SLUM MAPPING VIA REMOTE SENSING IMAGERY USING DEEP LEARNING

FINAL YEAR PROJECT REPORT

BY

Hamna Moieez

Syed Waleed Hyder

In Partial Fulfillment of the Requirements for the degree
Bachelors of Engineering in Software Engineering (BESE)

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST)
Islamabad, Pakistan

June 14, 2020

DECLARATION

We hereby declare that this project report entitled “Slum Mapping Via Remote Sensing Imagery Using Deep Learning” submitted to the “School of Electrical Engineering & Computer Science”, is a record of an original work done by us under the guidance of Supervisor “Dr. Muhammad Shahzad” and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

Team Members	**Signature**
Syed Waleed Hyder	Waleed
Hamna Moieez	Hamna
Advisor	**Signature**
Dr. Muhammad Shahzad	Dr SHAHZAD

DEDICATION

We would like to dedicate this thesis to our family and friends whose unparalleled support and encouragement helped us throughout the course of these four years. We would also like to dedicate this thesis to all the poor people who are living in the slums in substandard living conditions.

ACKNOWLEDGEMENT

We would like to express our gratitude and indebtedness to our project advisor Dr. Muhammad Shahzad and co-advisor Prof. Dr. Faisal Shafait whose expertise and invaluable guidance and insights allowed us to comprehensively complete this project and without whose guidance we would not have been able to reach our goal.

Additionally, we would also like to thank Prof. Dr. Xiaoxiang Zhu, Dr. Thomas Stark, and Lichao Mou at the Technical University of Munich (TUM), Munich, Germany for their support, guidance, and collaboration in our research activities.

ABSTRACT

Slum mapping is crucial to the modern explosive growth in population and urbanization. With a rapid increase in population, the need for housing and residence also proportionally increases. In countries where half of the population is living below the poverty line, it becomes increasingly difficult to accommodate the explosive growth in requirement. Therefore informal settlements, slums, pop up in areas across the country. In Pakistan, where the situation is similar, there are densely populated slums in almost every major city. The situation exacerbates when these slums are virtually hidden from the government either due to location, negligence, or on purpose. With no proper resources to sustain a healthy life, these slums are practically prisons situated in mainland areas where people are thrown to die.

In this thesis, we address this challenge by undertaking the extensive process of mapping and recording the slum areas. We visit the two major cities in the country, Karachi, and Islamabad, and figure out slum settlements along with their coordinates. We collect data from different satellites of those areas and perform exploratory analysis. We temporally analyze the data and perform segmentation analysis and mapping of these slums. In addition to this, we deploy working on a web-based platform and allow easy access of trained networks. We opensource the dataset and the code to make sure future research directions are open and further work can be done. Essentially, we conclude that such an automated pipelined system would allow regulatory authorities to pass necessary laws to rehabilitate the population of these areas.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Poverty has presented itself as one of the most pressing issues for our society over time. Poverty eradication is regarded as the foremost goal of Sustainable Development Goals (SDGs) developed by the United Nations (UN, 2010). Poverty not only hinders the collective growth of the country, but also halts the economic growth and sustenance. This stagnant development discombobulates the masses and paralyzes the government. One of the major factors to this explosive growth in poverty is increasing world population. Due to the lack of proper resources and settlement areas, people who can not afford a living start to settle in densely packed areas called slums. To add to the already devastated issue, these slums go unnoticed. Governments turn a blind eye either purposefully or unintentionally because generally they do not exist on the map. A country with an abundance of slums has a lot on its plate already, therefore it turns a blind eye towards this and does not take necessary steps to document such settlements and rehabilitate the people living there.

When there are a few number of such settlements or slums, it is controllable. But majority of the Asian countries have an abundance of slums which adds to the overall poverty rate. In addition to this, the crime rate in such informal settlements is always higher. Since these areas are virtually hidden, the inhabitants take it to themselves to run the day to day functioning. Therefore several major and minor crimes are on a constant rise.

All of these issues can be handled effectively if the government were able to localize, map and document these informal settlements, slums, and then can essentially plan for rehabilitation. The manual task of documenting each area is laborious and daunting, therefore it is prudent to construct a working solution leveraging the compute power and abundance of data available. An automated solution to such a problem would be the mapping and localization of slums to allow the documentation. The next step would be to constantly monitor these areas in a temporal fashion to register any change over the years. Once this data is available and indexed, then these slums can be monitored regularly by the concerned authorities and NGOs to plug any loopholes. Once these basic steps are implemented, rehabilitation and shifting of residents from these areas can be initiated on a massive scale.

In this project we utilize the freely available satellite data to collect and document these slums across various cities of Pakistan. We constantly monitor the changes by learning to localize, map and segment the slums. We develop an automated deep learning based solution to deal with this problem in a remote sensing fashion.

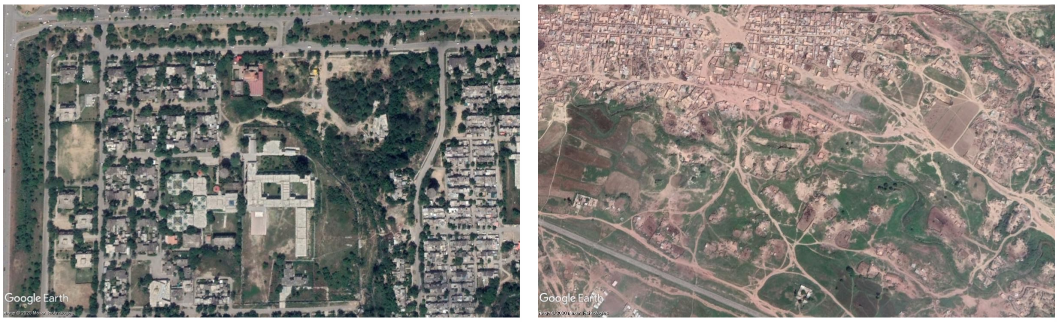


Fig. 1.1: A general overview of two areas in Islamabad. The one on the right contains the slum, whereas the one on the left does not.This figure establishes the stark difference between the two settlements.

Figure 1.1 visually illustrates the striking difference between two different settlements. The figure also visually confirms how the slums are densely populated as opposed to the population in the normal residence area. It is intuitive to follow that due to these dense packings, the living conditions are substandard and there are no proper basic amenities of life. Furthermore, it is also important to note that the area without the slum (the one of the left in the Figure 1.1) is located in a rather modern urbanized location whereas the slum area is remote. Therefore location is also one of the major factors due to which these slums go unnoticed.

1. EFFECT OF GLOBAL URBANIZATION

Unprecedented urbanization has led to an increased population in urbanized locations. According to an estimate by the United Nations, (UN, 2015), approximately two-thirds of mankind lives in urban areas. Trapped in this vicious cycle of poverty, the entire world has observed a spike in rural-urban migration as people migrate to megacities for better hopes of settling. Consequently, informal dwellings arise within these cities.

Slums, conversely known as informal settlements, are regions with substandard living conditions, deprived of basic necessities like availability of clean water and proper sanitation. Moreover, lack of adequate housing facilities and localities often overcrowded by the residents are characterized as features of slum areas. In a report by UN-Habitat, around one-quarter of the Earth's urban inhabitants reside in slum settlements presently.

2. A REMOTE SENSING PERSPECTIVE

It is important to note that slums cannot be eradicated by removing them rather by transforming them into better habitats. (Cobbet, W., 2013) Nonetheless, we only have meager information about the dynamics of such areas, for instance, location, dimensions, boundaries, and population. This is where remote sensing comes into play i.e. RS can be used effectively to build holistic strategies to improve people's living conditions via mapping and monitoring the spatial along with temporal dynamics of slums. (Sustainable Urbanization)

Over the years, the use of high-resolution satellite imagery has become of high interest in the research community in order to effectively understand the morphology of little, heterogeneous structures like buildings and urban environments when fed to several object detection and semantic segmentation algorithms.

In the next section, we look into past work on slum mapping using remotely sensed imagery to understand the intricate nature of slum dwellings.

Chapter 2

LITERATURE REVIEW

SLUMS

Pakistan has witnessed rapid urbanization in recent years, the speed which Pakistan has never witnessed before. According to the figures for the recent census in 2017, the urban population accounts for 36.4%. For comparison, the urban population accounted for 32.5% in 1998. As far as the future is concerned, it has been estimated by the United Nations Population Division that the urban population will account for almost 50% in Pakistan.

The reason for this growth in urbanization is better economic opportunities in the cities. The research and analysis show that the per capita income in the major cities of Pakistan is higher than the overall per capita income of Pakistan. Therefore, the poverty rate in large cities is low. Also, urban centers contribute the most to the GDP and economy in general. (United Nations)

But the uncontrolled population growth in the cities and rapid urbanization can increase the load on the city infrastructure which ultimately leads to poor housing, inadequate water supply. The biggest confrontation all the major cities in Pakistan face due to the lack of management, planning, and absence of urbanization policies is the development of urban slums in the cities. These urban slums stunt the growth of the cities and the people living in those cities. Hence, the same cities which provide better economic opportunities for the people, even fail to provide basic civic services like proper housing, adequate water supply, worse or no sewerage systems. We can see the example of this in every major city by visiting the slums in Karachi, Lahore, and Islamabad. (Wurm, M.; Stark, T.; Zhu, X.X.; Weigand, M.; Taubenböck, H., 2019)

Pakistan is not alone in this space, all the countries especially Asian countries are facing the challenges of rapid urbanization.

REMOTE SENSING

During recent years, the availability of Very High Resolution (VHR) images from satellites has enabled many applications in the field of urban remote sensing. These VHR images from satellites are also increasingly used in the slum mapping context. Also with the advent of new image processing techniques especially deep learning for computer vision has enabled us to analyze these VHR images like never before. We will now discuss the previous work done in remote sensing for slum mapping.

REMOTE SENSING AND DEEP LEARNING

Semantic Segmentation

Given an image, semantic segmentation is semantic labeling every pixel of the image any one class of the domain. For example, in the case of slum mapping, we have two classes i.e. background and foreground (slum).

Why Deep Learning?

Recent advances in the field of deep learning have shown that the deep learning models learn the mid and high-level features efficiently and effectively. Convolutional Neural Networks (CNNs) excel in extracting the image features representation from the raw images. In large scale recognition (Simonyan, K., Zisserman, A., 2014), object detection (Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016), and semantic segmentation (Long, J., Shelhamer, E., Darrell, 2014), CNN's are found to be very effective.

State-of-the-art deep learning models for semantic segmentation are FCN (Long, J., Shelhamer, E., Darrell, 2014), U-Net (O. Ronneberger, P. Fischer, and T. Brox, 2015), Mask-RCNN (He, Gkioxari, Dollar, & Girshick, 2017).

Fully Convolutional Networks for Semantic Segmentation

FCN (Long, J., Shelhamer, E., Darrell, 2014) uses fully convolutional layers for the image semantic segmentation. The problem which arises from the pooling i.e. the field of view is increased but losing the detailed information is solved using the encoder-decoder architecture, which uses the skip connections from the encoder to preserve the features. Instead of the fully-connected layer at the end, which is used for object classification, deconvolution is applied to get the output of the shape of the image.

U-Net: Convolutional Networks for Biomedical Image Segmentation

U-Net (O. Ronneberger, P. Fischer, and T. Brox, 2015) is an improvement over the FCN (Long, J., Shelhamer, E., Darrell, 2014)designed for biomedical semantic segmentation. It utilizes more skip connections to preserve the features from every scale in the encoder and concatenates it to the layers in the decoder. It also solves the problem of vanishing gradients in FCN (Long, J., Shelhamer, E., Darrell, 2014) as the gradients directly flow to the earlier layers in the neural network.

Mask R-CNN

MaskRCNN (He, Gkioxari, Dollar, & Girshick, 2017) is a renowned semantic segmentation architecture that uses a combination of Faster RCNN and FCN. The basic principle on which mask R-CNN works is pretty easy to grasp. Firstly, it uses a faster R-CNN that assigns a bounding box along with a class label for each candidate object. After these assignments, FCN (Long, J., Shelhamer, E., Darrell, 2014) comes into play and adds another branch to the initial outputs i.e. an object mask. Object mask is basically a binary mask which is an indicator of the pixels where the object is detected within the bounding box. This results in the extraction of the much finer spatial information about the detected object.

Transfer Learning for Semantic Segmentation using Convolutional Neural Networks

In deep learning, training is usually done using new datasets on pre-trained networks on larger datasets e.g., ImageNet (Deng et al., 2009), Pascal VOC (Everingham et al., 2010) or COCO (Lin et al., 2014). This phenomenon is called transfer learning where we transfer the already learned weights from one domain to another domain. It is the application of knowledge learned from one dataset to another dataset. Transfer learning has proven to boost the performance in semantic segmentation. Another advantage of transfer learning is the reduced learning time, as the network does not need to be trained from scratch.

In our project, we have used the pre-trained VGG19 (Simonyan and Zisserman, 2014) on ImageNet (Deng et al., 2009). Hence, ImageNet (Deng et al., 2009) is our source dataset. We have a source dataset (ImageNet) different from the target data (slum data). Hence, we have used a type of transfer learning called inductive transfer learning (Pan and Yang (2010)).

Chapter 3

PROBLEM DEFINITION

Slums, characterized as heavily populated urban areas with substandard living conditions, are emerging as a dominant type of settlement in various cities across Pakistan. Exacerbated by in-migration and population growth, this is out of the city government's control. The initial stage for slum eradication requires detection and mapping of slum areas that has been done via inefficient ways with long temporal gaps like consensus surveys in the past. Slum-mapping through aerial imagery using Deep Learning (Convolutional Neural Networks), is the solution that can effectively be used for reliable and timely access to data required for slum eradication. The satellite data from various cities in Pakistan (Islamabad and Karachi) is collected and in turn fed to a model for semantic segmentation to slum and non-slum regions. The objective is to produce a generalized mapping model that can map slums in a new city of Pakistan with few training samples. This approach will help to efficiently monitor slum growth across different areas in Pakistan.

- One of the major challenges is to solve the challenge of an imbalanced dataset. As the slum pixels are present in a very low ratio in the dataset.
- Second most important challenge is the unavailability of large datasets. So we have to utilize the dataset optimally.
- Third important challenge is to transfer the learned knowledge from one city to another city on which we have no dataset available.

Chapter 4

METHODOLOGY

In this section, we present a methodological framework to approach the problem at hand. We go through a series of steps to provide an overview of the methodology employed as shown in figure 1.

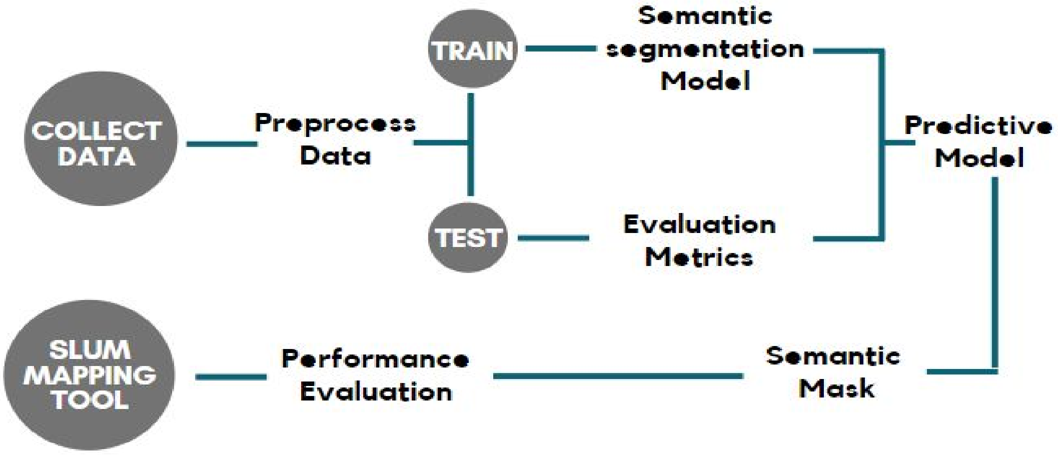


Fig. 4.1: A general overview of various steps to approach the slum segmentation problem. The entire process is split into two main stages: data collection and segmentation pipelines. The slum mapping tool is a web-based platform where the trained networks are deployed.

1. DATA COLLECTION

Introduction:

Data collection is one of the most important contributions of this project. As per the goal and objective of the project, our focus was to map the slums in Pakistan. Karachi, Lahore, Islamabad are the urban centers of Pakistan. As urbanization has increased in Pakistan, these three cities Karachi, Lahore, Islamabad saw the largest number of increases in the slums area.

As far as image data for slums in Pakistan, we were unable to find the data on any city in Pakistan. However, we found the list of names and locations of the slums in Karachi and Islamabad. We divided these cities into zones, curated the list, created the annotations/polygons from the list, then verified the slums by visiting the slum area and talking to the local residents of those areas.

Cities:

Karachi has the largest slum area and population in Pakistan, so this study on slums in Pakistan would have been incomplete without the addition of Karachi. Islamabad, as a newly built and well-planned city, as compared to Karachi and Lahore, is a very interesting case to be studied. As the group members of the project also belong to Karachi and Islamabad, these two cities were finalized for the data collection as the team members are better aware of the ground reality and truth of these cities. Lahore can be considered for future contributions and improvement of the project.

As the German supervisor (Thomas Stark) provided us the data of Mumbai city, we were also able to compare the study and results of cities in Pakistan with Mumbai, a coastal city of India.

Zones:

After the finalization of the cities, the cities were divided into areas and districts. For Islamabad, Zone 1, as the most populated and the well-planned zone, was finalized for the data collection whereas, in Karachi, District Central and District South were finalized.

Listing:

After the zones and districts were finalized, we looked for the data. Although we did not find the polygon or image dataset on the slums, we were able to find the data on the names and location of the data of slums. We curated a list of slums in our region of interest. The sources for the list were newspapers, surveys, past research on slums.

Following is the list of few slums which are on the list:

Table 4.1: The location based distribution and details of slum data collected for various locations in Pakistan (Islamabad, Karachi Central, Karachi South). The table illustrates the location (latitude and longitude) in addition to the area in sq. m snapped by these locations.

Count	Location Name	Latitude	Longitude	Area (sq. m)
1 - Islamabad				
1.1	FranceColony F-7/4	33°43'12.03"N	73°3'44.40"E	40,286
1.2	ChristianColony F-6/2	33°43'49.04"N	73°4'23.10"E	38,793
1.3	ChristianColony G-8/1	33°41'26.48"N	73°3'5.37"E	52,550
2 - Karachi Central (64.9 sq. km)				
2.1	khameso_goth_new_karachi	24°58'1.43"N	67° 2'51.55"E	683,500
2.2	NusratBhuttoColony_1	24°57'40.36"N	67° 2'46.51"E	168,564
2.3	KhamoshColony	24°53'52.80"N	67° 2'18.51"E	42,774
2.4	KausarNiaziColony1	24°55'41.76"N	67° 3'16.40"E	307,285
2.5	UmarFarooqColony	24°56'15.28"N	67° 1'32.38"E	301, 563
2.6	GothGolimar	24°53'54.21"N	67° 1'14.44"E	370,552
2.7	Moosa Colony	24°55'15.68"N	67° 2'51.99"E	243,920
2.8	KausarNiaziColony2	24°55'30.13"N	67° 3'0.28"E	59,319
2.9	PaposhNagar	24°55'36.49"N	67° 1'6.89"E	184,201
2.10	gulberg_2	24°56'47.25"N	67° 4'4.50"E	139,859
3 - Karachi South (89.3 sq. km)				
3.1	ManzoorColony	24°50'49.89"N	25 67° 4'53.83"E	1,017,737
3.2	AzamBasti	24°50'54.74"N	67° 5'53.96"E	1,151,218
3.3	ChanesarGoth	24°51'3.44"N	67° 3'52.59"E	258,365
3.4	BundGali_Kharadar_SaddarTown	24°51'16.44"N	67° 0'1.28"E	260,989

Annotations

After the finalization of the list, the slums were annotated on Google Earth. Polygons were drawn with the high-resolution earth images in the background. Those polygons can be converted to the mask with the help of QGIS. The polygons that were drawn also contained the latitude and longitude information about that slum.

Verification

Collecting the data from just a software raises the question of the validity of the data collected. To remove the ambiguity, we also visited the sites/slums on which the data collected. We visited all the slums in Islamabad to verify them. In Karachi, we verified some of the slums by visiting them. For the rest of the slums, we contacted the local resident nearby to verify the existence of the slum.

Screenshots

Karachi Central

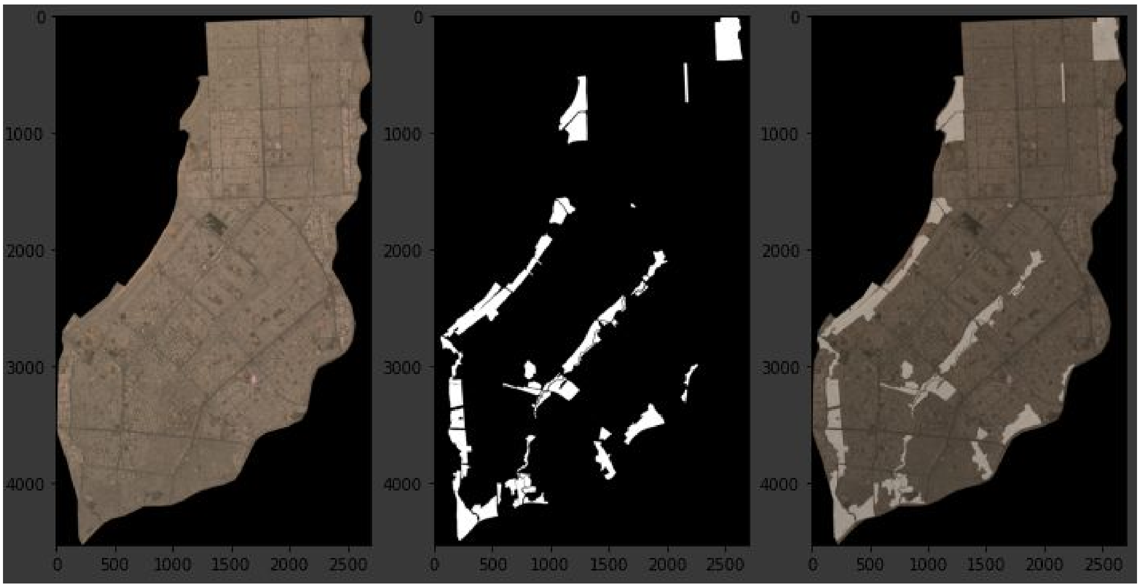


Fig. 4.2: (a-leftmost image) a complete image of the entire Karachi Central satellite imagery (b- center image) depicts the ground truth for all the slums (mask) for the entire Karachi Central while (c-rightmost image) depicts the mask overlaid on the original satellite data.

Karachi South

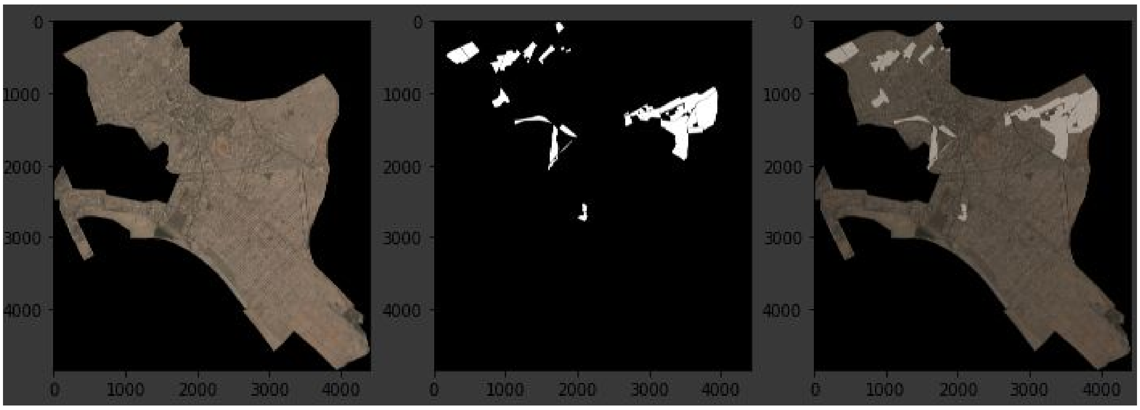


Fig. 4.3: (a-leftmost image) a complete image of the entire Karachi South satellite imagery (b- center image) depicts the ground truth for all the slums (mask) for the entire Karachi South while (c-rightmost image) depicts the mask overlaid on the original satellite data.

Islamabad Zone 1

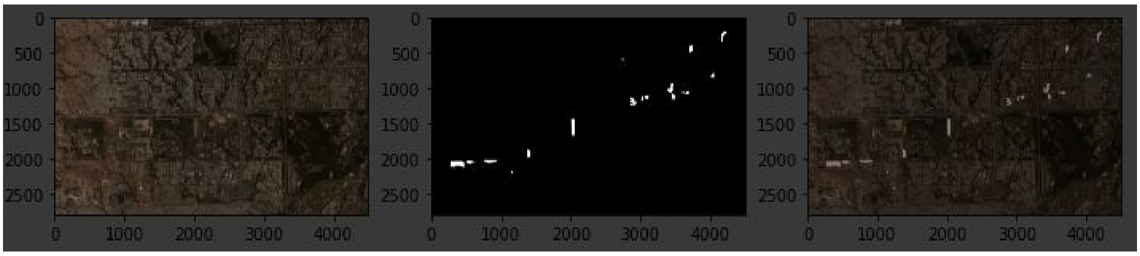


Fig. 4.4: (a-leftmost image) a complete image of the entire islamabad satellite imagery (b- center image) depicts the ground truth for all the slums (mask) for the entire Islamabad while (c-rightmost image) depicts the mask overlaid on the original satellite data.

2. DATA PREPROCESSING

After obtaining data from Google Earth and overlaying it on the satellite data (planet-3m resolution), we perform some preprocessing so that data can be fed to the network. We employ the following techniques.

- Data Tiling → Since the data obtained via satellite has huge dimensions, we perform the tiling operation. This slices the data into 224 x 224 crops for both satellite image and slum masks.
- Data Filtering → The satellite data we received was quite imbalanced, i.e. greater quantity of non-slum area as compared to slum area. In order to mitigate the imbalanced dataset effect we filtered the data thus obtained by removing tiles with no slum area.
- Data Augmentation → Next, we implement rotation for each of the remaining crops to amplify data size.

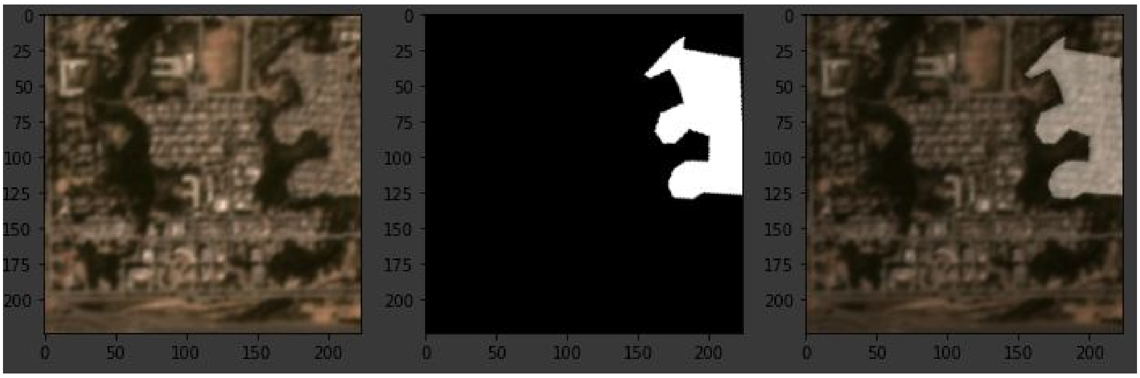


Fig. 4.5: (a-leftmost image) 224 x224 tile demonstrating the slum image data, (b- center image) depicts the ground truth mask while (c-rightmost image) depicts the mask overlaid on the original data tile.

3. DEEP LEARNING PIPELINE

The next step in the process was to build a deep learning architecture. For this purpose, we explored various models, and using these models we trained the model on the pre-prepared training dataset and tested on the test dataset. The details on the model architectures are explained in chapter 5.

4. INFERENCE PIPELINE

Inference from the network was performed using the trained model checkpoint and test accuracy and loss were reported.

5. TOOLS USED

- Google Earth Pro
- QGIS
- Tensorflow
- opencv
- GDAL

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

1. FULLY CONVOLUTIONAL NETWORK (FCN)

One of the state of the art architectures for semantic segmentation is FCN, initially developed by Long et al. (2015) that enables an end-to-end and pixel-to-pixel training for segmentation and drawing robust predictions from random sized input data. The model performs dense forward pass along with backprop computations on the entire scene/input for training and predictions. Inside the entire network subsampled pooling along with upsampling layers enable a pixel-wise inference and learning.

The Visual Geometry Group (VGG) of Oxford University (Simonyan and Zisserman, 2014) created a CNN based classification architecture called the VGG19, we use this model for our experiments. The small receptive fields (3 x 3 pixels) are convolved with the input image at each pixel. Following that, we say that two convolutional layers have an effective receptive field of 5 × 5. Similarly, a stack of four convolutional layers has a 9 × 9 effective receptive field. This approach is useful in a way that it

- Instead of a single rectification layer, it Incorporates four nonlinear rectification layers and as a result the decision function becomes more distinctive.
- Causes a decrease in the number of parameters, hence reducing the space complexity. Since $4(3^2 C^2) = 36C^2$ results in a lesser number of trainable weights as compared to a single 9 × 9 convolutional layer which produce $9^2 C^2 = 81C^2$.

To convert a CNN-VGG19 to an FCN architecture a few alterations are needed. The fully connected layer of the classification network is replaced by a (1 x 1) fully convolutional layer with a number of classes as the channel dimension that is 2 in our case denoted as slum/non-slum. Additionally, transposed convolutional layers are introduced for bilinear upsampling that convert coarse outputs to pixel-dense outputs. This deconv operation is precisely a reverse operation of the forward and backward passes of the convolution. Upsampling is performed for end-to-end learning by backpropagation from a pixel-wise loss (Long et al., 2015). The following figure depicts the FCN-VGG19 architecture.

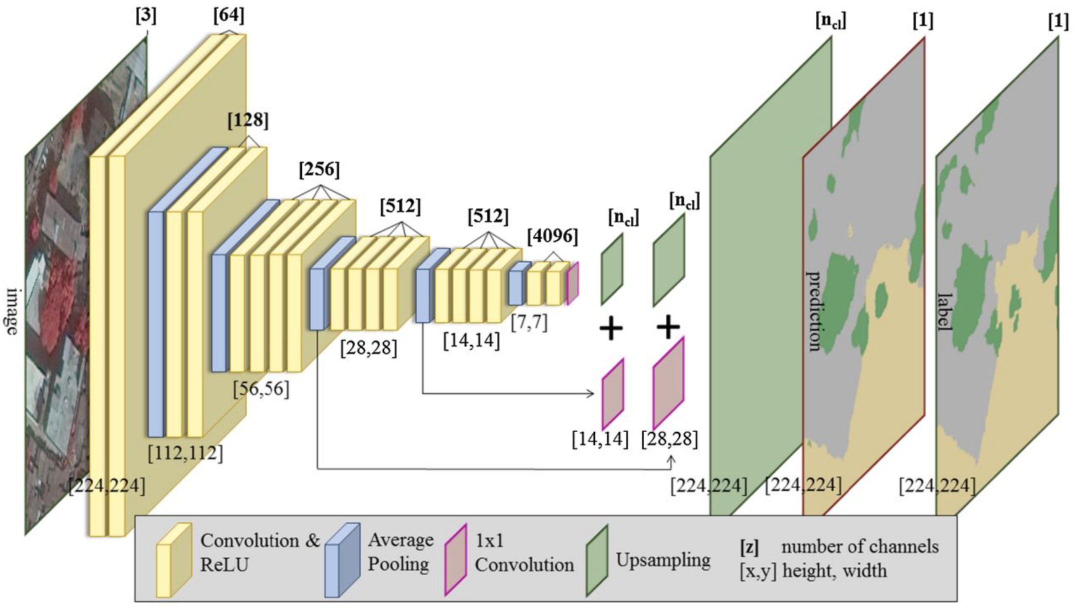


Fig.5.1: Basic network diagram of FCN-VGG19 architecture (Long et al. (2015), the image is taken from the original paper. FCN-VGG19 is one of the networks used throughout this thesis. Each tile represents an operation. Best viewed in color.

2. GAUSSIAN PROCESS REGRESSION NETWORK

In addition to employing standard networks for learning to localize and map slums, we also formulated the approach in a Bayesian setting. It is intuitive to follow that the distribution of slums across a certain region is rather Rayleigh than normal due to the non-uniformity in the density of an area. Slums are dense settlements in confined areas, and therefore due to this, the segmentation becomes an issue. Formulating segmentation in the Bayesian setting allows the utilization of learned priors to enhance the learned embedding for the task of segmentation. A standard Bayesian setting consists of computing a prior, and a likelihood to eventually obtain a posterior. In this task, we formulate the prior as a learnable Gaussian Process (GP) to learn the pixel distribution of the slums.

In the first step, we fit a GP on slum images employing a standard Radial Basis Function (RBF) as a kernel. This is because of the fact that in an RBF, the entire value depends on the distance computation. In this task, our goal is to learn the slum pixel distribution of slums in an area, in other words, how far is a slum pixel from others. Let an area of some m.sq. by m.sq be given by

$$X = \left\{x_1, x_2, \ldots, x_n\right\}$$

where set X are the learned features of that area. In theory, owing to the density of slum and sparsity of the adjoining areas, the pixel distribution is Rayleigh i.e.

$$f\,(x, \sigma) = x/\sigma^2 e^{(-x^2)/(2\sigma^2))} \quad x > 0$$

We can remodel this distribution by employing a learn-able GP with an RBF kernel. The RBF kernel can be written as:

$$k(x, x') = exp(-1/2\|x - x'\|_2^2)$$

A GP can be defined by a kernel $k(x, x')$ and a mean?. Once a GP fit has been obtained, in each pass over a single batch of the slum, we modify the extracted embedding from the baseline neural network by applying a linear transformation on two embeddings, one from the neural network, one from the GP. Each batch is also passed to the GP to obtain the learned feature maps. A GP in this case can be written as

$$f \sim GP(\sigma, k(x, x'))$$

Let our backbone (likelihood) neural network be $f'(x)$. We can extract embedding from the trained NN and from GP followed by a linear transformation, we use summation in this case. We can rewrite it as

$$f'' = GP(\sigma, k(x_i, x_i') + f'(x_i))$$

where x_i is for a single image in the batch.

Since a GP fitted on slum images will have learned the distribution of slum, when a linear summation transformation is applied, the areas corresponding to slum get higher weights and are enhanced. This all happens before the network's backward pass and penalization. Since it is important that the neural network adjusts its weights in accordance with the enhanced slum areas to provide effective slum segmentation. We discuss the results in upcoming chapters.

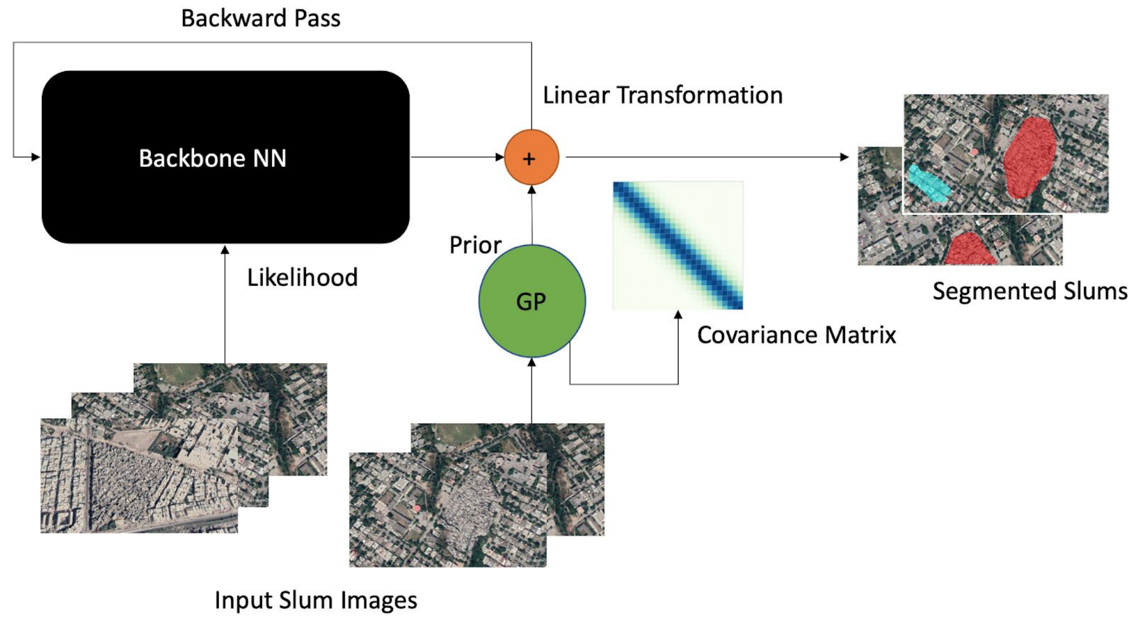


Fig.5.2: GP-FCN architecture design. The backbone NN can be any standard baseline neural network used as a likelihood. The green circle represents a GP trained on the slum images with RBF kernel (the covariance matrix is of the RBF kernel). A linear operation is shown as the orange circle and the posterior maps are generated eventually.

3. UNet

U-Net is a semantic segmentation model which is popular in the biomedical image segmentation. Like FCN, we have used the VGG19 as an encoder for the U-Net. The input shape of the model is 224x224x3 and the output shape is 224x224x2.

We have used a modified U-Net with a total of four skip connections. A U-Net is an encoder-decoder network architecture, where encoder acts as a downsampler and decoder acts as an upsampler. As explained in the literature review, using the pre-trained network can help to reduce the learning time and helps to learn the more robust features, and reduce the number of trainable parameters, we have used VGG19 as our encoder network. We will use the intermediate outputs from the pre-trained network as skip layers.

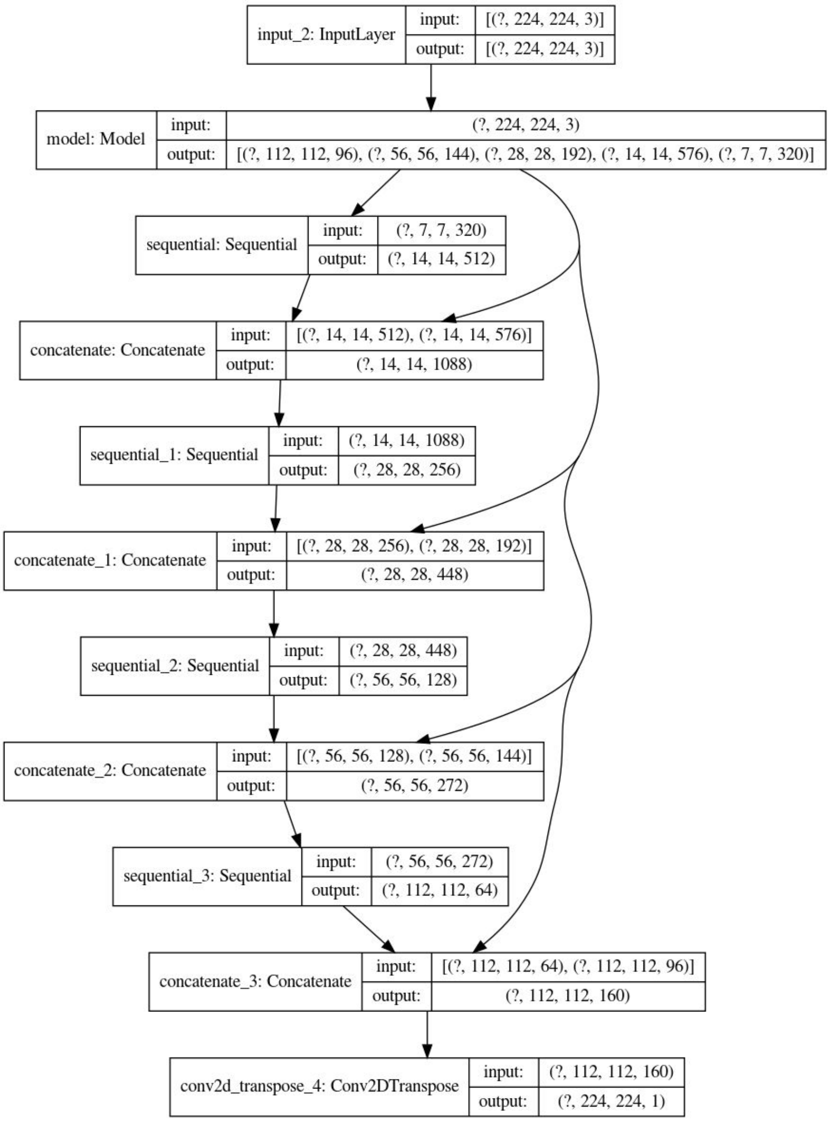

input_2: InputLayer
input: [(?, 224, 224, 3)]
output: [(?, 224, 224, 3)]
model: Model
input: (?, 224, 224, 3)
output: [(?, 112, 112, 96), (?, 56, 56, 144), (?, 28, 28, 192), (?, 14, 14, 576), (?, 7, 7, 320)]
sequential: Sequential
input: (?, 7, 7, 320)
output: (?, 14, 14, 512)
concatenate: Concatenate
input: [(?, 14, 14, 512), (?, 14, 14, 576)]
output: (?, 14, 14, 1088)
sequential_1: Sequential
input: (?, 14, 14, 1088)
output: (?, 28, 28, 256)
concatenate_1: Concatenate
input: [(?, 28, 28, 256), (?, 28, 28, 192)]
output: (?, 28, 28, 448)
sequential_2: Sequential
input: (?, 28, 28, 448)
output: (?, 56, 56, 128)
concatenate_2: Concatenate
input: [(?, 56, 56, 128), (?, 56, 56, 144)]
output: (?, 56, 56, 272)
sequential_3: Sequential
input: (?, 56, 56, 272)
output: (?, 112, 112, 64)
concatenate_3: Concatenate
input: [(?, 112, 112, 64), (?, 112, 112, 96)]
output: (?, 112, 112, 160)
conv2d_transpose_4: Conv2DTranspose
input: (?, 112, 112, 160)
output: (?, 224, 224, 1)

Fig. 5.3: UNet architecture design generated by the network. The architecture diagram represents the layers and the shapes of input at each layer. The arrows represent how the input progresses through the network in a single forward pass, the skip connections are represented as well.

4. MaskRCNN

MaskRCNN (He, Gkioxari, Dollar, & Girshick, 2017) is a renowned semantic segmentation architecture that uses a combination of Faster RCNN and FCN. The basic principle on which mask R-CNN works is pretty easy to grasp. Firstly, it uses a faster R-CNN that assigns a bounding box along with a class label for each candidate object. After these assignments, FCN comes into play and adds another branch to the initial outputs i.e. an object mask. Object mask is basically a binary mask which is an indicator of the pixels where the object is detected within the bounding box. This results in the extraction of the much finer spatial information about the detected object.

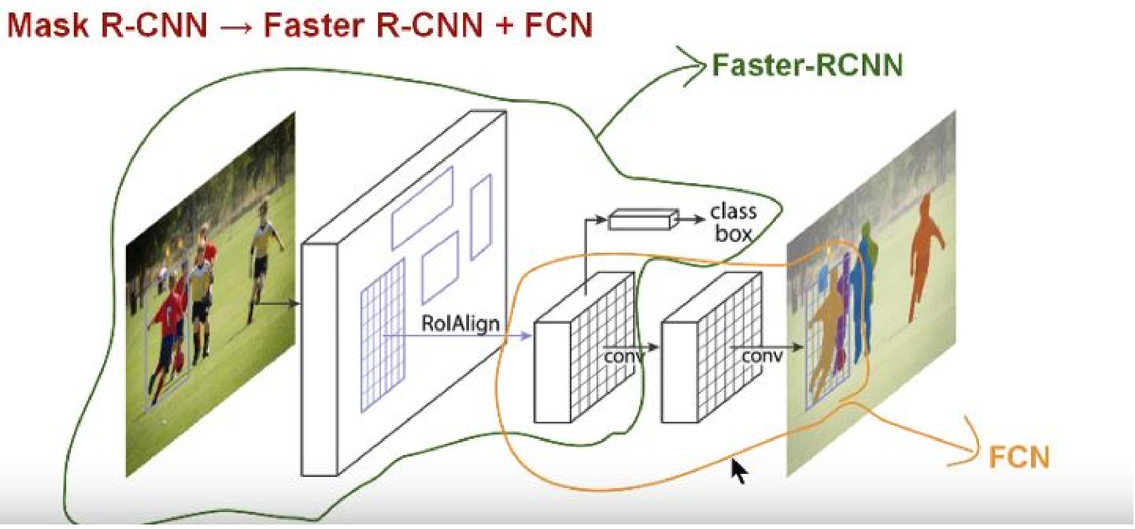


Fig. 5.4: A standard Mask R-CNN architecture design. The Faster-RCNN and FCN blocks are shown in the diagram. The output from Faster-RCNN is fed to a few FCN layers. The image is taken from (Towards Data Science). Best viewed in color.

Chapter 6

IMPLEMENTATION AND TESTING

All of the models mentioned in the previous sections were separately trained and tested to draw out results. The implementation and testing details of each architecture are given below.

MASK R-CNN

For our problem, we trained a Mask R-CNN on the Mumbai dataset and then tested it on the images of Islamabad and Karachi obtained through Google Earth images of 200m resolution. For the implementation we employed an open-source Mask R-CNN (W.Abdulla, 2017) and fine-tuned the pre-trained model with hyperparameters of the network being, a number of epochs = 128, batch size = 2. Along with that, for loss optimization, we used the Adam optimizer that had a starting learning rate = 10exp−4 with decaying factor of 10 after 50 and 120 epochs.

U-Net

We have cross-entropy loss for the U-Net. Adam optimizer is used with the learning rate of 0.001. Bve used the binaratch size used for training U-Net is 2. Epochs used for training the network are 50.

FCN

We have used the binary cross-entropy loss for the U-Net. Adam optimizer is used with the learning rate of 0.001. Batch size used for training U-Net is 2. Epochs used for training the network are 50.

METHOD

As explained in the literature review, using the pre-trained network as an encoder network can help to reduce the learning time and helps to learn the more robust features, and reduce the number of trainable parameters. Therefore, we have used VGG19 as our encoder network. We will use the intermediate outputs from the pre-trained network as skip layers.

The encoder decoder architecture like FCN and U-Net is called hourglass architecture as it is like the shape of hourglass.

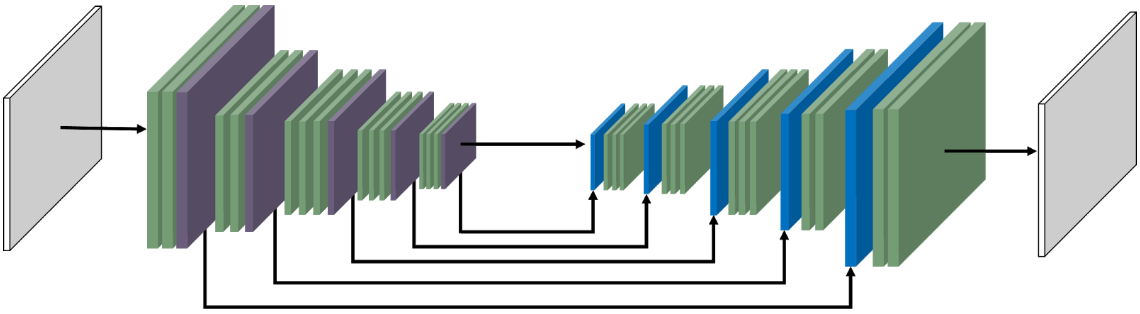


Fig.6.1: Hourglass architecture

LOSS FUNCTION

We have used sparse softmax cross entropy loss function within TensorFlow™ for training the FCN to measure the performance of the model. This loss is a sum of the errors made for each example during the training stage, which shows how good or bad a certain model performs after each epoch of optimization.

ACCURACY MEASURE

Intersection over Union (IoU) or Jacquard Index is the most popular accuracy metric for image segmentation. Since this metric is a community standard in terms of evaluation of segmentation networks, therefore we employ IoU to evaluate the trained networks. The IoU is intuitive in the sense that its name is self explanatory. The IoU can be defined as follows.

$$\text{IoU} = \frac{x_1 \cap x_2}{x_1 \cup x_2}$$

Chapter 7

RESULTS AND DISCUSSION

Now we train the dataset on different model and report results of evaluation matrices

Mask R-CNN

Table 7.1: The evaluation metrics results on training (Mumbai) and testing (Islamabad, Karachi Central, Karachi South) data using Mask R-CNN architecture.

Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
Mumbai	92.6	96.3
Islamabad	86.4	88.7
Karachi South	75.2	87.1
Karachi Central	60.8	84.4

U-Net

Table 7.2: The evaluation metrics results using various locations (Mumbai, Karachi Central, Karachi South) as training data trained on UNet architecture.

Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
Mumbai	94.5	98.8
Karachi South	86.57	92.87
Karachi Central	82.36	90.76

FCN

Table 7.3: The evaluation metrics results on training (Mumbai) and testing (Islamabad, Karachi Central, Karachi South) data using FCN architecture.

Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
Mumbai	67.47	97.22
Islamabad	48.36	94.05

Karachi South	41.29	85.26
Karachi Central	36.25	81.67

It is visible from the table above that the results produced by U-Net trained on Karachi Central are the best so this model is saved and used for the next step.

GP-FCN

We trained GP and FCN separately and only used GP in inference mode when training FCN. Initially we trained a GP to fit the slum images and only used it in inference mode when training FCN. Our GP-FCN's forward pass is a combination of a pass of FCN, followed by a linear combination with GP's output before being subjected to the backward pass.

Table 7.4: The evaluation metrics results on training (Mumbai) and testing (Islamabad, Karachi Central, Karachi South) data using GP-FCN architecture.

Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
Mumbai	68.25	98.43
Islamabad	50.01	95.66
Karachi South	42.73	86.06
Karachi Central	36.89	82.01

1. COMPARATIVE RESULTS

Now that all the architectures have been trained and tested, we compare their results on evaluation metrics.

Table shows that FCN and GP-FCN show comparative results with GP-FCN giving slightly improved results.

Table 7.5: The evaluation metrics results on and testing (Islamabad, Karachi Central, Karachi South) data comparing results of FCN and GP-FCN architectures.

Model	Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
FCN	Islamabad	48.36	94.05
	Karachi South	41.29	85.26
	Karachi Central	36.25	81.67
GP-FCN	Islamabad	50.01	95.66
	Karachi South	42.73	86.06
	Karachi Central	36.89	82.01

Table shows the fluctuation in test results as the model is trained on data from different locations.

Table 7.6: The evaluation metrics results on and testing data using data from different locations comparing results of UNet architecture based on training datasets.

Model	Area	Intersection over union (IoU%)	Overall Accuracy (OA%)
UNet (Trained on Mumbai)	Islamabad	86.79	94.33
	Karachi South	84.53	92.25
	Karachi Central	78.15	89.21
UNet (Trained on Karachi Central)	Islamabad	88.69	94.05
	Karachi South	86.62	92.89
	Mumbai	74.96	85.72
UNet (Trained on Karachi South)	Mumbai	74.88	85.72
	Islamabad	88.54	94.05
	Karachi Central	81.88	90.37

2. OUTPUT

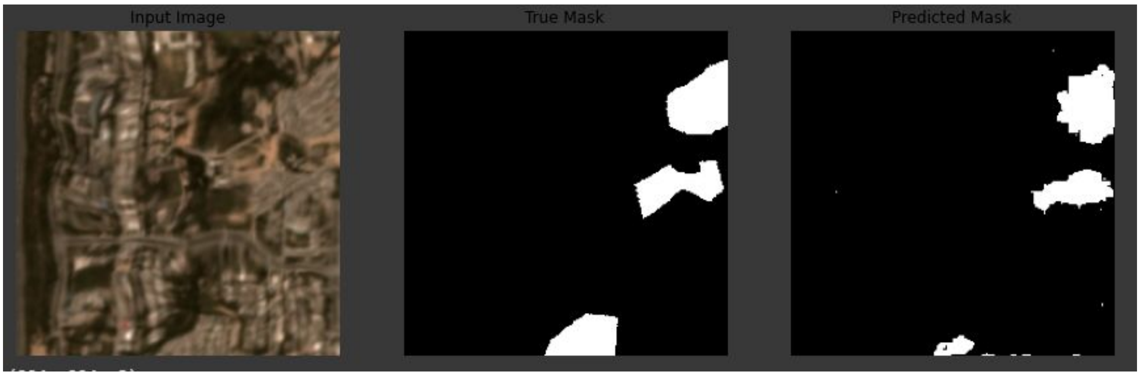
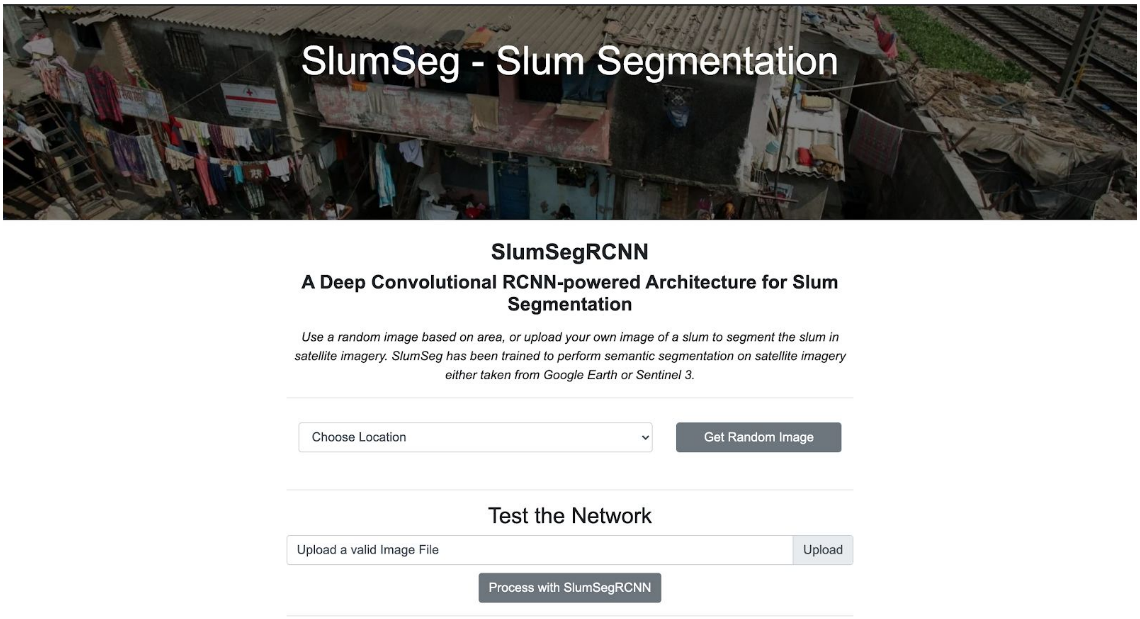


Fig. 7.1: FCN-VGG19 output with (a-leftmost image) demonstrating the slum image data, (b- center image) depicts the ground truth mask while (c-rightmost image) depicts the predicted mask

3. WEB DEMO

We show our findings and results with the help of a website. This web demo uses a Mask R-CNN deployed on a server and displays the inference done by the model on the test data. Following are the salient features of the website.



SlumSeg - Slum Segmentation

SlumSegRCNN

A Deep Convolutional RCNN-powered Architecture for Slum Segmentation

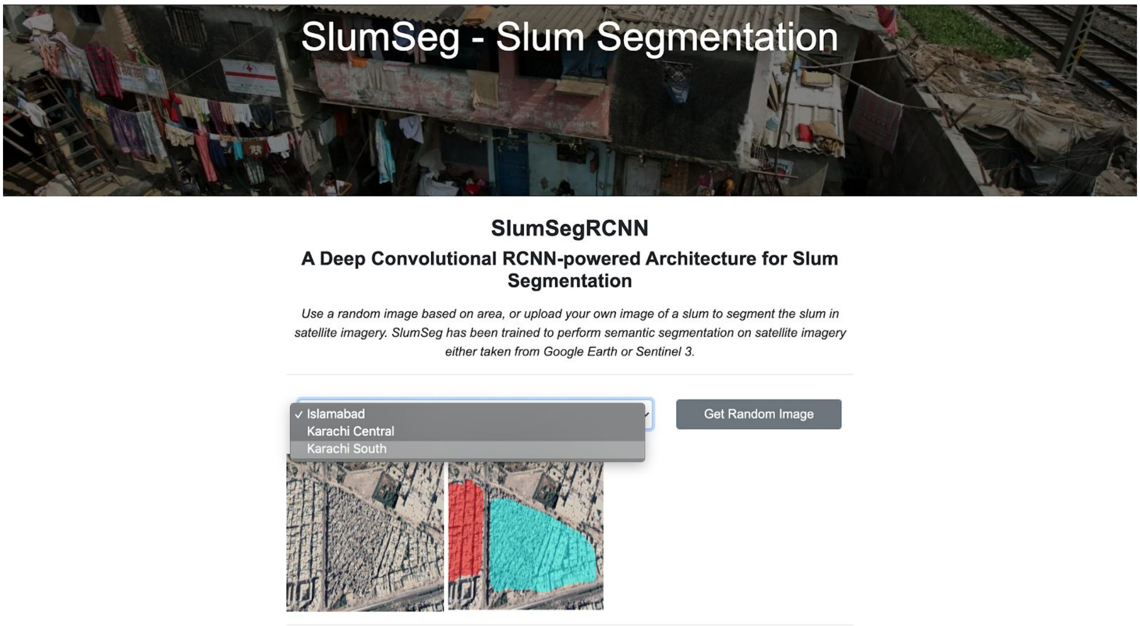
Use a random image based on area, or upload your own image of a slum to segment the slum in satellite imagery. SlumSeg has been trained to perform semantic segmentation on satellite imagery either taken from Google Earth or Sentinel 3.

Test the Network

Fig.7.2: Website - at a glance. The initial web interface available to the user once they visit the website.

- Random Image Selector

  You can choose the model results based on locations namely, Islamabad, Karachi Central, Karachi South. This will choose a random image from the chosen location and display the pre-saved model prediction on that image.



SlumSeg - Slum Segmentation

SlumSegRCNN

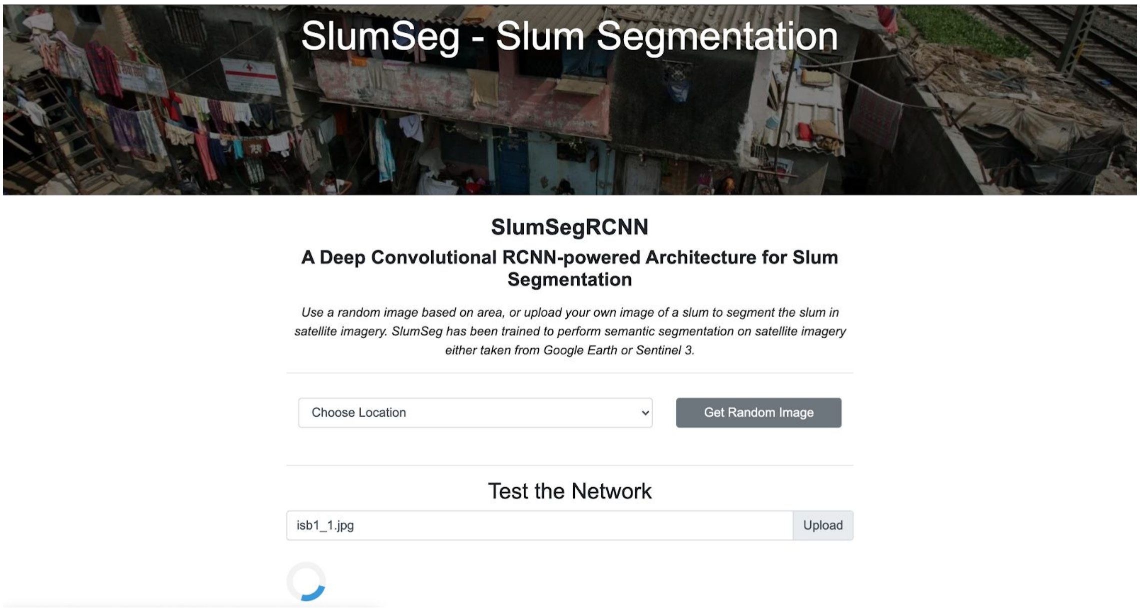
A Deep Convolutional RCNN-powered Architecture for Slum Segmentation

Use a random image based on area, or upload your own image of a slum to segment the slum in satellite imagery. SlumSeg has been trained to perform semantic segmentation on satellite imagery either taken from Google Earth or Sentinel 3.

Fig. 7.3: The random image results generated based on selected location. This feature lets the user select a location from the dropdown list. A random image and the segmented results are displayed from that area.

- Image Chooser

  The website also allows you to pick your own image in which a slum needs to be mapped. An image chooser utility will help you select an image from your device and runs a live model inference on the image selected.



SlumSeg - Slum Segmentation

SlumSegRCNN

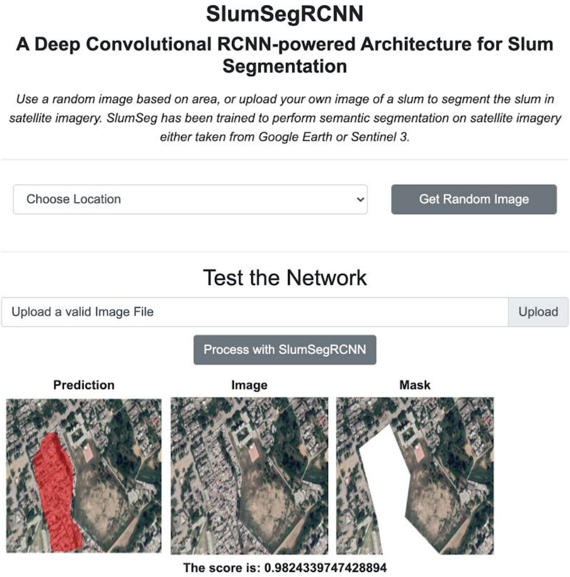
A Deep Convolutional RCNN-powered Architecture for Slum Segmentation

Use a random image based on area, or upload your own image of a slum to segment the slum in satellite imagery. SlumSeg has been trained to perform semantic segmentation on satellite imagery either taken from Google Earth or Sentinel 3.

Test the Network

isb1_1.jpg

Fig7.4:. Upload your own image for live network testing. This allows the user to upload a satellite image of a slum to get the prediction from the network. The results along with the image, the ground truth and the predicted mask are displayed.



SlumSegRCNN

A Deep Convolutional RCNN-powered Architecture for Slum Segmentation

Use a random image based on area, or upload your own image of a slum to segment the slum in satellite imagery. SlumSeg has been trained to perform semantic segmentation on satellite imagery either taken from Google Earth or Sentinel 3.

Test the Network

Upload a valid Image File

The score is: 0.9824339747428894

Fig.7.5: The result generated by the underlying model showing network prediction as well as ground truth. These results are displayed when the user uploads a slum image to be fed to the network running in the inference mode.

You can access the website here.

Chapter 8

CONCLUSION AND FUTURE WORK

There are numerous slums out of the rehabilitation plan of the government and NGOs largely because they are unknown. This leads to the constant deterioration of those areas where there are already no proper facilities to sustain life. With slum mapping, localization and segmentation, we can help bring those areas on the map and drive the attention of concerned authorities towards it. In this project, we experimented with satellite data from various sources, namely planet, sentinel, and Google Earth employing various architectures to report the performance of different deep learning architectures on slum segmentation problems in various cities of Pakistan. To make sure there was a document dataset available of various slum settlements across Pakistan, we collected an extensive dataset and performed slum segmentation and analysis.

In the end we can conclude following points based on the experimentation and the literature review:

1. Transfer learning is very important to learn the representation of one city using pretrained networks. Pretrained networks converge much more quickly, and are also more robust than training a network from scratch.
2. Transfer learning is also very useful to transfer the learned knowledge of one city to another city. We can say that at least in Pakistan, we can use this technique to map the slums of other cities using the learned knowledge of one city.
3. If we can collect the data on the zone or district of a city, train the model on it and try to map the slums in other parts of the city, we can get better results than directly using the knowledge from other cities.
4. U-Net performs better for our task as it uses more skip connections, therefore utilizes the low level knowledge efficiently.

5. FCN does not work well for our task due to the vanishing gradient problem as our problem dataset is very imbalanced.

Based on our current knowledge gained from the experiments, following things can be done to improve over our work:

1. Use the multi-resolution data cube which comprises data from different satellites. This will allow models to learn more robust features of the slums. Especially, if we combine the low resolution and high resolution images.
2. Estimating the poverty rate from different slum settlements by learning population density from satellite imagery. The eventual research direction would be to develop an end-to-end pipeline to not only segment the slum in an area but to also estimate the population density and eventually poverty rate of a city.

Chapter 9

BIBLIOGRAPHY

(https://www.pk.undp.org/content/pakistan/en/home/library/development_policy/dap-vol5-iss4-sustainable-urbanization.html)

#Envision2030 Goal 1: No Poverty | United Nations Enable. (2010). Retrieved from Un.org website: https://www.un.org/development/desa/disabilities/envision2030-goal1.html

Cobbett, W. How cities can get rid of slums by supporting them. Informal City Dialogues, 22 April 2013.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. IEEE Computer Vision and Pattern Recognition (CVPR).

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., and Pal, C. (2016). The importance of skip connections in biomedical image segmentation. In Deep Learning and Data Labeling for Medical Applications (Springer). 179–187

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016. Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 38 (1), 142–158.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2017.322

He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 ´ IEEE International Conference on. pp. 2980–2988. IEEE (2017)

Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2014), arXiv:1411.4038 [cs.CV]

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

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. In: Proc. International Conference on Learning Representations.

UN-Habitat. Informal Settlements; UN-Habitat: New York, NY, USA, 2015; pp. 1–8.

W. Abdulla, “Mask r-cnn for object detection and instance segmentation on keras and tensorflow.” https://github.com/matterport/Mask_RCNN, 2017.

Wurm, M.; Stark, T.; Zhu, X.X.; Weigand, M.; Taubenböck, H. Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. ISPRS J. Photogramm. Remote Sens. 2019, 150, 59–69