choi, "Influence-balanced loss for imbalanced visual classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 735–744.
- [36] M. S. Reza and J. Ma, "Imbalanced histopathological breast cancer image classification with convolutional neural network," in *Proc. IEEE 14th Int. Conf. Signal Process.*, 2018, pp. 619–624.
- [37] V. H. Barella, L. P. Garcia, M. P. de Souto, A. C. Lorena, and A. de Carvalho, "Data complexity measures for imbalanced classification tasks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.
- [38] M. Burduja and R. T. Ionescu, "Unsupervised medical image alignment with curriculum learning," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 3787–3791.
- [39] Y. Huang et al., "Curricularface: Adaptive curriculum learning loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5901–5910.
- [40] E. Elyan, C. F. Moreno-Garcia, and C. Jayne, "Cdsmote: Class decomposition and synthetic minority class oversampling technique for imbalanced-data classification," *Neural Comput. Appl.*, vol. 33, pp. 2839–2851, 2021.
- [41] A. Ali-Gombe, E. Elyan, and C. Jayne, "Multiple fake classes GaN for data augmentation in face image dataset," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.
- [42] J. Kim, J. Jeong, and J. Shin, "M2m: Imbalanced classification via major-to-minor translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13896–13905.
- [43] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, and F. Herrera, "Dynamic ensemble selection for multi-class imbalanced datasets," *Inf. Sci.*, vol. 445–446, 2018, pp. 22–37.
- [44] 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 J. Photogrammetry Remote Sens.*, vol. 150, pp. 59–69, 2019.
- [45] N. Karim, U. Khalid, N. Meeker, and S. Samarasinghe, "Adversarial training for face recognition systems using contrastive adversarial learning and triplet loss fine-tuning," 2021, *arXiv:2110.04459*.
- [46] S. O. Ngesthi, I. Setyawan, and I. K. Timotius, "The effect of partial fine tuning on Alexnet for skin lesions classification," in *Proc. 13th Int. Conf. Inf. Technol. Elect. Eng.*, 2021, pp. 147–152.
- [47] S. Hershey et al., "The benefit of temporally-strong labels in audio event classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 366–370.
- [48] M. Saini and S. Susan, "Vggin-Net: Deep transfer network for imbalanced breast cancer dataset," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 1, pp. 752–762, Jan./Feb. 2022.
- [49] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9062–9071.
- [50] Y. Li, Z. Shao, X. Huang, B. Cai, and S. Peng, "Meta-FSEO: A meta-learning fast adaptation with self-supervised embedding optimization for few-shot remote sensing scene classification," *Remote Sens.*, vol. 13, no. 14, 2021, Art. no. 2776.
- [51] E. Koukouraki, L. Vanneschi, and M. Painho, "Few-shot learning for post-earthquake urban damage detection," *Remote Sens.*, vol. 14, no. 1, 2022, Art. no. 40.
- [52] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.
- [53] G. Yang, H.-C. Li, W. Yang, K. Fu, T. Celik, and W. J. Emery, "Variational Bayesian change detection of remote sensing images based on spatially variant Gaussian mixture model and separability criterion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 3, pp. 849–861, Mar. 2019.
- [54] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, vol. 37, pp. 1613–1622.
- [55] M. Li, A. Stein, and K. M. De Beurs, "A Bayesian characterization of urban land use configurations from VHR remote sensing images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 92, 2020, Art. no. 102175.
- [56] M. Rußwurm, M. Ali, X. X. Zhu, Y. Gal, and M. Körner, "Model and data uncertainty for satellite time series forecasting with deep recurrent models," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 7025–7028.
- [57] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," 2022, *arXiv:2107.03342*.
- [58] P. Ebel, V. S. F. Garnot, M. Schmitt, J. D. Wegner, and X. X. Zhu, "UnCrtaintTS: Uncertainty quantification for cloud removal in optical satellite time series," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2023, pp. 2085–2095.
- [59] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, vol. 48, pp. 1050–1059.
- [60] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," 2015, *arXiv:1511.06068*.
- [61] Z. Li, T. Zhang, S. Cheng, J. Zhu, and J. Li, "Stochastic gradient Hamiltonian Monte Carlo with variance reduction for Bayesian inference," *Mach. Learn.*, vol. 108, pp. 1701–1727, Sep. 2019.
- [62] A. Lemay et al., "Improving the repeatability of deep learning models with Monte Carlo dropout," *NPJ Digit. Med.*, vol. 5, Nov. 2022, Art. no. 174.
- [63] Y. Xiao and W. Y. Wang, "Quantifying uncertainties in natural language processing tasks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 7322–7329.
- [64] T. J. Loftus et al., "Uncertainty-aware deep learning in healthcare: A scoping review," *PLoS Digit. Health*, vol. 1, pp. 1–15, 2022.
- [65] S. Ma, J. Huang, Y. Xie, and N. Yi, "Identification of breast cancer prognosis markers using integrative sparse boosting," *Methods Inf. Med.*, vol. 51, no. 2, pp. 152–161, 2012.
- [66] A. Jungo et al., "On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation," in *Proc. 21st Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, Granada, Spain, Sep. 16–20, 2018, pp. 682–690.

- [67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.
- [68] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
- [69] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.



Thomas Stark received the M.Sc. degree in geodesy and geoinformation in 2018 from the Technical University of Munich, Munich, Germany, where he is currently working toward the Ph.D. degree on the topic of "Towards detecting global urban poverty" with the Department of Aerospace and Geodesy, Data Science in Earth Observation.

In 2017, he joined the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Weßling, Germany, as a Research Associate. His current research interests include encompass a robust and dynamic exploration of the field of computer vision, and delving into its multifaceted intricacies. A particular area of profound fascination lies in his keen interest in harnessing the potential of remote sensing data, which adds an extra layer of dimensionality to his investigations. This interest stems from a genuine passion for unraveling the hidden insights concealed within vast datasets, a pursuit driven by a profound commitment to the UN Sustainable Development Goals. With an unwavering dedication to these global objectives, he aspires to pave the way for a more sustainable and equitable future by ingeniously applying the principles of computer vision and adept data analysis techniques.



Michael Wurm received the diploma degree (Mag. rer. nat.) in geography with a specialization in remote sensing, GIS, and spatial research from the University of Graz, Graz, Austria, in 2007, and the Ph.D. degree (Dr. rer. nat.) in surveying and geoinformation from the Graz University of Technology, Graz, in 2013.

He was with the Institute of Digital Image Processing, Joanneum Research, Graz, in 2007. In 2008, he joined the University of Würzburg, Germany, where he was involved in interdisciplinary research between Earth observation data and social sciences. Since 2011, he has been with the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Weßling, Germany. In 2022, he became the Head of the "City and Society" team where he is involved in topics on urban geography, urban remote sensing, and urban morphology, and slum mapping research. Since 2013, he has been a Lecturer with the University of Graz.



Xiao Xiang Zhu (Fellow, IEEE) received the M.Sc., Dr.Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She was the Founding Head with the Department "EO Data Science," Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen, Germany. She was a Guest Scientist or a Visiting Professor with the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, The University of Tokyo, Tokyo, Japan, and the University of California at Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. Since 2019, she has been a Co-Coordinator with the Munich Data Science Research School, Munich. Since 2019, she has been heading the research field "Aeronautics, Space, and Transport" with Helmholtz Artificial Intelligence. Since May 2020, she has been the Principal Investigator (PI) and the Director with the International Future AI lab "AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond," Munich. Since October 2020, she has been serving as the Director with the Munich Data Science Institute (MDSI), TUM, where she is currently the Chair Professor of data science in Earth observation. Her main research interests include remote sensing and Earth observation, signal processing, machine learning, and data science, with their applications in tackling societal grand challenges, e.g., global urbanization, United Nations (UN's) Sustainable Development Goals (SDGs), and climate change.

Dr. Zhu is a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin-Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She serves on the scientific advisory board of several research organizations, including the German Research Center for Geosciences (GFZ) and the Potsdam Institute for Climate Impact Research (PIK). She is an Associate Editor of *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*. She serves as an Area Editor responsible for special issues of *IEEE Signal Processing Magazine*.



Hannes Taubenböck received the diploma degree in geography from the Ludwig-Maximilians Universität München, Munich, Germany, in 2004, the Ph.D. (Dr.rer.nat.) degree in geography from the Julius Maximilian's University of Würzburg, Würzburg, Germany, in 2008, and the Habilitation degree in geography from the University of Würzburg, Würzburg, in 2019.

In 2005, he joined the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Weßling, Germany. After a postdoctoral research phase with the University of Würzburg (2007–2010), he returned in 2010 to DLR-DFD as a Scientific Employee. Since 2022, he has been a Professor with the Institute of Geography and Geology, University of Würzburg, for the working group Earth observation. Since 2022, he has also been the Head of the Geo-Risks and Civil Security Department, DFD. His current research interests include urban remote sensing topics, from the development of algorithms for information extraction to value adding to classification products for findings in urban geography.