

Deprived Area Mapping using Deep Learning on Sentinel-2 Imagery

Dr. Ryan Engstrom
Data Science

*George Washington Univseristy
Washington, D.C., United States*
rengstro@gwu.edu

Dr. Amit Jafari
Data Science

*George Washington Univeristy
Washington, D.C., United States*
ajafari@email.gwu.edu

Arathi Nair
Data Science

*George Washington Univseristy
Washington, D.C., United States*
arathinair@gwmail.gwu.edu

Abstract— Majority of populations in low- and middle-income countries (LMIC) live in deprived neighborhoods characterized by poverty and substandard living conditions. The policy efforts of local-to-international stakeholders to provide sustainable development requires accurate and scalable methods to map deprived areas across LMIC cities. Therefore, measuring poverty levels for efficient allotment of financial aids and resources becomes an important step to help poverty reduction. The data collected by conducting surveys for mapping remote or developing cities not only require huge financial investments but can also be time and labor intensive. To overcome this challenge, we utilize open-source Sentinel-2 images that will help us bridge the data gap and provide us adequate data for mapping deprived areas cost-effectively. In this project, we propose an Autoencoder - unsupervised learning framework using Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) based on deep neural network that will enable us to identify and classify deprived neighborhoods in Lagos (Nigeria), Accra (Ghana) and Nairobi (Kenya). The model will be trained on unlabeled satellite imagery to learn hidden geographical features and tested for transferability of model's learnings to predict and map deprived areas in other cities. Furthermore, we aim to build a model deployment pipeline that can be adapted to map poverty at a global scale using transfer learning.

Keywords— Deprived area mapping, deep learning, unsupervised learning, remote sensing, autoencoders, transfer learning

I. INTRODUCTION

In recent years, rapid urbanization and population growth has led to uneven city expansions and poor infrastructure development. This is being experienced especially in low-and middle-income countries (LMIC) which makes up a large share of the world's population consisting of a diverse group by size, population, and income levels. According to a report from the United Nations (UN), the global population will reach 9.7 billion by 2050, and 68% of population will live in urban areas. The rapid growth in population will undoubtedly result in socioeconomic inequalities in developing countries. In most cases, these cities lack basic infrastructure and substandard of living due to rapid growth of slum-like communities.

Urban sustainable development is an increasing concern worldwide, and global institutions like United Nations and World Bank are centering their Sustainable Development Goals to tackle the urbanization challenges in these countries. Sustainable Development Goal 1 ('End poverty in all its forms everywhere') and Sustainable Development Goal 11 ('Make cities and human settlements inclusive, safe, resilient and sustainable') aims to enable LMIC cities to tackle poverty and better plan for its future population. To

facilitate these initiatives, decision-makers use neighborhood deprivation maps to estimate number of people living in these areas to allocate funds, plan, and evaluate policies for long term urban planning and governance.

Collecting relevant spatial and contextual information about these slum-like neighborhoods in LMIC becomes a crucial to detect the level of marginalization. Although, the pre-existing deprivation mapping techniques like national census and household survey data collection and manual field-based mapping provide information for area segmentation. These approaches pose shortcomings in terms of high financial investment and manpower required for implementation. Leveraging the usage of satellite imagery in remote sensing has the potential to evolve as a vital source for providing large wealth of hidden geographical data at a global scale with an advantage of being a cost-effective technology.

In this paper, we implement an unsupervised deep learning algorithm using Autoencoder for extracting geographical features using raw satellite imagery. This algorithm will perform image classification on remote sensing data using Google Earth Sentinel-2 images. These satellite images are preprocessed and extracted from Google Earth Engine in Sentinel-2 (100m) spatial resolution for three cities: Lagos (Nigeria), Accra (Ghana) and Nairobi (Kenya). Geographical data being highly unstructured, the autoencoder will extract important hidden features to learn and estimate poverty for these cities. Finally, we attempt to build a transfer learning framework using these pure satellite images for poverty prediction as an alternate approach to the expensive surveys currently implemented.

II. PROBLEM STATEMENT

The proposed deep learning model aims to address three key objectives.

A. Extract cloud-free Sentinel-2 image

First, the cloud-free Sentinel-2 satellite images are sourced from Google Earth Engine for each city using Sentinel Hub S2 cloudless machine learning algorithm that detects and eliminates clouds/shadows reflected on Earth's surface.

B. Train CNN & MLP Autoencoders

After the image extraction, we train Convolutional Neural Network and Multi-Layer Perceptron Autoencoders for image reconstruction by optimal learning and feature extraction from the provided 100m grid cells.

C. Image classification

Later, the model's performance is evaluated for its ability to classify labeled images in the existing training dataset which are divided into three classes: 0 - built-up area, 1 - deprived, 2 - non-built-up areas.

III. SOLUTION AND METHODOLOGY

A. Google Earth Engine

In order to integrate raw satellite images in the deep learning model, sourcing cloud free Sentinel-2 images from Google Earth Engine was one of the main challenges in data gathering and preprocessing. Clear satellite images will enable the model to learn the hidden geographical features of these cities much more accurately. This was done by implementing Sentinel Hub S2 cloudless machine learning algorithm which detects and masks the clouds/cloud shadows reflected on the earth's surface. The cloud mask algorithm uses two image collections from Google Earth Engine; Sentinel-2 (S2) surface reflectance (SR) collection is used to selected satellite images from a particular date range and Sentinel-2 cloud probability collection is used to identify clouds and shadows projected by low- reflectance near-infrared (NIR) pixels.

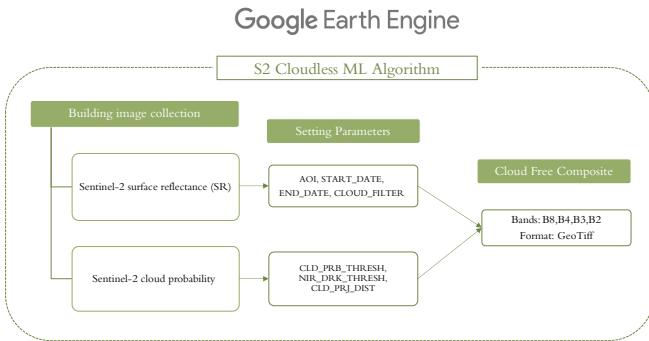


Figure 1. Framework of the Sentinel-2 cloud masking algorithm for satellite image extraction.

The parameters defined to filter the S2 image collection and determine cloud and cloud shadows identification were as follows.

Parameter	Type	Description
AOI	ee.Geometry	Area of interest
START_DATE	string	Image collection start date (inclusive)
END_DATE	string	Image collection end date (exclusive)
CLOUD_FILTER	integer	Maximum image cloud cover percent allowed in image collection
CLD_PRB_THRESH	integer	Cloud probability (%); values greater than are considered cloud
NIR_DRK_THRESH	float	Near-infrared reflectance; values less than are considered potential cloud shadow
CLD_PRJ_DIST	float	Maximum distance (km) to search for cloud shadows from cloud edges
BUFFER	integer	Distance (m) to dilate the edge of cloud-identified objects

The Sentinel-2 collection was developed by selecting area of interest (AOI) i.e., city's geographical coordinates to create area boundary and setting date parameters to capture s2cloudless images for each city. To add cloud probability layer on the derived images, cloud mask components were set to identify water pixels in the image by adding Scene Classification Layer (SCL) band and NIR_DRK_THRESH was fine tuned to determine the projection of cloud shadows in the images. The tables below show the optimal parameters that were set for cloud-shadow masking for all three cities.

LAGOS	
Surface Reflectance Collection	Cloud Probability Collection
START_DATE = '2021-12-20' END_DATE = '2022-01-23' CLOUD_FILTER = 60	START_DATE = '2021-12-20' END_DATE = '2022-01-23' CLOUD_FILTER = 60
CLD_PRB_THRESH = 65	CLD_PRB_THRESH = 60
NIR_DRK_THRESH = 0.80	NIR_DRK_THRESH = 1
CLD_PRJ_DIST = 2	CLD_PRJ_DIST = 6
BUFFER = 100	BUFFER = 100

[Github Code](#)

ACCRA	
Surface Reflectance Collection	Cloud Probability Collection
START_DATE = '2021-11-14' END_DATE = '2021-12-23' CLOUD_FILTER = 30	START_DATE = '2021-11-14' END_DATE = '2021-12-23' CLOUD_FILTER = 30
CLD_PRB_THRESH = 85	CLD_PRB_THRESH = 80
NIR_DRK_THRESH = 0.50	NIR_DRK_THRESH = 1
CLD_PRJ_DIST = 2	CLD_PRJ_DIST = 2
BUFFER = 65	BUFFER = 90

[Github Code](#)

NAIROBI	
Surface Reflectance Collection	Cloud Probability Collection
START_DATE = '2022-01-03' END_DATE = '2022-01-14' CLOUD_FILTER = 60	START_DATE = '2022-01-03' END_DATE = '2022-01-14' CLOUD_FILTER = 60
CLD_PRB_THRESH = 50	CLD_PRB_THRESH = 50
NIR_DRK_THRESH = 0.15	NIR_DRK_THRESH = 0.15
CLD_PRJ_DIST = 1	CLD_PRJ_DIST = 3
BUFFER = 50	BUFFER = 70

[Github Code](#)

This cloud detection algorithm detects the likelihood of clouds and shadows present in the given image collection and then mask the cloud pixels to replace them with cloud-free pixels from a different timeframe. Finally, all the mosaics were merged for obtaining a single cloud-free image. Shown below are the results of the cloud free images that will be used for training the autoencoder models for feature learning.

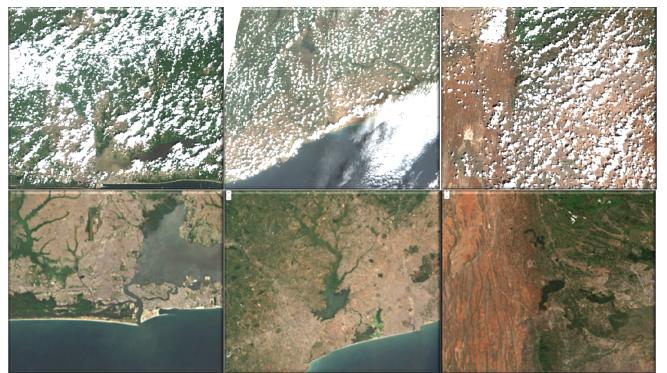


Figure 2. Cloud-free Sentinel-2 images for Lagos, Accra and Nairobi resp.

B. Training Autoencoder

The unsupervised autoencoder methods that we focus on in this paper are primarily intended to extract embedded geographical representations that can then serve as features for supervised classification. The above processed Google Earth images for the three cities were sliced into 10×10 pixels (100m) grid cells for training the autoencoders. A total of 1030089 images were clipped from Lagos, Accra, and Nairobi satellite images to input into the autoencoders.

1) Convolutional Neural Networks

The main task of the Convolutional (CNN) Autoencoder was to reconstruct the satellite images (10×10 pixels) inputted into the model and minimize reconstruction error by learning the optimal filters. A typical autoencoder consist of four parts: encoder, bottleneck, decoder, and reconstruction loss. The encoder reduces the high-dimensional input to the low-level code, which is called the bottleneck. From the bottleneck layer, the decoder attempts to reconstruct the data from the learned encoding. Finally, the reconstruction error (RE) measures how well the decoder performed in creating an output similar to the input.

Computation of the spatial dimension of the output size is represented by a function of the input size (W), the receptive field of the convolutional layer (F), the stride applied (S) and the amount of zero padding used (P). The formula used in determining the dimension of the feature maps in convolutional neural network are as follow.

$$\text{Encoder: } (W - F + 2P) / S + 1 \quad (1)$$

$$\text{Decoder: } S * (W - 1) + F - 2P \quad (2)$$

The autoencoder were trained using a total of 1030089 images in 100 m spatial size were inputted into the model after image normalization to fit each pixel values between 0 - 255 to match the RGB standard.

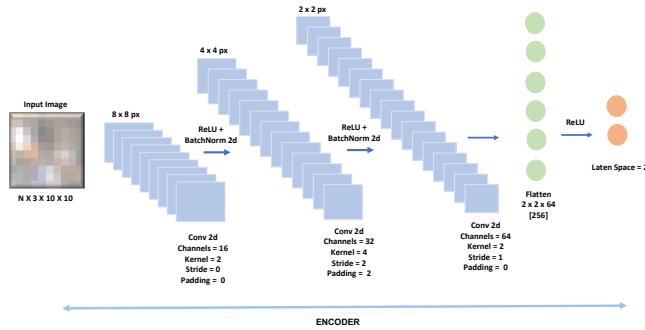


Figure 3. A chart of a CNN Encoder where the input image is passed through the model to reduce dimensionality. (GitHub code)

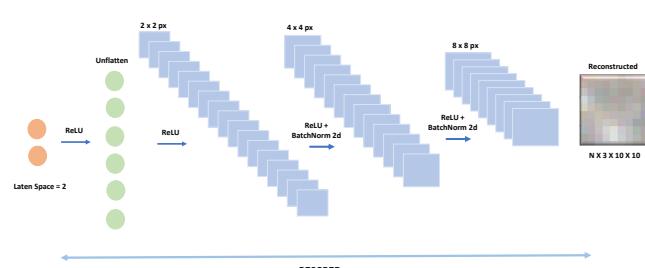


Figure 4. A chart of a CNN Decoder where the input image is passed through the model to reconstruct original image.

The above CNN autoencoder model was implemented using PyTorch. The satellite images in size 10×10 pixels with 3 channels (RGB) were passed into the encoder to compress the high dimensional images to a smallest possible dimension, embedding the content of the original image into the latent space of 2×2 pixels. This latent space vector is used by the decoder to reconstruct the compressed image back into 10×10 pixels with the lowest possible reconstruction error (RE). To obtain the reconstruction error (RE) between the original input and reconstructed output the model is trained to minimize the loss function mean squared error (MSE). In between each convolutional layer is an activation function ReLU (Rectified linear unit) conducting nonlinear transformation except for the last fully connected output layer which uses Sigmoid as an activation function.

To enforce robustness of the CNN autoencoder, it was trained as a denoised autoencoder. By adding white noise to the autoencoder we combat the tendency of the regular autoencoder to overfit and forces the model to learn important features from the satellite images. The graph below show the results obtained after training on satellite images for all three cities - Lagos, Accra and Nairobi.

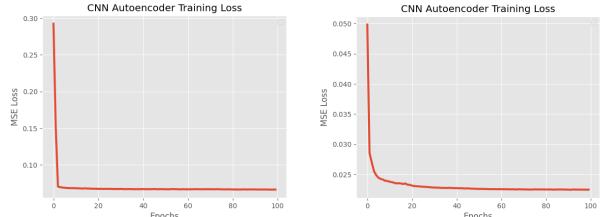


Figure 5. Training loss of the CNN Autoencoder over 100 epochs with batch size 32 & 10 respectively.

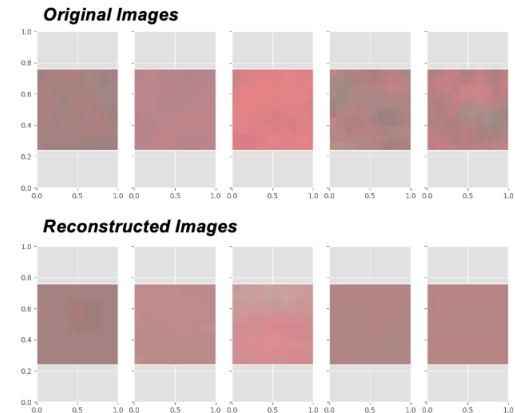


Figure 6. 10×10 pixel denoised image samples reconstructed by CNN Autoencoder.

The denoised CNN autoencoder trained over 100 epochs using a batch size 32 and the learning rate was set as 0.001. The reconstruction error was measured using Mean Squared Error (MSE). The results in Figure 5 show the lowest reconstruction error MSE produced for model trained with batch size 32 was of 0.06 and for batch size 10 was 0.02.

2) Multi-Layer Perceptron Neural Networks

Alternatively, a MLP autoencoder was also trained for the image reconstruction using linear neural networks. MLP autoencoder is represented as a feed forward neural network that consists of three layers nodes - an input, a hidden and an output layer. Each layer in a MLP is fully connected with

the next layer. In the hidden layers, each node is operated with a leaky rectified linear unit (LReLU) activation function that avoids vanishing gradient problem introduced by sigmoid and dying neuron problem caused by using ReLU. This enables the networks to distinguish and separate useful patterns from the satellite images. The model is trained to calculate the reconstruction error (RE) between the predicted output and the expected output values by minimizing the loss function MSE.

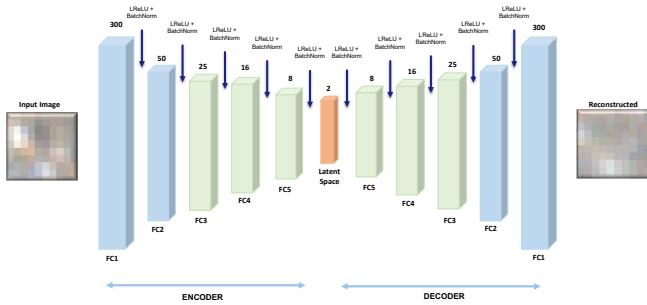


Figure 7. A chart of MLP autoencoder where the input image is passed through fully connected layers for image reconstruction.

Furthermore to build a robust model, white noise was added to the input image forcing the model to separate the important signals from high-dimensional data. The figures below show the results from the MLP autoencoder model which was implemented in PyTorch. The results were obtained after the training the autoencoder on satellite images for all three cities - Lagos, Accra and Nairobi.

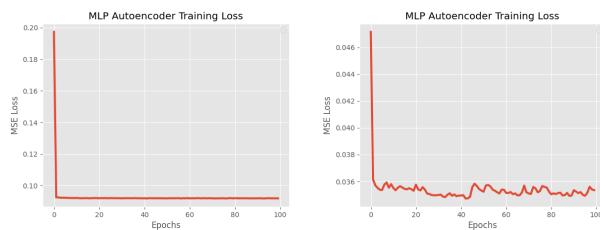


Figure 8. Training loss of the MLP Autoencoder over 100 epochs with batch size 32 & 10 respectively.

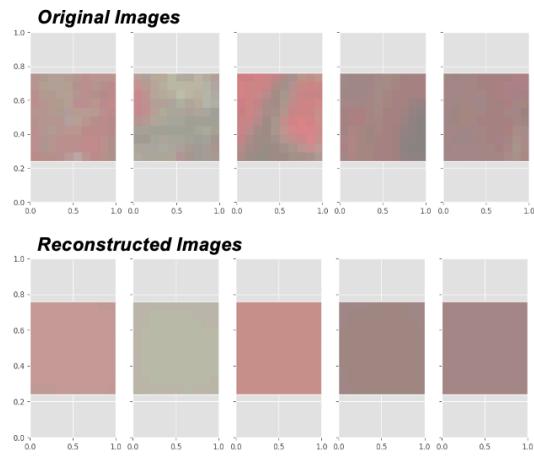


Figure 9. 10x10 pixel denoised image samples reconstructed by MLP Autoencoder,

The denoised MLP autoencoder produced the lowest MSE score of 0.09 for model trained with a batch size of 32 and 0.03 after training with the batch size of 10.

After training the CNN & MLP autoencoders, the encoder weights, and biases were used as a pretrained network which would act as a feature vector facilitating classification of deprived areas in the labeled training dataset.

C. Image Classification on Labeled Dataset

To address our research goal of identifying deprived areas for a given city, the developed pretrained autoencoder models were tested on the labeled training dataset. For this analysis, we have considered images for build-up (label - 0) and deprived (label - 1) areas. Images with label 2 have been ignored as they pertain to non-built-up areas like vegetation and water regions. The encoder models are used as a classifier along with the saved weights from pretrained autoencoders to attempt segregate the images into two categories i.e., 0 – built-up areas and 1 - deprived areas based on the learned geographical features. The training data was split into training and validation data to train the saved encoder model and then the accuracy of the autoencoder model was determined by evaluating it on the test data.

The evaluation of the pretrained models were conducted in two phases. First, the pertained CNN & MLP models are used to classify and predict deprived areas on the Lagos & Accra training dataset, Second, the results achieved for Lagos are compared with results obtained from CNN & MLP models trained to classify Lagos images without the use of pertained autoencoders.

It was also observed that the training dataset for Lagos was highly imbalanced. To address class imbalance, the deprived class was supplemented with additional images generated by image rolling and Variational Autoencoder (VAE) methods.

IV. RESULTS

The tables below show the performance of the trained encoder models from CNN and MLP autoencoders. The models were tested on the labeled training dataset of Lagos and Accra for its capability to detect and classify deprived areas from built-up areas in the dataset. Each dataset was divided into two image sets: one set was split into training (70%) and validation (30%) for training the encoder models for image classification and second set was purely used to test the model's prediction accuracy.

TABLE 1. IMAGE CLASSIFICATION ON LAGOS TRAINING DATASET

W/O Scheduler Batch: 32	CNN		MLP	
	Build-up (Class - 0)	Deprived (Class - 1)	Build-up (Class - 0)	Deprived (Class - 1)
F1 Score	0.93	0.17	0.83	0.06
Precision	0.99	0.10	0.98	0.03
Recall	0.87	0.61	0.71	0.43
Accuracy	0.87		0.71	

Confusion Matrix	CNN		MLP	
	Built-Up	Deprived	Built-Up	Deprived
	2095	306	2095	306
	21	33	21	33
	Built-Up	Deprived	Built-Up	Deprived

Macro Avg F1	0.55	0.44
Weighted Avg F1	0.91	0.81

W/O Scheduler Batch: 10	CNN		MLP	
	<i>Build-up (Class - 0)</i>	<i>Deprived (Class - 1)</i>	<i>Build-up (Class - 0)</i>	<i>Deprived (Class - 1)</i>
F1 Score	0.90	0.16	0.80	0.05
Precision	0.99	0.09	0.98	0.39
Recall	0.83	0.74	0.68	0.39
Accuracy	0.83		0.67	

Confusion Matrix	CNN		MLP	
	Built-Up	Deprived	Built-Up	Deprived
	1991	410	1622	779
	14	40	33	21
	Built-Up	Deprived	Built-Up	Deprived

Macro Avg F1	0.53	0.42
Weighted Avg F1	0.89	0.78

Confusion Matrix	CNN		MLP	
	Built-Up	Deprived	Built-Up	Deprived
	1912	489	1628	773
	15	39	28	26
	Built-Up	Deprived	Built-Up	Deprived

Macro Avg F1	0.51	0.43
Weighted Avg F1	0.87	0.79

Without Scheduler Batch: 10	CNN		MLP	
	<i>Build-up (Class - 0)</i>	<i>Deprived (Class - 1)</i>	<i>Build-up (Class - 0)</i>	<i>Deprived (Class - 1)</i>
F1 Score	0.87	0.12	0.78	0.05
Precision	0.99	0.06	0.98	0.03
Recall	0.78	0.67	0.64	0.44
Accuracy	0.78		0.64	

Confusion Matrix	CNN		MLP	
	Built-Up	Deprived	Built-Up	Deprived
	1870	531	18	36
	18	36	Built-Up	Deprived

Without Scheduler Batch: 10	CNN		MLP	
	Build-up (Class - 0)	Deprived (Class - 1)	Build-up (Class - 0)	Deprived (Class - 1)
MLP				
Confusion Matrix	1547	854	30	24
	Built-Up	Deprived	Built-Up	Deprived
Macro Avg F1	0.49	0.41		
Weighted Avg F1	0.86	0.76		

For Lagos, the CNN autoencoder model trained with batch size 32 over 100 epochs gave the best results with a F1 score of 93% for built-up area (class -1) and 17% for deprived area (class - 0). On the test set, the model was able to detect 2095 out of 2401 built-up area images and 33 out of 54 deprived images accurately.

Since the data for these two classes were imbalanced in the existing training dataset, additional images were added to the training dataset using data augmentation like image rolling, image shifting and oversampling along with generating new images using the autoencoder model

TABLE 2. IMAGE CLASSIFICATION ON LAGOS TRAINING DATASET WITH DATA AUGMENTATION

Without Scheduler Batch: 32	CNN		MLP	
	Build-up (Class - 0)	Deprived (Class - 1)	Build-up (Class - 0)	Deprived (Class - 1)
F1 Score	0.94	0.25	0.93	0.07
Precision	0.99	0.15	0.99	0.04
Recall	0.90	0.80	0.89	0.22
Accuracy	0.89		0.88	
CNN				
Confusion Matrix	2153	248	11	43
	Built-Up	Deprived	Built-Up	Deprived

Without Scheduler Batch: 32	CNN		MLP	
	Build-up (Class - 0)	Deprived (Class - 1)	Build-up (Class - 0)	Deprived (Class - 1)
MLP				
	2138	263	42	12
	Built-Up	Deprived	Built-Up	Deprived
Macro Avg F1	0.63	0.50		
Weighted Avg F1	0.78	0.91		

The results obtained after including additional data to train the classifier saw a significant increase in the F1 scores of the deprived class. The CNN autoencoder model trained on 10 batch size over 100 epochs achieved the highest F1 of 28%. On the test set, the model was able to detect 43 out of 54 images accurately.

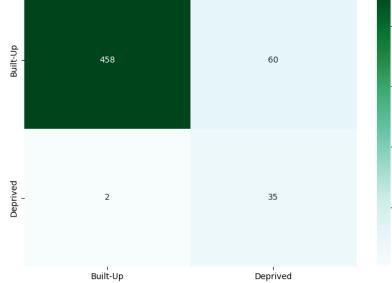
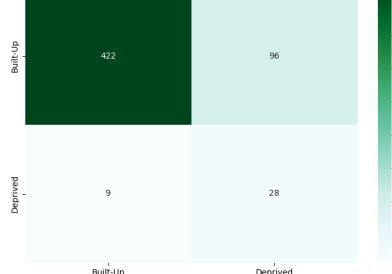
Furthermore, the result obtained by applying pretrained autoencoders were compared with the classification results obtained by CNN & MLP models implemented without pretrained autoencoder. The below table shows the model performances after including data augmentation. The results using both the methods (i.e., with and without pretrained autoencoder) produced close results with a minor improved performance by the pertained CNN autoencoder with a F1 score of 28%. However, the performance of the MLP model without pretrained autoencoder provided a better F1 score of 18 %.

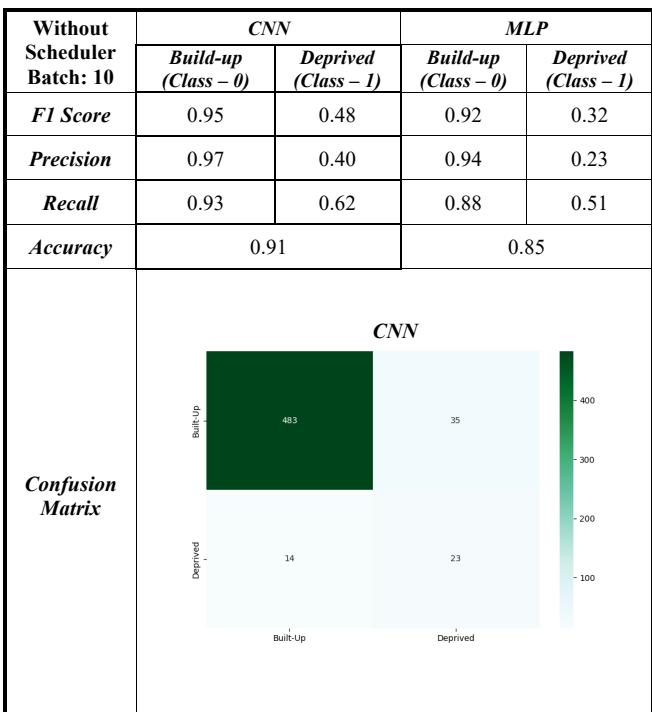
TABLE 3. IMAGE CLASSIFICATION ON LAGOS TRAINING DATASET COMPARISON WITHOUT PRETAINED AUTOENCODERS

Lagos	With Pretrained Autoencoder		Without Pretrained Autoencoder	
	CNN	MLP	CNN	MLP
	Deprived (Class - 1)	Deprived (Class - 1)	Deprived (Class - 1)	Deprived (Class - 1)
F1 Score	0.28	0.07	0.26	0.18
Precision	0.17	0.04	0.15	0.10
Recall	0.85	0.22	0.87	0.65
Macro Avg F1	0.61	0.50	0.58	0.55

Similarly, the pretrained CNN & MLP autoencoder models were tested on classifying the deprived class in the Accra training dataset.

TABLE 4. IMAGE CLASSIFICATION ON ACCRA TRAINING DATASET

Without Scheduler Batch: 32	<i>CNN</i>		<i>MLP</i>									
	<i>Build-up</i> (Class - 0)	<i>Deprived</i> (Class - 1)	<i>Build-up</i> (Class - 0)	<i>Deprived</i> (Class - 1)								
<i>F1 Score</i>	0.94	0.53	0.89	0.35								
<i>Precision</i>	1.00	0.37	0.98	0.23								
<i>Recall</i>	0.88	0.95	0.81	0.76								
<i>Accuracy</i>	0.89		0.81									
<i>Confusion Matrix</i>	<i>CNN</i>											
	 <table border="1" data-bbox="300 505 692 788"> <tr> <td></td> <td>Built-Up</td> <td>Deprived</td> </tr> <tr> <td>Built-Up</td> <td>458</td> <td>60</td> </tr> <tr> <td>Deprived</td> <td>2</td> <td>35</td> </tr> </table>		Built-Up	Deprived	Built-Up	458	60	Deprived	2	35		
	Built-Up	Deprived										
Built-Up	458	60										
Deprived	2	35										
	<i>MLP</i>											
	 <table border="1" data-bbox="300 860 692 1134"> <tr> <td></td> <td>Built-Up</td> <td>Deprived</td> </tr> <tr> <td>Built-Up</td> <td>422</td> <td>96</td> </tr> <tr> <td>Deprived</td> <td>9</td> <td>28</td> </tr> </table>		Built-Up	Deprived	Built-Up	422	96	Deprived	9	28		
	Built-Up	Deprived										
Built-Up	422	96										
Deprived	9	28										
<i>Macro Avg F1</i>	0.73		0.62									
<i>Weighted Avg F1</i>	0.91		0.85									



Without Scheduler Batch: 10	<i>CNN</i>		<i>MLP</i>													
	<i>Build-up</i> (Class = 0)	<i>Deprived</i> (Class = 1)	<i>Build-up</i> (Class = 0)	<i>Deprived</i> (Class = 1)												
	<p style="text-align: center;"><i>MLP</i></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 10px;">Build-Up</td> <td style="padding: 10px; text-align: center;">483</td> <td style="padding: 10px; text-align: center;">35</td> <td style="padding: 10px;"></td> </tr> <tr> <td style="padding: 10px;">Deprived</td> <td style="padding: 10px; text-align: center;">14</td> <td style="padding: 10px; text-align: center;">23</td> <td style="padding: 10px;"></td> </tr> <tr> <td></td> <td style="padding: 10px; text-align: center;">Build-Up</td> <td style="padding: 10px; text-align: center;">Deprived</td> <td></td> </tr> </table>				Build-Up	483	35		Deprived	14	23			Build-Up	Deprived	
Build-Up	483	35														
Deprived	14	23														
	Build-Up	Deprived														
<i>Macro Avg F1</i>	0.72		0.62													
<i>Weighted Avg F1</i>	0.92		0.88													

As shown in the tables above, the CNN model trained using batch size 32 over 100 epochs achieved the highest F1 score with 53% for detecting deprived class followed by MLP model trained using batch size 32 over 100 epochs with a F1 score of 35% for deprived class. On the test set, CNN model accurately detected 35 out of 37 images as deprived and MLP model accurately detect 28 out of 37 deprived images.

V. CONCLUSION

In this study we experimented with a novel concept of transfer learning using satellite imagery for mapping deprived areas in LMIC cities. The idea being to develop a self-learning unsupervised model to learn high level feature representations of the satellite images. We demonstrated the application of CNN and MLP Autoencoder models trained solely on cloud free satellite images for distinguishing deprived areas in Lagos and Accra without explicitly needing to collect any additional contextual feature or covariate data. Using the features learned by the autoencoder models, we were able to detect and estimate the deprived areas in these cities. The challenges of extracting complete cloud free satellite images and insufficient training data deprived class had to be addressed to develop a high performing transfer learning model and requires further improvements. However, the CNN autoencoder model provided with promising results for detecting deprived areas especially in the labeled training dataset for Accra. Further refinement in the cloud free satellite images and adding more data will immensely aid in improving the accuracy of the model predictions. Our approach has great potential to simplify remote sensing tasks and help solve global sustainability challenges.

VI. ACKNOWLEDGMENT

We acknowledge IdeaMaps for providing training data for this research paper. We would also like to thank for the support provided by the Department of Geography, George Washington University.

VII. REFERENCES

- [1] Sentinel-2 Cloud Masking with s2cloudless -
<https://developers.google.com/earth-engine/tutorials/community/sentinel-2-s2cloudless>
- [2] Evaluating the Ability to Use Contextual Features Derived from Multi-Scale Satellite Imagery to Map Spatial Patterns of Urban Attributes and Population Distributions -
<https://www.mdpi.com/2072-4292/13/19/3962>
- [3] The Role of Earth Observation in an Integrated Deprived Area Mapping “System” for Low-to-Middle Income Countries - <https://www.mdpi.com/2072-4292/12/6/982>
- [4] Deep Internal Learning for Inpainting of Cloud-Affected Regions in Satellite Imagery -
<https://www.mdpi.com/2072-4292/14/6/1342>
- [5] A Convolutional Autoencoder Topology for Classification in High-Dimensional Noisy Image Datasets -
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8622369/>
- [6] Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study -
<https://www.mdpi.com/1424-8220/21/23/8083>
- [7] Transfer Learning from Deep Features for Remote Sensing and Poverty Mapping -
https://www.researchgate.net/publication/282403466_Tran
- [8] Improving Image Autoencoder Embeddings with Perceptual Loss -
https://www.researchgate.net/publication/338547072_Improving_Image_Autoencoder_EMBEDDINGS_with_Perceptual_Loss
- [9] Extracting and Composing Robust Features with Denoising Autoencoders -
https://www.researchgate.net/publication/221346269_Extracting_and_composing_robust_features_with_denoising_autoencoders
- [10] Super Resolution for Noisy Images Using Convolutional Neural Networks

<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10677/1067710/Super-resolution-for-noisy-images-via-deep-convolutional-neural-network/10.1117/12.2297664.full?SSO=1>