Tribhuvan University
Institute of Engineering
Pulchowk Campus

A
Final Defense Report
On

Rapid Earthquake Assessment from Satellite Imagery using RPN and Yolo v3

THESIS NUMBER: 073/MSCS/665

Submitted by:
Saurav Lal Karn

Submitted to:
Department of Electronics and Computer Engineering
November 2019
Tribhuvan University

Institute of Engineering
Pulchowk Campus

A
Final Defense Report
On

Rapid Earthquake Assessment from Satellite Imagery using RPN and Yolo v3

THESIS NUMBER: 073/MSCS/665

Submitted by:
Saurav Lal Karn

Submitted to:
Department of Electronics and Computer Engineering
November 2019

Acknowledgement

I am deeply grateful to my advisor, Associate Professor Dr. Sanjeeb Prasad Pandey, without whom I would not have been able to work on the implementation. Despite his busy schedule, Dr. Aman Shakya was very patient in answering all of my questions and offering invaluable assistance. His innovative thinking, determination and critical reviews helped me to assess various interesting domains.

I would also like to thank Dr. Shakya for helping to conceive the roadmap for our work, also Associate Professor Dr. Dibakar Raj Pant for guiding on the tools and techniques necessary for problem solving.

I would like to thank Dr. Surendra Shrestha for providing us with the opportunity to work on this implementation

Abstract

Nepal being in highly earthquake prone region suffers from earthquakes frequently. The relief that is to be sent to the affected area requires rapid earliest assessment of the impact in the area. The number of damaged buildings provides us with the necessary information and can be used to assess the impact. Disaster damage assessment is one of the most important parts in providing information about the impact to the affected areas after the disaster. Rapid Earthquake damage assessment can be done via the satellite imagery of the affected areas. This work implements the Region Proposal Network(RPN) and You only look once (Yolo) v3 for generating region proposals and detection. Sliding window approach has been implemented for the method to work on large satellite imagery. The obtained detections are compared with the ground truth

Table of contents

List of abbreviations

1.	AWS	:	Amazon Web Service
2.	CNN	:	Convolutional Neural Network
3.	CPU	:	Central Processing Unit
4.	Fast R-CNN	:	Fast Region Convolutional Neural Network
5.	Faster R-CNN	:	Faster Region Convolutional Neural Network
6.	GB	:	GigaByte
7.	Ghz	:	GigaHertz
8.	GIS	:	Geographic Information System
9.	IEEE	:	Institute of Electrical and Electronics Engineers
10.	ISODATA	:	Iterative Self-Organizing Data Analysis Techniques
11.	KB	:	KiloByte
12.	MB	:	MegaByte
13.	MSCOCO	:	Microsoft Common Objects in Context
14.	PC	:	Personal Computer
15.	R-CNN	:	Region Convolutional Neural Network
16.	RPN	:	Region Proposal Network
17.	ROI	:	Region Of Interests
18.	SAR	:	Synthetic Aperture Radar
19.	SSD	:	Single Shot Multibox Detector
20.	SW	:	Sliding window
21.	VHR	:	Very High Resolution
22.	VOC	:	Visual Object Classes
23.	Yolo	:	You Only Look Once

List of figures

1.Introduction

1.1 Background

In context of Nepal, Earthquakes are more frequent due to its fragile geography and location between the Eurasian and Indian tectonic plates[1]. In such conditions, planning the relief works is a huge task for the government. Planning relief works require information about the damage caused by disaster. Multiple factors are responsible for determining the damage caused by the disaster. Some of these factors are damage to structures, land mass changes, number of deaths and impact on people's life. Over the past few decades, researchers have tried multiple ways to assess the damage from the disaster. In the above given factors, number of deaths and impact on people's lives cannot be determined via remote sensing. So the researchers have focused on structural damage and land mass changes. The structural damage assessment is normally done via the personnel present at affected areas. This results in loss of valuable time and due to this loss, damage assessment cannot be done in time thus hampering the planning of relief works. To avoid this loss, Remote sensing is used to monitor areas through satellites. In past times, Satellite imagery were of poor quality however this has been solved in modern times due to the introduction of multiple free and powerful satellites which provide high resolution imagery of the areas.

Remote sensing can be defined as the acquisition of information about an area without being in physical contact with the area[2]. Remote sensing is being used in multiple areas like earth geography, survey and earth related sciences. The application of remote sensing are endless (Military, Disaster response etc). Remote sensing generally utilizes data acquired from satellites to analyze and work on the respective field. One of the key fields of Remote sensing is the Disaster and Humanitarian efforts. In case of disaster, Remote sensing can be called as time saver field as it reduces the loss of valuable time. Over time multiple approaches have been applied for the remote sensing techniques. One of these techniques is the object detection via

imagery captured from high resolution camera onboard the satellite. Multiple methods and ways have been introduced to detect specific objects[6].Some of these are Edge Detection Techniques[3], Probability model[4], Segmentation[5], Isodata[6] and etc.

In this work, Detection approach is considered. Detection algorithms can be roughly divided into two categories i.e One stage detectors and Two stage detectors[7]. One stage detectors apply algorithm runs directly over the possible locations while two stage detectors generates proposal regions over an image and classifies those regions. Both of these approaches have their own benefits and setbacks. Like one stage detectors have short processing time but lacks on accuracy while the two stage detectors have higher accuracy but have large processing time. In this work, benefits of both detectors has been integrated. Yolo, Yolo V2, Yolo V3, SSD are one stage detectors while R-CNN, Fast R-CNN, Faster R-CNN falls in the category of two stage detectors [7].

This work deals with assessing the damage caused by the earthquake by estimating the number of damaged buildings just after the disaster. This work incorporates the use of satellite images of affected areas to estimate the number of damaged buildings. This work incorporates the backbone of Two stage detectors (Faster R-CNN) i.e Region Proposal Network along with the (One stage detectors)Yolo V3 classifier due to its great classification accuracy. This work also incorporates sliding window[8] approach to process on large satellite imagery so that processing can be done on small computing machine. The segmentation approach implemented in this work is highly generalizable and can be used to detect other objects than “Building” by training with dataset with required labels.

1.2 Problem Statement

Detection in satellite imagery is quite troublesome task as the images are of very high resolution thus requiring more processing power over the normal images. Combination of the best features of both one stage and two stage detectors is equally challenging task.

1.3 Objectives

1. To determine the number of damaged buildings in the affected areas using satellite imagery.
2. To provide rapid initial assessment about the affected areas.

1.4 Scope of the work

The scope of this work is to identify damaged buildings and provide an assessment of damage. This work can be useful for humanitarian work. With adding more class and proper training, this method can be used to detect other objects as well in satellite imagery.

2.Literature Review

Assessing the damage from the disaster is a very important task in case of aid distribution and disaster management area. While being import, assessing the damage is a very hard task. In past times, the assessment was done via the personnel present at the disaster stuck site. Assessing the damage of disaster depends on multiple factors. Associate Professor,lan Noy of Victoria Business School gave an idea about assessing the damage caused by the earthquake [9]. They tried to assess the damage by converting all the damage indicators, including mortality, morbidity, and other impacts on human lives (e.g. displacement) – as well as damage to infrastructure and housing – into an aggregate measure of human life years lost. In this approach, total years lost was calculated as the sum of years lost due to death, injury/affected, and financial damage.F. Yamazaki from Earthquake Disaster Mitigation Research Center, NIED, Hyogo, Japan based in Institute of Industrial Science, University of Tokyo, Tokyo, Japan applied the use of remote sensing as well as GIS for damage assessment[10] in case of damage caused by the flood by using the SAR images obtained from the IKONOS satellite of the affected areas by determining the liquefied areas.

Another interesting research was done by F. Dell'Acqua and P. Gamba on the tools and techniques for the damage assessment[11]. They also researched on multiple algorithms for data interpretation and information extraction. They used the earth observation data for the analysis and analyzed the multiple methods for assessing the damage caused by the earthquake. In the same regards one of the early research papers published in IEEE was Building detection from high-resolution satellite image by Wei Liu and V. Prinet [12] which dealt with the building detection using probability model. The authors worked on probabilistic model to extract buildings images from satellite imagery. However, this project cannot be generalized i.e. model created for one scene cannot be used for another scene. Another interesting approach was implemented by Amy Zhang, Xianming Liu, Andreas Gros and Tobias Tiecke was Building Detection from Satellite Images on a Global Scale[13]. Their approach used the image

segmentation and classification model. They employed the use of pixel wise segmentation to detect buildings..

In the same manner, another work on damage assessment was done by L. Chiroiu and G. André [14]. The authors employed the use of high resolution satellite imagery for the assessment of 2001, bhuj, India earthquake by using the optical and radar satellite imagery of the affected area as basis for assessment. Likewise, M.R.Archana, Jenis, Shiny George, S.A.Abiram gave an interesting approach for Earthquake assessment using Pre-event and Post-event imagery.by focusing on similarity between both images by assuming that the buildings must have rectangular footprint and isolated. Another interesting work was done by Chandan Dinesh Parape, Masayuki Tamura of Kyoto University Japan [18] by employing the usage of the morphological operators for segmenting images while ISODATA for the feature extraction and classification. This method used the pre disaster and post-disaster images for processing and comparing with the ground truth. One of the major flaws of this work was this method cannot be generalized to segment images as the reflectance changes from image to image.

In the same field, Facebook AI Research group[33] has been doing some advanced research in recent times. They proposed a new approach to segment the buildings by utilizing the two stage procedure with Region Proposal Network in the first stage and Predicting the class and box offset. This method works by extracting proposed multiple regions of interest and classification is done in a second step. This method has considerable accuracy on smaller images but fails to predict considerably in larger images. YOLO(You only look once) [30] is one stage detector algorithm.This approach uses single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. Yolo v3 is usually suitable for the object detection on videos due to its fast processing. This method detects well in small image but fails to detect in larger images

3.Methodology

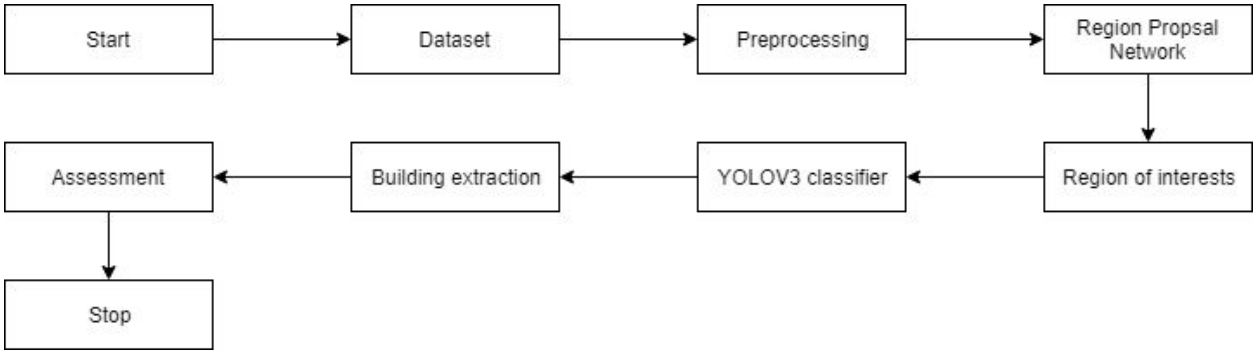

Figure 3.1 : Block Diagram of methodology

3.1 Dataset

In this implemented work, dataset works as a key factor. With correct dataset, the method can be utilized to its utmost potential. 2 standard datasets and 1 custom dataset (Nepal) has been acquired for training the neural nets. The datasets has been described below:

3.1.1 Crowd AI mapping challenge dataset

Crowd AI mapping challenge dataset[23] has been acquired to train Region Proposal Network. This dataset is fairly well balanced and has been used by multiple groups for the mapping challenge. The dataset consists of 280742 images along with the annotations for all of them. Over all the dataset in compressed format has a size of about 3.31 GB. A sample of dataset is shown in Figure 3.2.

Figure 3.2(a)

Figure 3.2(b)

Figure 3.2(c)

Figure 3.2(d)

Figure 3.2 Crowd AI mapping challenge dataset sample

The annotations for the crowd ai mapping challenge is in the MS COCO format. COCO is the abbreviation for the Common Objects in Context. This dataset format is in JavaScript Object Notation format. MS COCO format consists of mainly following sections

3.1.1.1 Info

Info section of COCO format contains the information about the dataset. This field consists of description of the dataset, url of the dataset if online available or empty, version of annotation, year it was created, contributor name and created date. This section provides an overall summary of the annotations.

3.1.1.2 Images

Images section of COCO format contains an array of objects containing the id of the image, width, height of image, filename, license and the captured date of the image. This section provides the path to the annotated image and is used to access the images for various purposes.

3.1.1.3 Annotations

Annotations section of COCO format contains an array of objects containing the id of annotation, linked image id, category id i.e. Category id of the class to be trained, segmentation of the objects in the annotated image, area of the annotated section, bbox i.e bounding box of the annotated object and is crowd flag to show large groups of objects.

3.1.1.4 Licenses

License section of COCO format consists of array of object containing the details about the license under which the dataset is made available.

3.1.1.5 Categories

Categories section contains the array of category under which the model is to be trained. The objects consists of id of the category to correctly identify the object while loading annotations, name of the category and super category if the category falls under other category.
A sample of dataset format is shown in Figure 3.3.

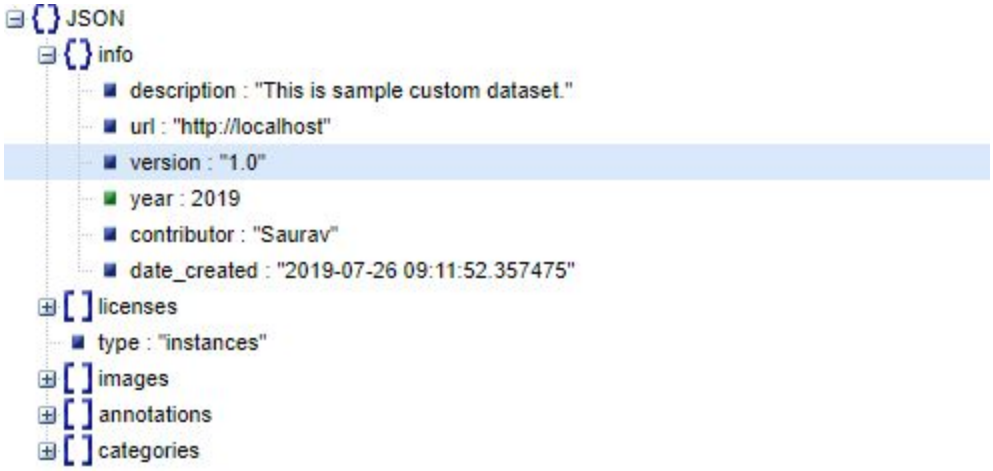

Figure 3.3 Sample of MS COCO format

3.1.2 AWS Spacenet challenge dataset

Another dataset used in this work is the Amazon Web Services Spacenet Challenge [24]. The Spacenet dataset consists of satellite images captured from 27 WorldView 2 Satellite images. The dataset on its complete set consists of images of 6 areas(Rio, Vegas, Paris, Shanghai, Khartoum, Atlanta). In this work, only the imagery of Rio has been used due to the restrictions of the processing power. 3.4 GB out of all the data provided has been used due to time and processing power restrictions. Sample of the dataset has been given shown in Figure 3.4.

Figure 3.4(a)

Figure 3.4(b)

Figure 3.4 AWS spacenet challenge dataset sample

The spacenet dataset provides building footprints as annotations. The annotations are in the GeoJSON format. GeoJSON is an open standard format designed for representing simple geographical features, along with their non spatial attributes[34]. GeoJSON format is based on JSON(JavaScript Object Notation). GeoJSON format consists of following parameters

3.1.2.1 Type

Type parameter defines the type of annotation. In this case, type parameter is FeatureCollection as the GeoJSON consists of features.

3.1.2.2 features

Features parameter contains an array of objects containing the feature collection of the specific image. The object consists of the type i.e. feature, properties which is an object consisting of details about the annotation along with the annotated timestamp, version, changeset, user, flag if the building is present, area, id as well as other parameters and geometry which is object consisting of the type of annotation i.e Polygon and the array of coordinates. This format is not very useful for the training straight away. However, after some processing the annotations were converted to the corresponding bounding box coordinates.

3.1.3 Custom Dataset

Another dataset used in this work has been manually created and annotated. For the dataset, Satellite imagery has been acquired from Bing Satellite Images[31] (due to best possible image present) via the SAS Nightly Software. Satellite imagery has been acquired for the areas in the Sindhupalchok District, Nepal. The imagery were of the zoom level of 20 as higher zoom only increased the size of image but was not clear. The zoom level 20 provided the most affordable resolution. These acquired images are quite large in size. For the ease of annotation, the images are cropped in the size of 512 * 512. These slices are smaller in size and can be used for the training. The generated slices of original images are annotated via the VGG annotator tool[32]. This tool provides the annotations in the MS COCO format. The MS COCO Format has been explained in detail in section 3.1.1 and is shown in Figure 3.3. Over all 993 images were annotated. Sample of dataset has been provided below:

Figure 3.5(a)

Figure 3.5(b)

Figure 3.5 Sample of Custom dataset for Nepal

3.2 Preprocessing

The acquired datasets needed to be pre processed before proceeding with the training process. The applied preprocessing steps are shown in section 3.2.1

3.2.1 Crowd AI Mapping Challenge Dataset

No preprocessing was required for the crowd ai mapping challenge as the dataset was well balanced and the annotations were in MS COCO format which is suitable for the training. The dataset images were less than 50 KB which made it efficient for the training on local PC.

3.2.2 AWS Spacenet Dataset

For the spacenet dataset, The images were over 0.5MB and the building footprints were in the form of latitude and longitude. The footprints were converted to bounding box using the utilities provided by the spacenet[24]. To handle large images, Masks were generated using the obtained bounding box and training was done only these masks. This method although decreased the time for training but due to loss of features showed very poor detections. To increase the accuracy, the large images were resized to 515*512. This reduced the size of image without significant loss of the features. After resizing, the annotation was converted to the Pascal VOC format which is suitable for the Yolo v3 classifier training. The masked image along with its original images are shown in Figure 3.5.

Figure 3.6(a)

Figure 3.6(b)

Figure 3.6(c)

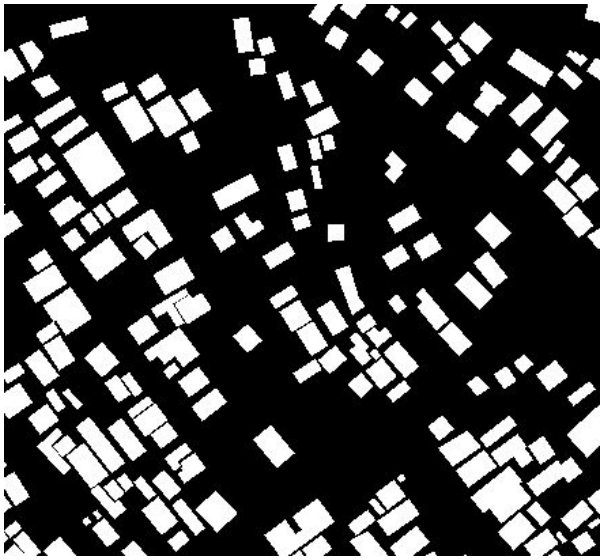

Figure 3.6(d)

Figure 3.6 Ground truth masks for annotated dataset sample

Custom dataset has been used to train both RPN and YOLO v3. For RPN, custom dataset annotations was already in MS COCO format thus no preprocessing was required. For the YOLO v3, the annotations were converted in the Yolo V3 annotation format. The sample of the annotation format is shown in figure below:

```
sindhupalchok/cropped/1_IMG-15.jpg 452,358,496,405,21
sindhupalchok/cropped/1_IMG-16.jpg 32,411,65,447,21 318,471,367,510,21 385,473,426,511,21 478,4
sindhupalchok/cropped/1_IMG-17.jpg 1,461,21,502,21
sindhupalchok/cropped/1_IMG-18.jpg
sindhupalchok/cropped/1_IMG-19.jpg 495,155,510,187,21 482,121,510,154,21 466,64,507,96,21 451,1
sindhupalchok/cropped/1_IMG-20.jpg 401,156,429,188,21 361,120,403,160,21 327,89,373,136,21 306,
sindhupalchok/cropped/1_IMG-21.jpg 390,393,418,419,21 412,319,449,348,21 288,364,311,384,21 241
sindhupalchok/cropped/1_IMG-22.jpg 466,438,501,466,21 402,414,444,479,21 363,468,402,510,21 431
sindhupalchok/cropped/1_IMG-23.jpg 322,488,357,507,21
sindhupalchok/cropped/1_IMG-24.jpg 313,470,354,506,21 50,416,125,466,21 5,456,47,504,21
sindhupalchok/cropped/1_IMG-25.jpg 132,43,187,96,21 101,6,135,48,21 331,112,365,157,21 385,1,43
sindhupalchok/cropped/1_IMG-26.jpg 163,115,194,143,21 101,111,135,152,21 93,177,140,203,21 45,1
sindhupalchok/cropped/1_IMG-27.jpg
sindhupalchok/cropped/1_IMG-28.jpg
sindhupalchok/cropped/1_IMG-29.jpg 175,1,210,20,21 224,2,256,24,21 328,4,375,42,21 405,43,446,7
sindhupalchok/cropped/1_IMG-30.jpg 104,40,152,100,21 324,181,359,214,21 484,272,508,311,21 469,
sindhupalchok/cropped/1_IMG-31.jpg 421,470,444,500,21 426,436,464,475,21 89,386,119,425,21 63,3
sindhupalchok/cropped/1_IMG-32.jpg 155,52,187,91,21 79,43,114,69,21 52,36,80,58,21 0,25,51,89,2
sindhupalchok/cropped/1_IMG-33.jpg 136,427,172,459,21 69,436,118,481,21 41,405,96,437,21 1,459,
sindhupalchok/cropped/1_IMG-34.jpg 392,360,431,396,21 347,377,379,408,21 101,379,152,416,21 71,
sindhupalchok/cropped/1_IMG-35.jpg 48,325,84,360,21 111,237,217,296,21 83,280,122,322,21 303,26
```

Figure 3.7: Yolo v3 annotation format

3.3 Region Proposal Network

The Region Proposal Network is the backbone of Faster R-CNN[25]. The RPN has proven to be very efficient to propose objects in image. To detect regions of images where object lies a small network is slide over a convolutional feature map that is the output by the last convolution layer. The architecture of RPN is shown in figure 3.8

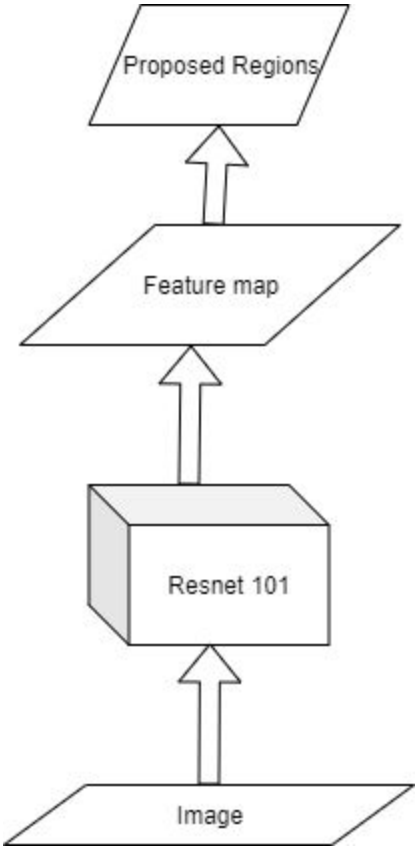

Figure 3.8 RPN Architecture

RPN can be defined as a shallow fully connected neural network (NN) first introduced in the Faster R-CNN (Faster region convolutional neural network) for proposing regions with a high probability of containing an object of interest[35]. RPN contains a classifier and regressor. RPN employs the usage of anchors which are basically the central point of the sliding window. In RPN, Classifier determines the probability of proposal having target object while Regressor regresses the coordinates of the proposals. In default configuration for the RPN, number of anchors per pixel is taken as 9 i.e 9 proposals are generated for every pixel in the image. However in this work, 21 anchors has been taken due to various sizes of buildings in satellite imagery. The total number of proposals for image can be given as:

$$P = W * H * K \quad \text{..... EQ 3.1}$$

Where W = Width of image

H = Height of image

K = Proposals per pixel

P = Total number or proposals

In this work, image with width and height of 512*512 is taken. Therefore,

W = 512

H = 512

K = 21

P = 512 *512 * 21 = 5,505,024 per image

Anchor placement has been shown for single pixel in figure 3.8.

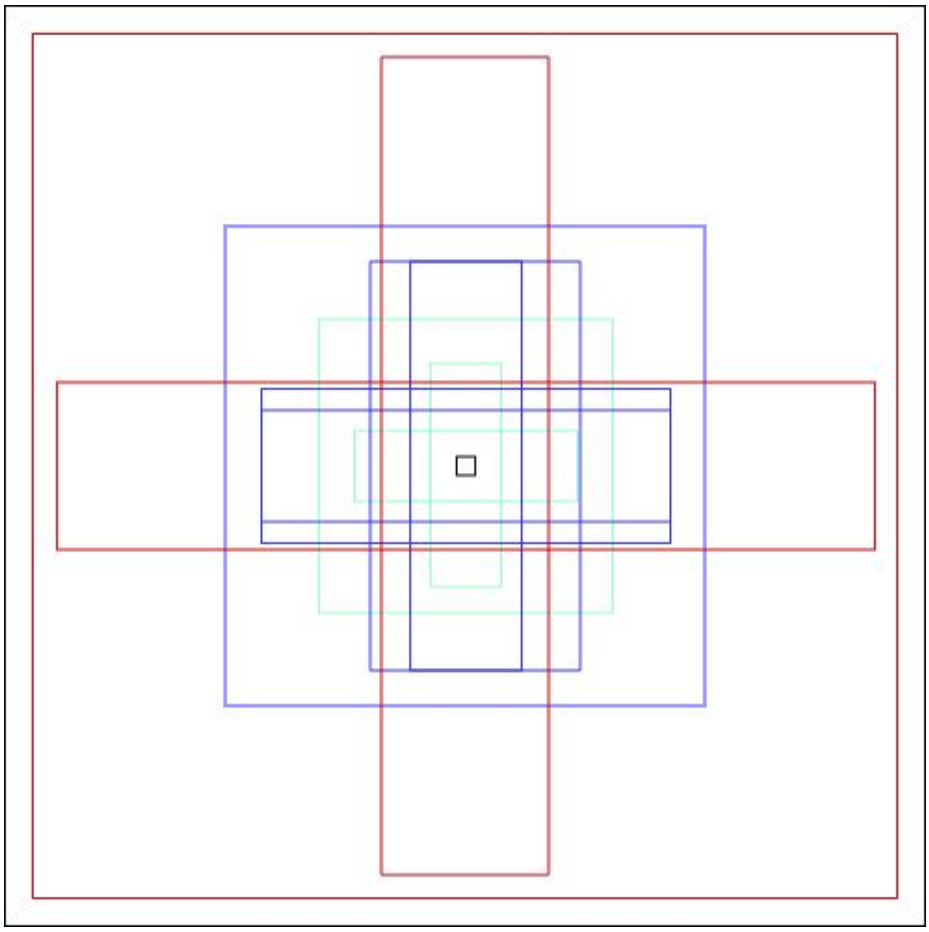

Figure 3.9: Anchor placement for pixel

As it can be seen, number of proposals are very large. To reduce the number of proposals, RPN has been used. RPN can be build over the Resnet, VGG, AlexNet and DeepNet. In this work, Resnet 101[26] has been used as the convolutional network to generate feature map from the image.

3.3.1 Resnet

Resnet is flexible network that can have a variety of deep networks of upto 152 layers. Resnet introduced skip connection to fit the input from the previous layer to the next layer without any modification of input. In this work, 101 layer network has been used.

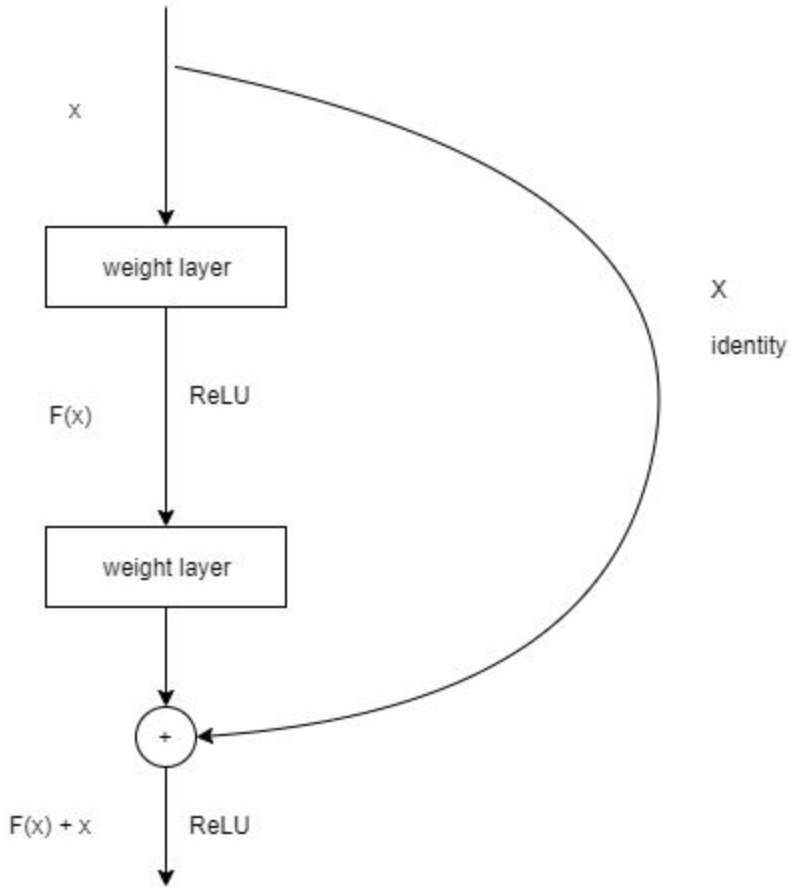

Figure 3.10 Skip Connections in Resnet

Skip connection architecture skips certain layers in the neural net and feeds the output of one layer to another layer directly as well as other layers.Skip connections are used to feed the information captured in initial layers and were lost during the processing to the required layers.

Resnet 101 architecture mainly consists of multiple blocks connected by the skip connections. The main architecture of the Resnet 101 architecture has been shown in Figure 3.11

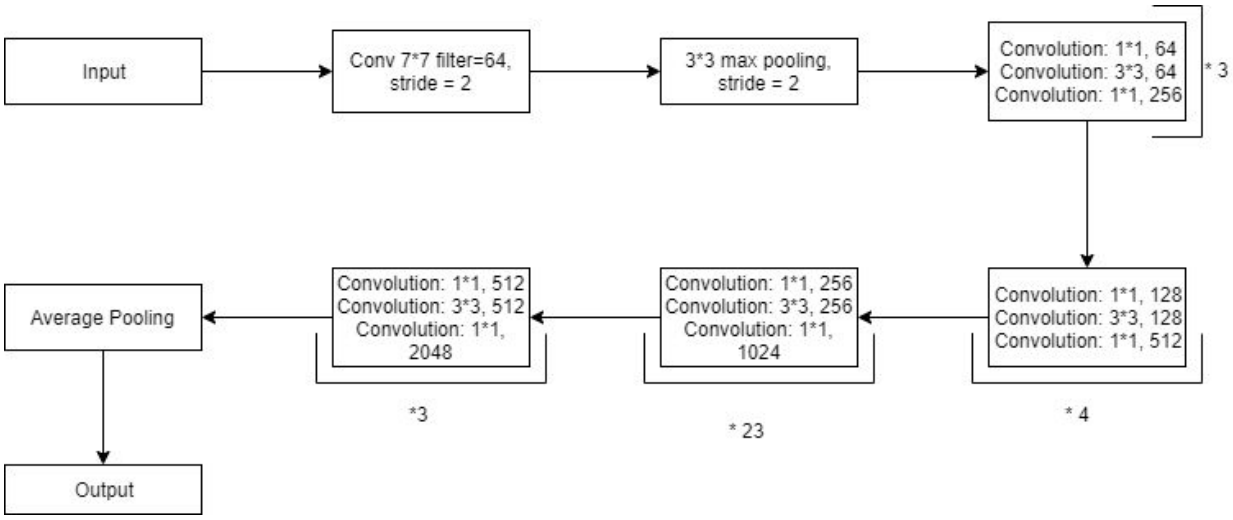

Figure 3.11: Resnet 101 Architecture

Resnet architecture employs the usage of skip connections. Resnet architecture employs the usage of convolution layer at first with filter size of 64,kernel size of 7*7 and stride with value 2. After convolution, max pooling is used with stride of 2. After max-pooling, Convolution layer with kernel size of 1*1 and filter of size 64 is used. The output max pooling is concatenated with the output of recent convolution layer and fed as input to the next convolution layer with size of 3*3 and filter size of 64. The output of layer before the recent layer is appended with the output of recent layer and is fed as input to the next layer. The same process is repeated as per the Figure 3.8. After the convolution operations are done, Average pooling is done on the obtained feature map.

3.3.1.1 Input layer

Input layer are predefined layers of keras which is used to instantiate a keras tensor [27]. Tensor can be any object from the underlying backend (Theano, TensorFlow or CNTK). Since, tensorflow has been used in this work, Input layer instantiates the Tensorflow object. This layer is used to data to the network. The input layer object can be used multiple times to concatenate with other layers.

3.3.2 Convolution layer

Convolution layer is used to create convolutional kernel that is convolved with the input layer. This architecture incorporated in this work utilizes 1D convolutional layer and 2D convolutional layer multiple times to produce required output of tensors. Convolution layer is generally used for the images. A sample of convolution is shown in Figure 3.12

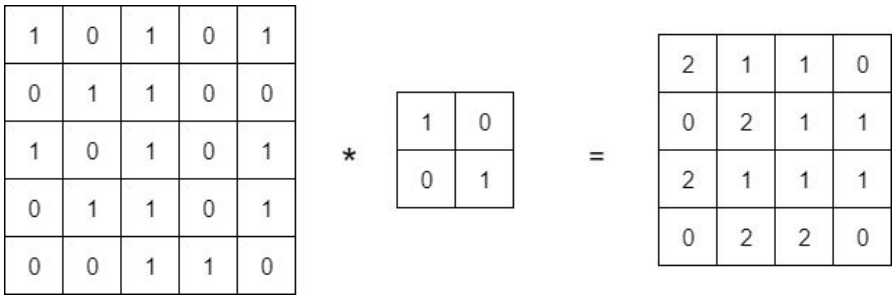

Figure 3.12: Convolution Operation

3.3.3 Batch Normalization layer

Batch normalization layer is applied to the output tensor to normalize the activations of the previous layer, i.e. applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1[27].

3.3.4 Activation layer

In Artificial neural networks, the activation function of a node defines the output of that node given on input or set of inputs[28]. In this method, Rectified linear unit activation function has been used. This layer returns the node values between the max defined output and 0. This method limits the output of the node. Mathematically ReLU is given as

$$y = max(0, x) \quad \text{EQ 3.2}$$

3.3.5 Max pooling layer

In this work, max-pooling layer has been integrated to downsample the input by reducing its dimensions. In this work, 2D max pooling layer has been integrated. This layer applies moving

window across the output matrix and returns max value within that window as output.An example for 2D max pooling layer is given below:

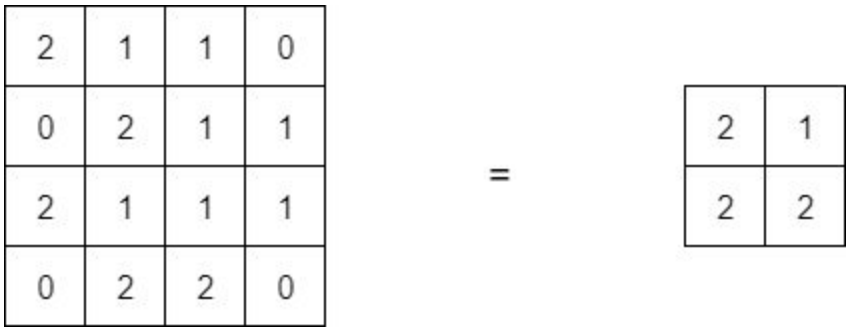

Figure 3.13: Max Pooling Operation

3.3.6 Upsampling layer

This layer up samples the input to a higher dimension. An example for the upsampling operation is shown in Figure 3.14

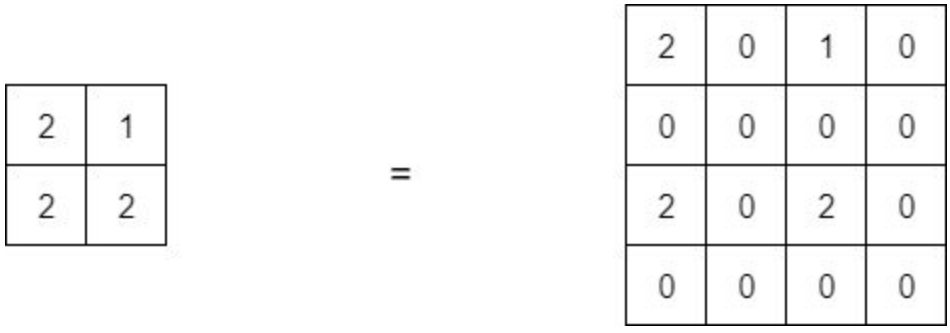

Figure 3.14: Upsampling operation

As it can be seen in figure 3.13, 2*2 matrix has been sampled to the 4*4 matrix.

3.3.7 Dense layer

Dense layer is also known as fully connected layer. Dense layer defines how the neurons are connected to the next layer of neurons. An example for the dense layer is shown in figure 3.15

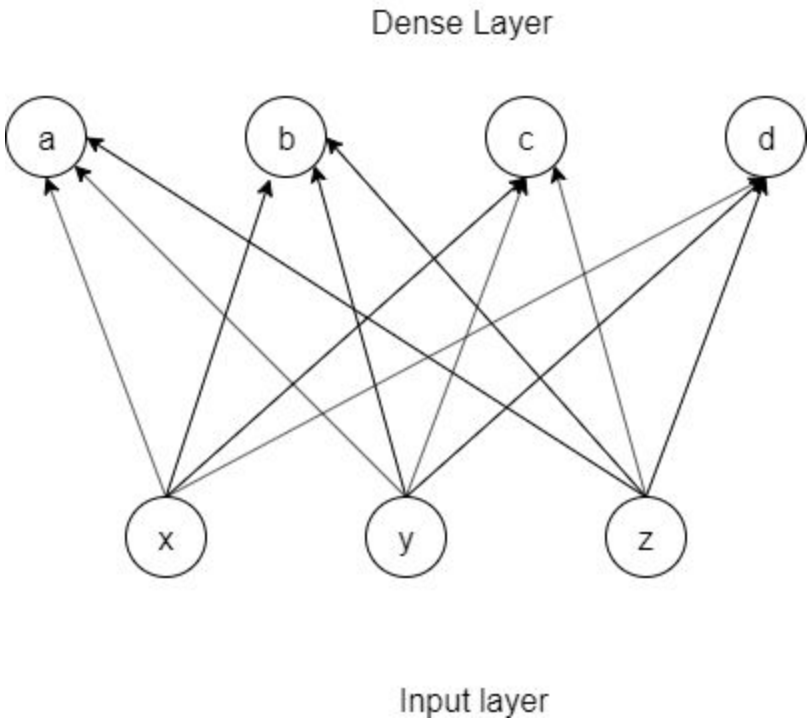

Figure 3.15: Dense layer

As it can be seen in Figure 3.15, All the neuron layers are connected to a dense layer weights which are in turn connected to the next layer.

3.3.8 Average Pooling layer

Average pooling layers acts like the max pooling layer as shown in section 3.3.5. The only difference between both is while the max pooling layer selects the max value while the average pooling selects the average of all the values. Average pooling maintains the maximum number of values over the max pooling as it considers all values. An example of average pooling for 2D is shown in Figure 3.16.

128	0	1	1
0	128	1	1
0	0	128	0
0	0	2	2

=

64	1
0	33

Figure 3.16: 2D average pooling operation

As it can be seen in Figure 3.16, average pooling selects the average of all values rather than maximum value thus maintaining larger number of features over max pooling.

After the feature map is generated, Intersection-over-union of every proposal is calculated. The proposal are assigned label based on following points[35].

1. The anchors with highest Intersection-over-union overlap with a ground truth box.
2. The anchors with Intersection-Over-Union Overlap higher than 0.7.

Intersection-over-union is calculated by the following equation

$$IoU = (A \cap GT)/(A \cup GT),\ A = actual\ box,\ GT = Ground\ Truth \ldots\ldots\ldots \text{EQ 3.3}$$

RPN outputs array of 4 numbers i.e x,y,w,h where (x,y) are the center of anchor box while w, h are the width and height of the proposed area respectively. As discussed above, RPN is a neural net trained to extract possible regions containing the objects in required category. In order to train the RPN, crowdAI mapping challenge dataset was used. For the training Mask R-CNN implementation by the matterport was referenced[29]. The training was done using train test split. The dataset was split in 60% training data, 20% validation data and 20% test data. The training was done on Inter(R) Core(TM) i5-7200 CPU @2.50GHz. 30 epochs were run for training. The training was completed in 14 days. The training obtained the validation loss of 5%.

For the training of RPN with custom dataset of Nepal, the dataset was split in 60% training data, 20% validation data and 20% test data. The training was done on the Google Colab Notebook with GPU environment and 12GB ram. Overall 160 epochs were run for training. The training obtained the validation loss of 18%.

3.4 Region of interest

The region proposal network provides us with the so called “proposals”. This proposal regions are extracted from the original image using Python Pillow library for the extraction of Region of interests. The obtained proposals are used as regions of interests and are sent as input to the Yolo v3 classifier

3.5 Yolo v3 classifier

Yolo[30] is the abbreviation for You only look once. Yolo V3 is a state of the art object detection algorithm. The Yolo algorithm is able to detect objects in real time and can be used to detect objects in video. Yolo V3 is an advanced version of Yolo and detects image upto 4x faster over its predecessor. Yolo applies single neural network to single image. The network divides image into regions and predicts bounding boxes and probabilities for each region. The yolo is in category of one stage detectors. One stage detectors are applied only once over the whole image at test time Yolo is comparatively faster than R-CNN but fails on satellite imagery as the satellite imagery is usually very large size. This implementation of Yolo V3 uses Darknet architecture. The architecture of Yolo v3 is given below:

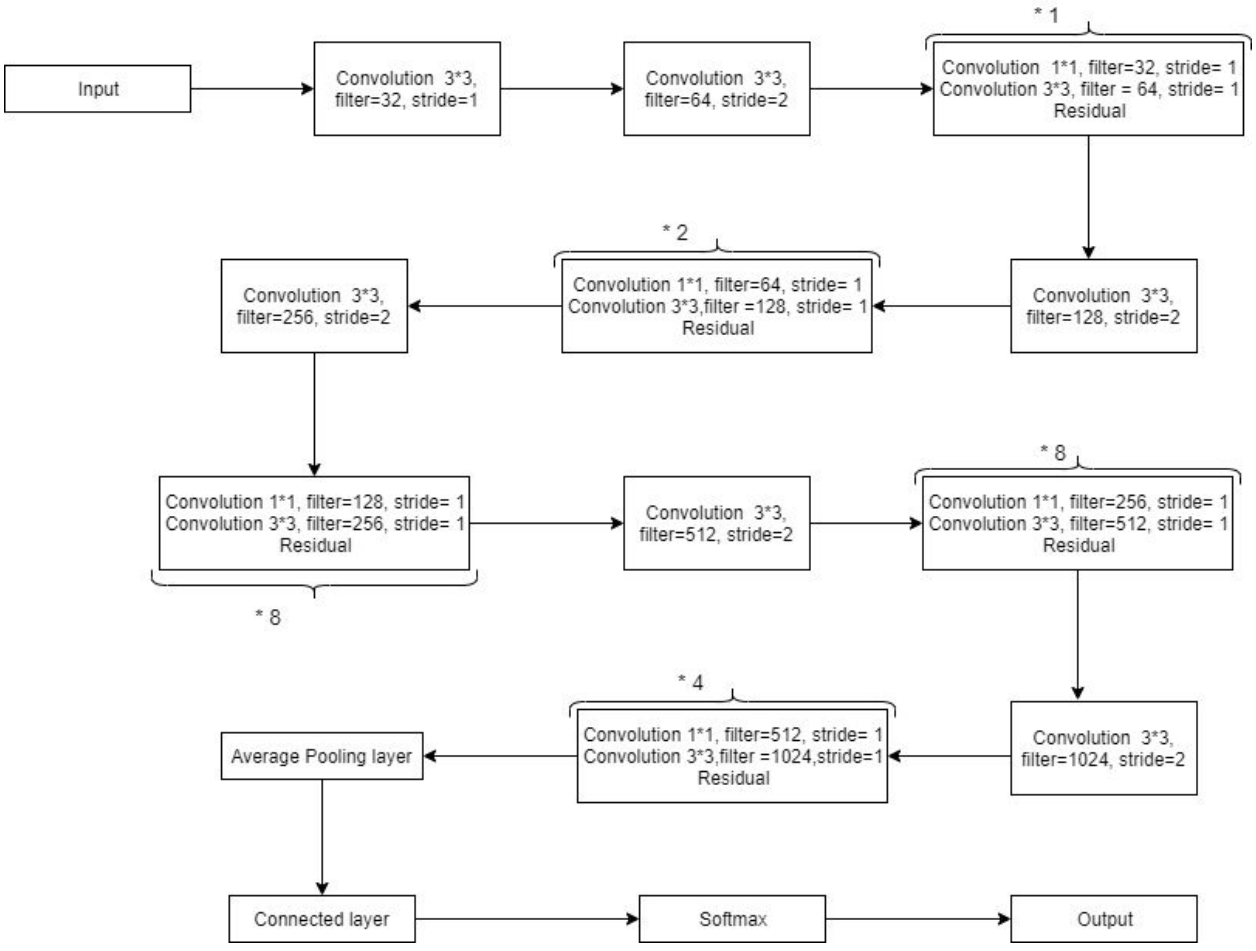

Figure 3.17:Yolo Architecture

As it can be seen in figure 3.17, yolo architecture uses multiple convolution layer in combination with average pooling layer and connected layer. For the activation function, yolov3 uses the softmax activation function.

3.5.1 Softmax activation function

Softmax function converts numbers into probabilities that sums up to 1.It is generally used to map the non-normalized output of a network to a probability distribution over predicted output classes[36].

Yolo v3 detects large number of objects in images but doesn't work in the case of satellite imagery.The default Yolo v3 classes doesn't include building let alone in satellite Imagery. So in order to accomplish our task of making yolo classifier to work on our method, Custom class of building is added to train the model using Spacenet dataset and annotations converted to Pascal VOC format. The dataset consisted of 6000 images. The dataset was split in 60% training, 20% validation and 20% test data. The training was concluded in two stages. On first stage only the heads of the network are trained. This is done to decrease the validation loss in short span of time. The heads were trained for 50 epochs. Once the heads are trained, weights are saved. In the second stage, overall network is trained for next 50 epochs. The block diagram of training Yolo v3 shown in Figure 3.18.

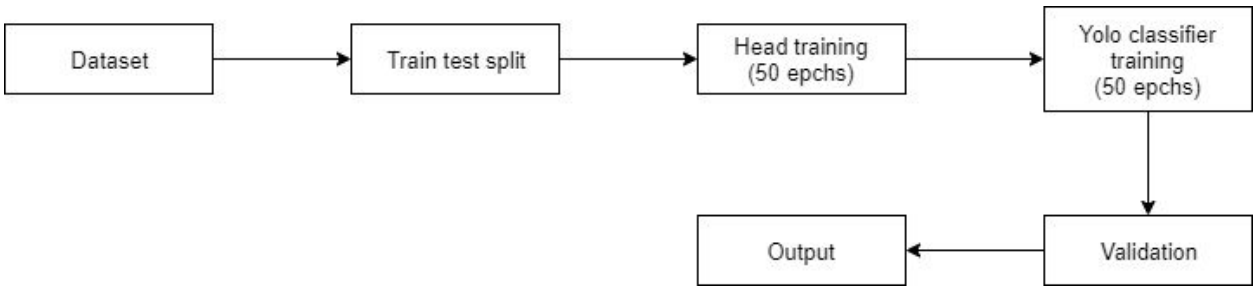

Figure 3.18 Training block diagram of Yolo v3 classifier

The training was performed on Inter(R) Core(TM) i5-7200 CPU @2.50GHz. The training was completed in 8 days. The training resulted in validation loss of 11.5%. The model is coarse due

to being not trained on large dataset. Dataset has been acquired for multiple cities from the official site of AWS spacenet.

The same training method has been applied to the Custom dataset training with Yolo V3. The heads are trained for 50 epochs to decrease the validation loss in short span of time. Once the heads are trained, weights are saved. In the second stage, overall network is trained for next 50 epochs. The training was done on the Google Colab Notebook with GPU environment and 12GB RAM. The training resulted in validation loss of 13.46%.

3.6 Building Extraction

Building extraction is the main section of this method. The ROIs generated from the RPN network are passed via the Yolo classifier. The ROIs which are said to be buildings are counted as buildings. The combination of these methods works quite well to detect buildings in small imagery but fails to detect buildings in large satellite imagery. In normal case, A single satellite imagery of an area is over 4000*4000 pixels. During experimentation, It was found out that the integrated method performs poorly in large imagery. In order to solve this issue, Sliding window approach was integrated. This approach passed only a section of image through the process and accumulates the results. As the complete image has been passed through the process, the accumulated results are compiled to give the predictions. The sliding window method greatly improved the accuracy of our method to detect buildings in large imagery. The output of the method is given later under Results section.

3.7 Assessment

Once the predictions are accumulated, The number of buildings detected in Pre-disaster and Post-Disaster images are compared. Based on this comparison, rapid damage assessment has been done. This method provides initial assessment of damage in the area. Assessment for Indonesian earthquake was done. The images for this assessment were obtained from Digital

Globe website. A sample assessment has been done for the Pipal data, Sindhupalchok, Nepal using images obtained from the Google Earth Pro.

3.8 With sliding window approach

The algorithm of the pipeline of the modified methodology with sliding window approach is shown in Figure 3.19:

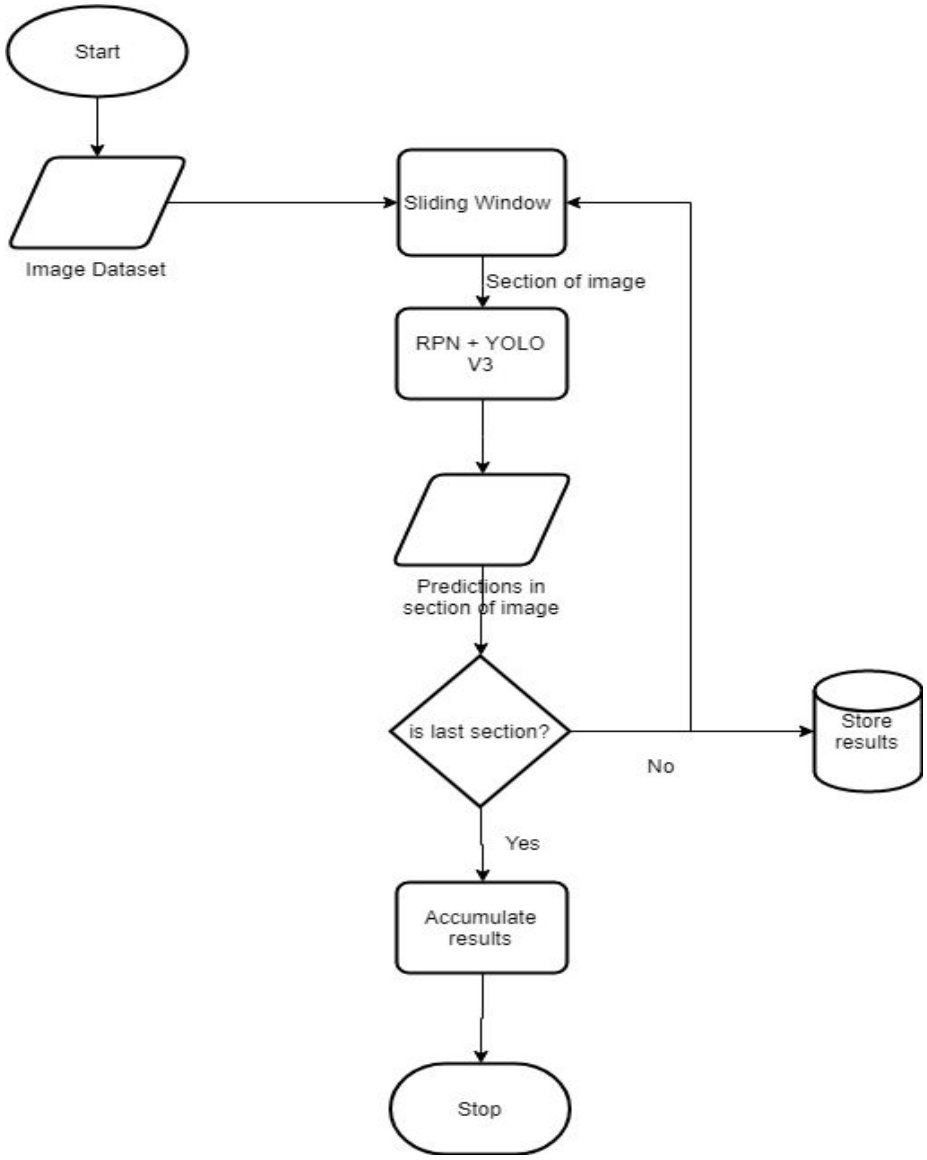

Figure 3.19 Flowchart for the combined approach

As it can be seen in Figure 3.19, sliding window reduces the memory consumption for processing larger images but also increasing the time required to perform detection in large satellite imagery.

3.9 Evaluation Metrics

The used method is evaluated via the precision and recall of the output. Precision can be defined as the fraction of relevant instances among the retrieved instances while Recall (Sensitivity) can be defined as a fraction of the total amount of relevant instances that were actually retrieved. Precision P and Recall R is given by the equation

$$P = (TP)/(TP + FP) \text{ …….. EQ 3.4}$$

$$R = (TP)/(TP + FN) \text{ ……...EQ 3.5}$$

Where TP is the true positive, FP is false positive and FN is false negative.

Overall accuracy can be mathematically expressed in terms of P and R and is given as

$$F = 2 * P * R/(P + R) \text{..... .EQ 3.6}$$

Where F is the f1 score.

4. Experiments and Results

4.1 Morphological Operators

The first method tried for extracting building was done using morphological operators over the image and segmentation using the surface reflectance. This method incorporated multiple use of the opening and closing operators to reduce the noise. Surface reflectance was incorporated to segment the buildings from the other objects. The output from the above algorithm is shown in figure 4.1.

This method relied heavily on the surface reflectance. As the surface reflectance of the building differs from scene to scene as well as building to building. One value of surface reflectance used to segment in one image may not be best suited for another image. Due to this problem, this method was rejected.

Figure 4.1(a)

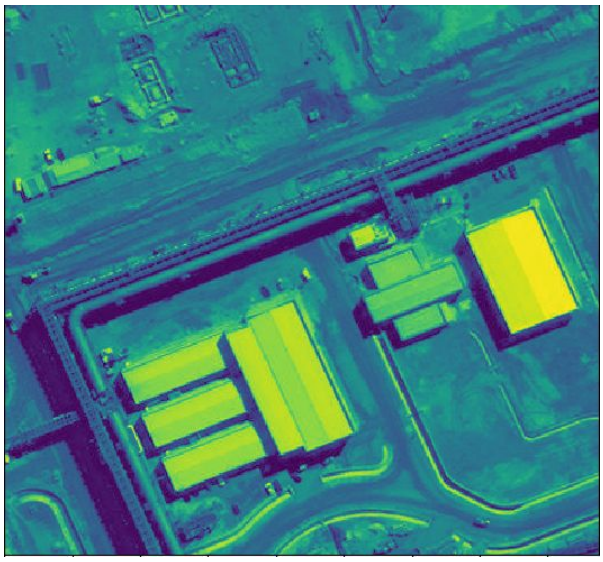

Figure 4.1(b)

Figure 4.1(c)

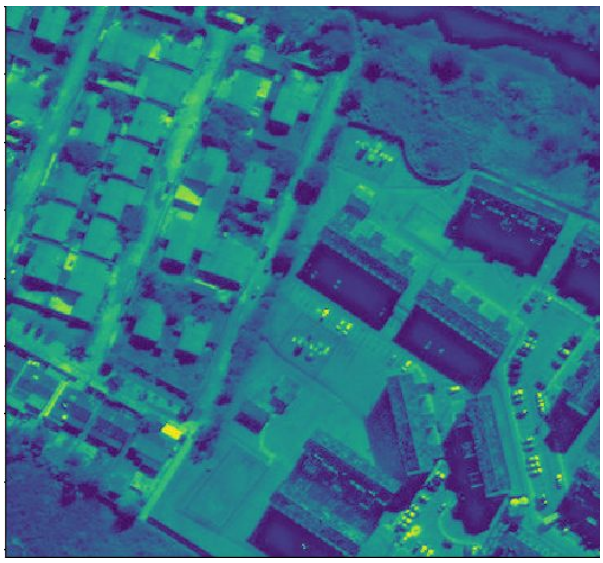

Figure 4.1(d)

Figure 4.1 Output of morphological operators

4.2 Mask R CNN

Mask R CNN being the most sophisticated algorithm for the double stage detector for image segmentation was developed by Facebook AI Research group[21]. This algorithm is an improvement over the older algorithms in series of R-CNN group. This algorithm implemented ROI pooling over the already existing backbone of Faster R-CNN i.e RPN. This method is suitable for the instance segmentation and predicts masks for the predicted segmentation. Mask R-CNN was trained on custom annotated dataset. The annotation was generated using VGG annotator tool[32]. The generated annotation was converted to the MS COCO format to be suitable for the training. This work utilized the two stage procedure with Region Proposal Network in the first stage and Predicting the class and box offset.The output of Mask R-CNN for the satellite imagery with size of 438 * 406 pixels along with the original image is shown in Figure 4.2.

Figure 4.2(a)

Figure 4.2(b)

Figure 4.2(c)

Figure 4.2(d)

Figure 4.2 Detection output of the Mask R-CNN for smaller images

As we can see in the Figure 4.2, Mask R-CNN gives the classification with significant accuracy in smaller images. As seen in the first image above, masks are predicted over 25 out of 35 buildings present. While in second image out of 6 buildings, masks are predicted over 5 buildings. However for the larger satellite images, this method fails to predict miserably as this algorithm needs to resize the image for prediction. The output for large satellite imagery (990 * 660) 4.3(a) is shown in Figure 4.3(b).

Figure 4.3(a) Original Satellite Image

Figure 4.3(b) Detection results of Mask R-CNN on larger image

As we can see in the Figure 4.3(b), The masks only overlaps over the 6 buildings over a large area. This method is slower to train and predict. The accuracy of this method is great in smaller images but fails to predict on larger images. To increase the accuracy, we shift towards much sophisticated algorithm.

4.3 Yolo V3

Yolo V3 is single stage detector which is used to detect objects in the image[30]. The algorithm applies over the whole image. The network divides image into regions and predicts bounding boxes and probabilities for each region. This method has great accuracy for the classifier but fails on larger images. The output for the implemented algorithm for smaller image is shown in Figure 4.4.

Figure 4.4(a)

Figure 4.4(b)

Figure 4.4(c)

Figure 4.4(d)

Figure 4.4 Detection result of Yolo v3 on smaller images

As shown in Figure 4.4, the trained yolo v3 classifier classifies much larger area and provide more output over the Mask R-CNN. Here, it can be noticed that the yolo v3 although giving more predictions also predicts non building areas as buildings. Here it should be noted that yolo v3 classifier classifies 31 out of 42 buildings correctly in above given images while incorrectly classifies 3 non building areas as buildings. The output of yolo classifier on large scale image is shown in Figure 4.5.

Figure 4.5(a) Original Image

Figure 4.5(b) Detection result of yolo v3 on large image

As it can be seen in Figure 4.5(b), this method classifies much better in satellite imagery over the Mask R-CNN. However it can be seen that, number of wrong areas has also been classified as buildings. This method has great accuracy for the smaller images but fails on larger images. Due to less accuracy on larger images on the amount of time required for the classification, new approach is implemented to combine backbone of Mask R CNN (RPN) and Yolo classifier. Implemented approach is described below

4.4 Combined approach

Due to shortcomings of above approach, A new approach is integrated to improve the process for the building extraction. The implemented approach uses a combination of Region Proposal Network to detect regions containing specific objects in our case 'building' and Yolo V3 classifier to classify if the proposed regions are buildings or not. The output of the combined approach for kunchok, sindhupalchok is given below:

Figure 4.6 Detection result of combined approach on larger images

As we can see, the number of buildings detected is very bad. To solve this issue, Sliding window approach is integrated in the pipeline. Under this method, A section of image is passed via the process at once and the output is accumulated. Once all the sections of images are processed, Accumulated output is mapped to the original image and output is shown. The output of the combined approach with sliding window method is shown in Figure 4.7

Figure 4.7 Detection result of combined approach with sliding window on large image

As it can be seen in Figure 4.7, Combined approach along with Sliding window approach has produced better results than other methods implemented above. One other prediction was done to test how the results vary for the large satellite imagery. The implemented approach was tested on Nepal's Pipal Danda.The image was acquired via the Google Earth pro. The output of implemented approach for Pipal Danda,Nepal is Figure 4.8

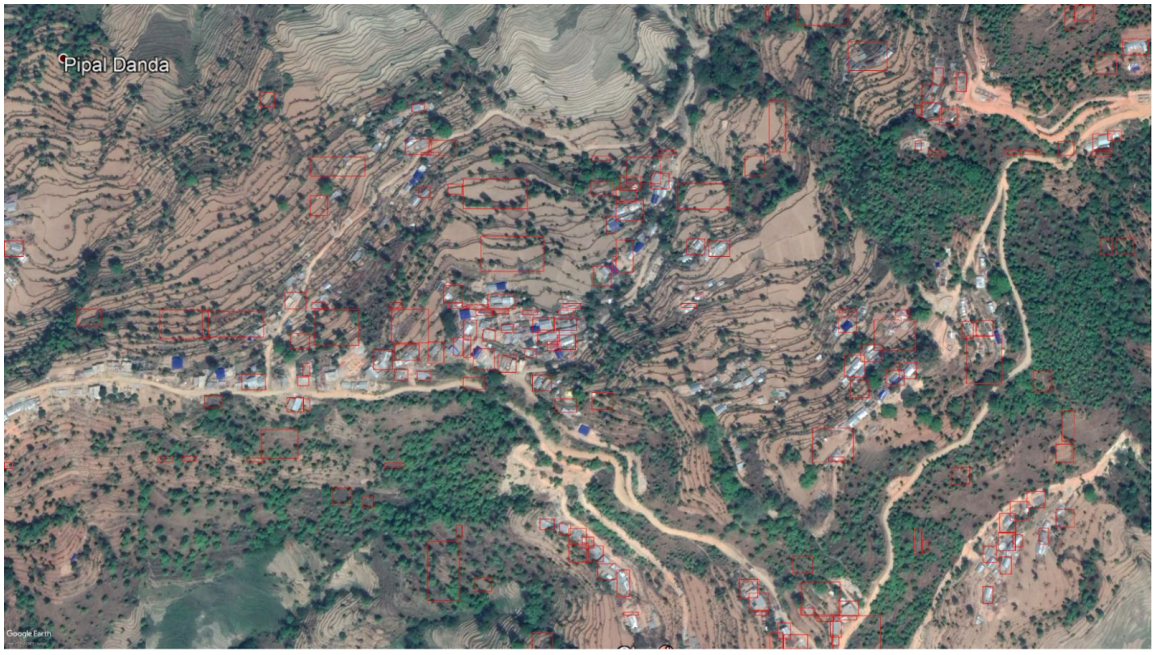

Figure 4.8 : Detection result on pipal danda

The multiple colored borders for bounding boxes are removed and only single color has been used to draw bounding box above the structure. As we can see in the figure 4.10, bounding box are detected over large number of buildings.

4.6 Custom Dataset

Both RPN and Yolo V3 were trained on manually annotated images of the Sindhupalchok district. The detection result shown for the kunchok, Sindhupalchok by the network trained on custom dataset is shown below:

Figure 4.9 Custom Dataset trained model prediction

As it can be seen in the figure above, the predictions are more accurate over the model trained on international standard dataset. The output of the combined approach trained on Custom dataset for multiple areas in nepal are shown below:

Figure 4.10(a)

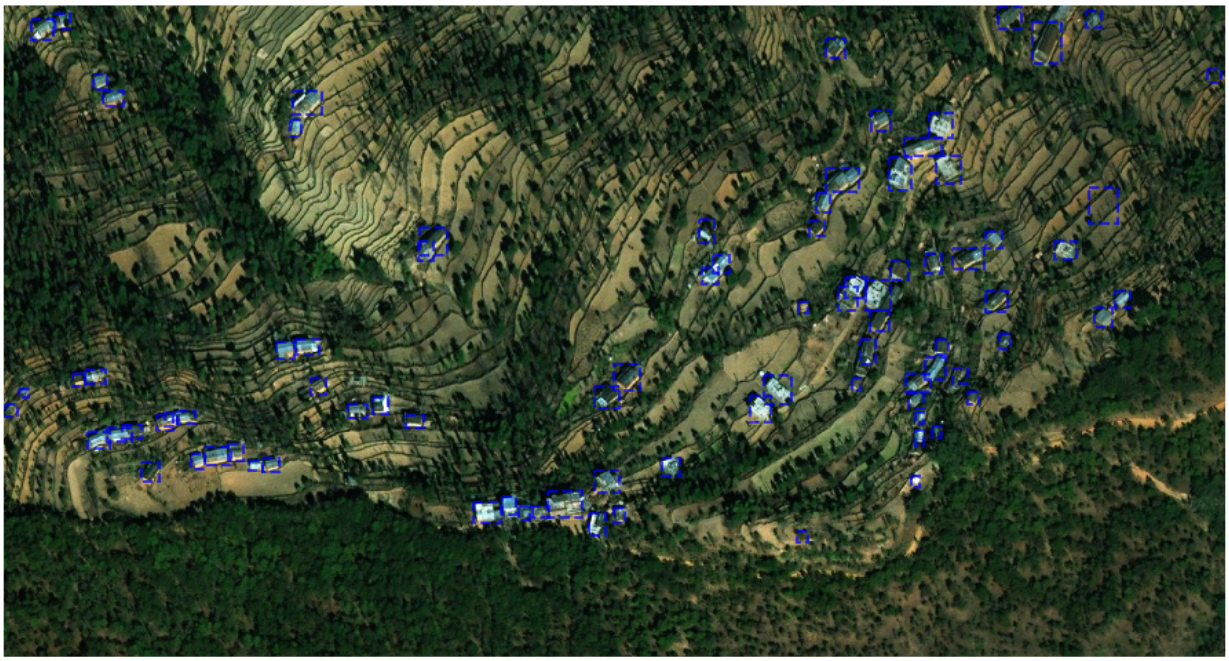

Figure 4.10(b)

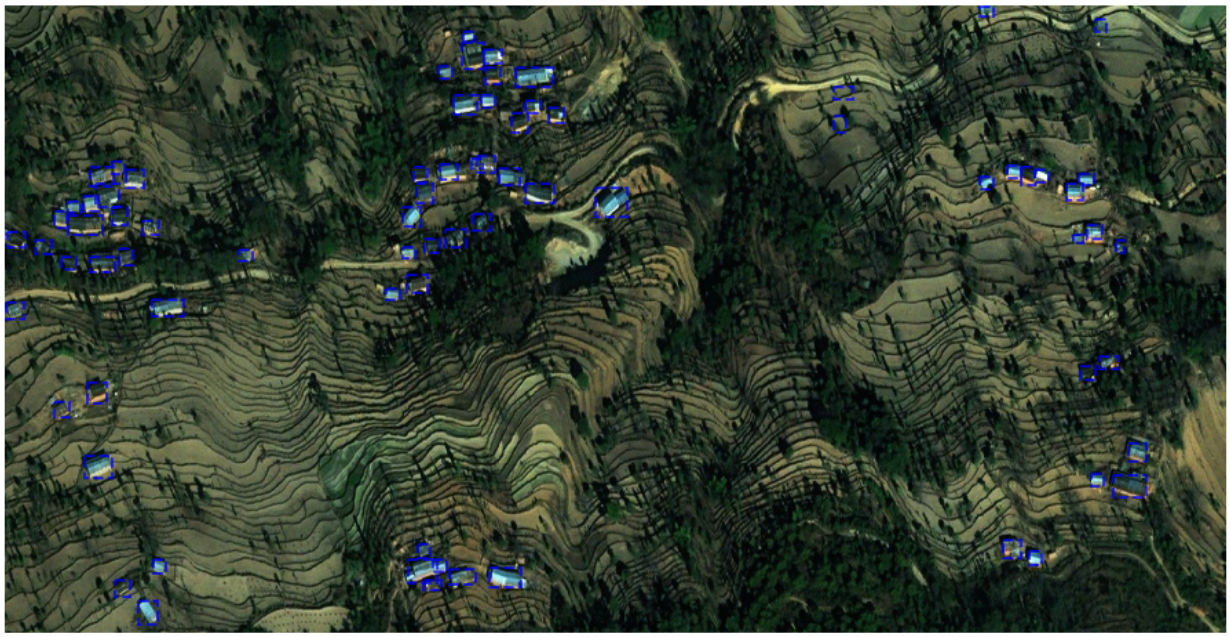

Figure 4.10(c)

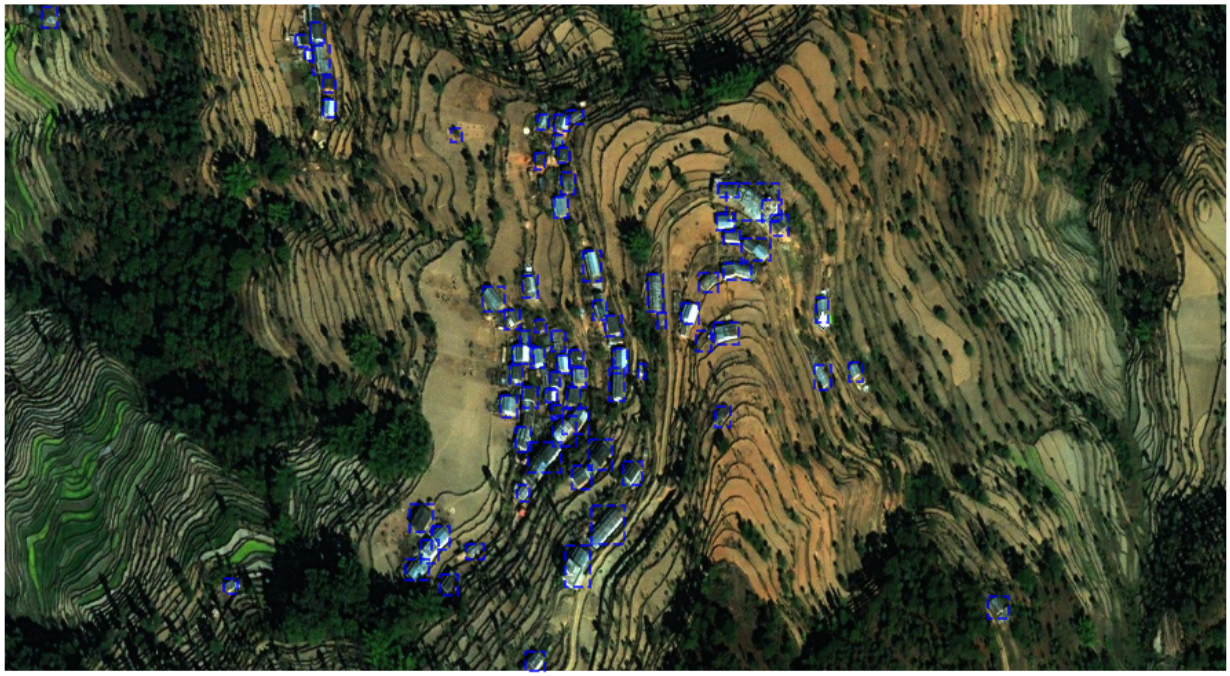

Figure 4.10(d)

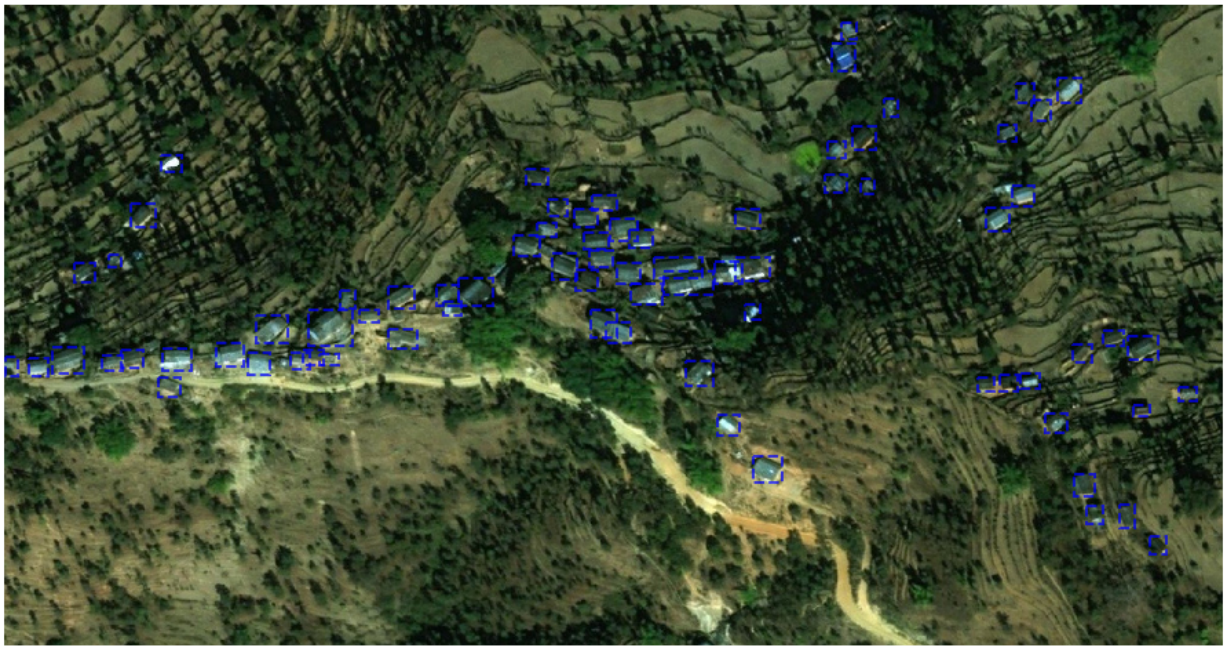

Figure 4.10(e)

Figure 4.10 Various detection results on Custom Dataset trained model

As it can be seen in Figure 4.10, custom trained dataset detects better for Nepal. It also predicts separate buildings as well as in the busy areas.

4.7 Custom Dataset Assessment

For the assessment of the disaster, Satellite imagery for Pipal Danda, Nepal of multiple timeline was obtained from Google earth pro. Image of the same location captured in 2018,2016 and 2015 after earthquake was obtained and assessment was done on the captured images to show the changes the specified area has gone over the interval of 3 years. An attempt was made to capture image just before 2015 earthquake but was not possible due to unavailability of day time images. The output for assessment for the Pipal Danda, Nepal is shown below in various images. At first assessment was done on the imagery of 2018 to show the present buildings in the area. Then 2016 and 2018 images were analyzed. At last the building count is compared and the percentage changes have been calculated.

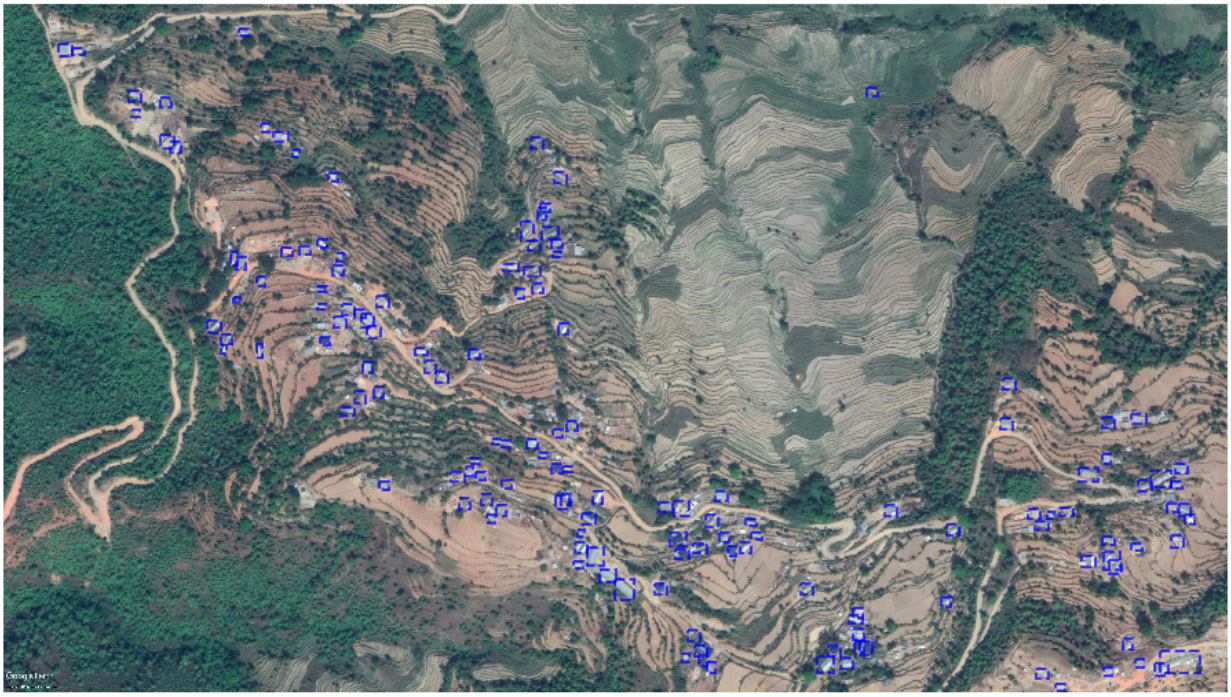

Figure 4.11 2018 image of Pipal Danda, Nepal

The above image has been captured in 2018 and acquired via Google Earth Pro. The combined approach detect 145 buildings.

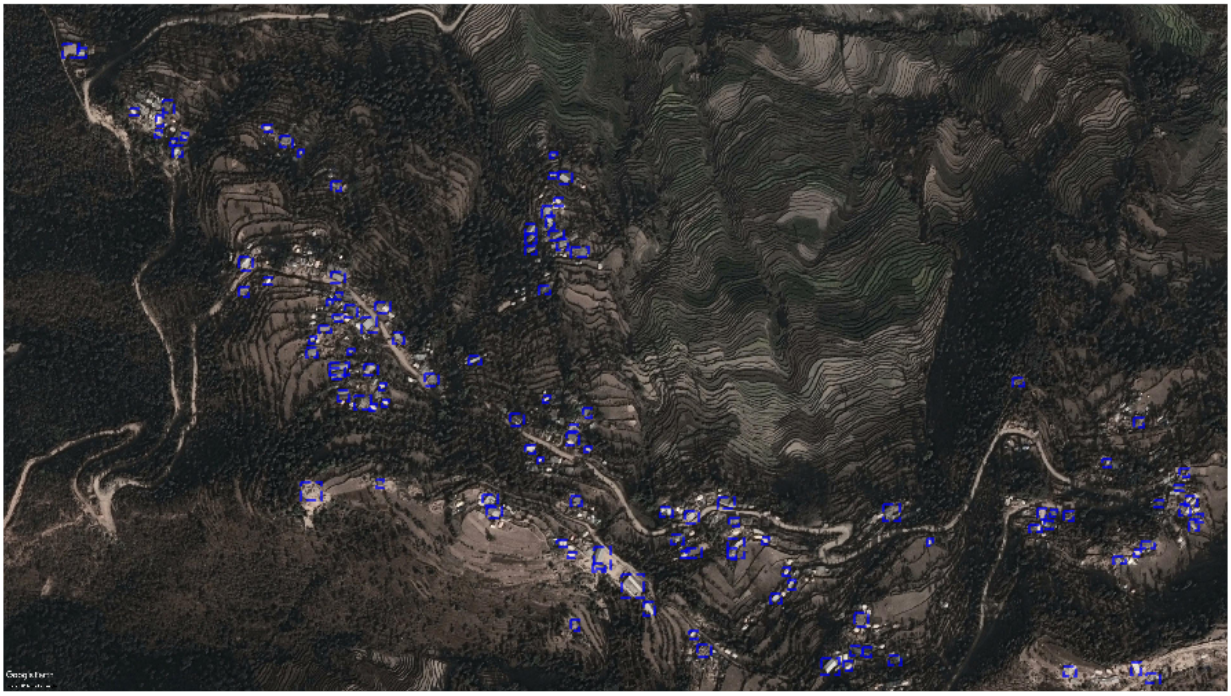

Figure 4.12 2016 image of Pipal Danda Nepal

The above image has been captured in 2016 and acquired via Google Earth Pro. The combined approach detect almost 115 buildings. As we can see, the number of buildings have changed by 26.08% in 2 years. Similarly, detection results on 2015 acquired images has been shown below:

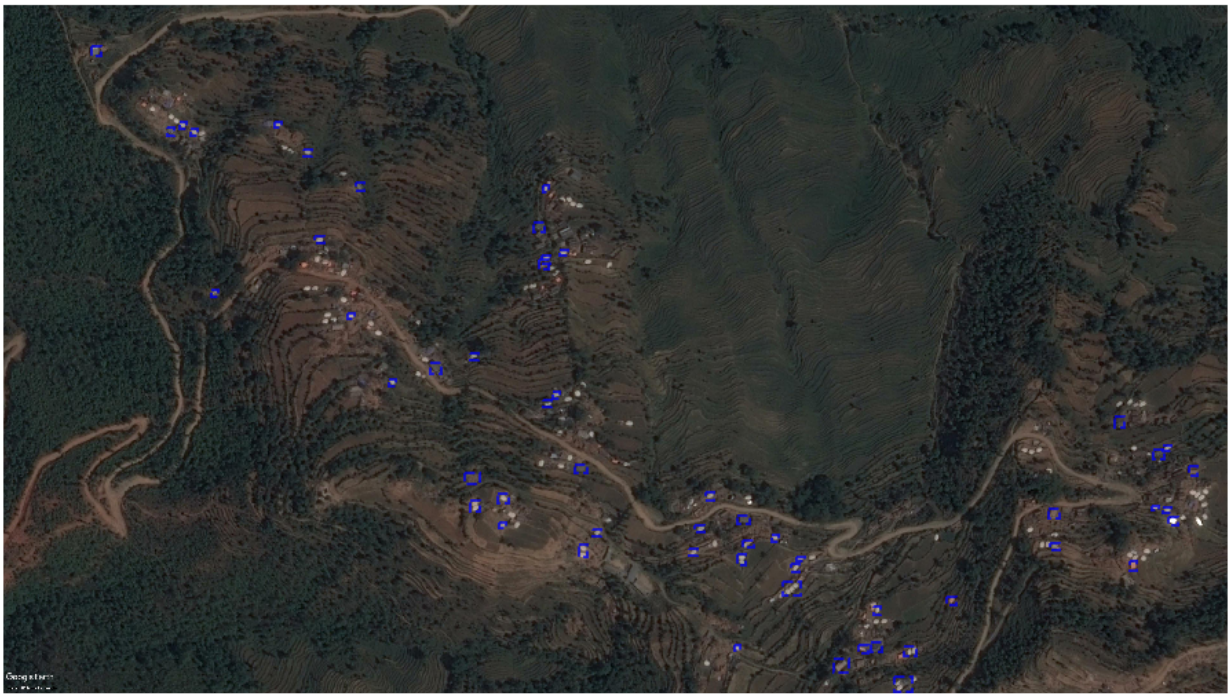

Figure 4.13 2015 image of Pipal Danda Nepal

The above image has been captured in 2015 just after the earthquake and was acquired via Google Earth Pro. The combined approach detect almost 55 buildings. This shows that the number of buildings have changed by 62.06%. This can be used to assess the damage the area has gone through. In this way, Damage assessment can be done.

4.8 Experiments

Experiments have been run to test the split of data. The experiment has been run on 60/40, 70/30,80/20 splits for yolo classifier. The training vs validation loss graph for the experiments has been shown below:

4.8.1 60-40 split

The training vs validation loss plot for 74 epochs is given below:

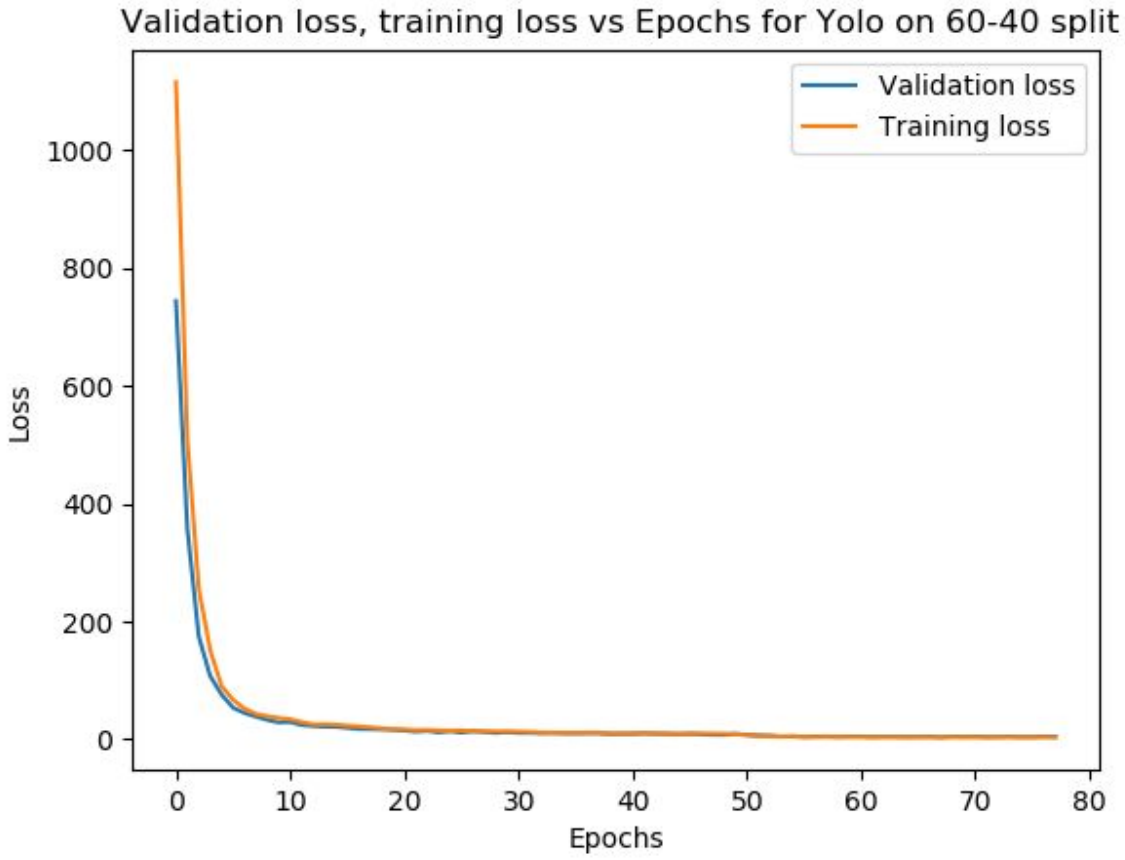

Figure 4.14: Training vs Validation loss for Yolo v3 classifier

As we can see in the Figure 4.14, the training loss and validation loss starts in thousand's at first but decays slowly over time. At first, validation loss is less than training loss, this is due to the small batch size. Training loss is calculated on the last batch while the validation loss is calculated on all validation dataset. However, after 42 epochs, model has learned, validation loss overshoots the training loss, Training and validation loss converge on 66 epochs but due to the use of callbacks, the training is done for extra 10 epochs so check if validation loss can be decreased. At the end, validation loss is 4.1% while training loss is 3.66%.

4.8.2 70-30 split

The training vs validation loss for 70-30 split is given below:

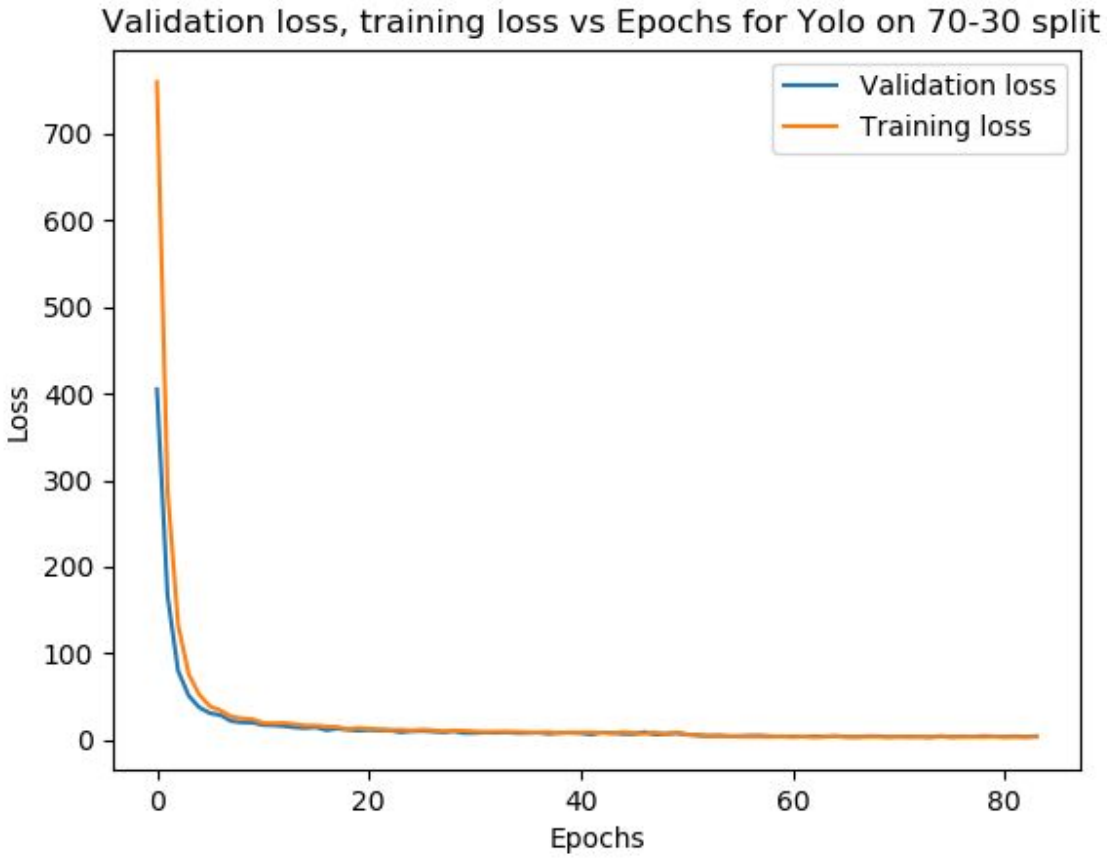

Figure 4.15: Training vs Validation loss for 70-30 split for Yolo v3 classifier

As we can see in the Figure 4.15, the training loss and validation loss starts in thousand's at first but decays slowly over time. At first, validation loss is less than training loss, this is due to the small batch size. Training loss is calculated on the last batch while the validation loss is calculated on all validation dataset. However, after 50 epochs, model has learned, validation loss overshoots the training loss, Training and validation loss converge on 75 epochs but due to the use of callbacks, the training is done for extra 10 epochs so check if validation loss can be decreased. At the end, validation loss is 4.55%..

4.8.3 80-20 split

The training vs validation loss graph for 80-20 split is given below:

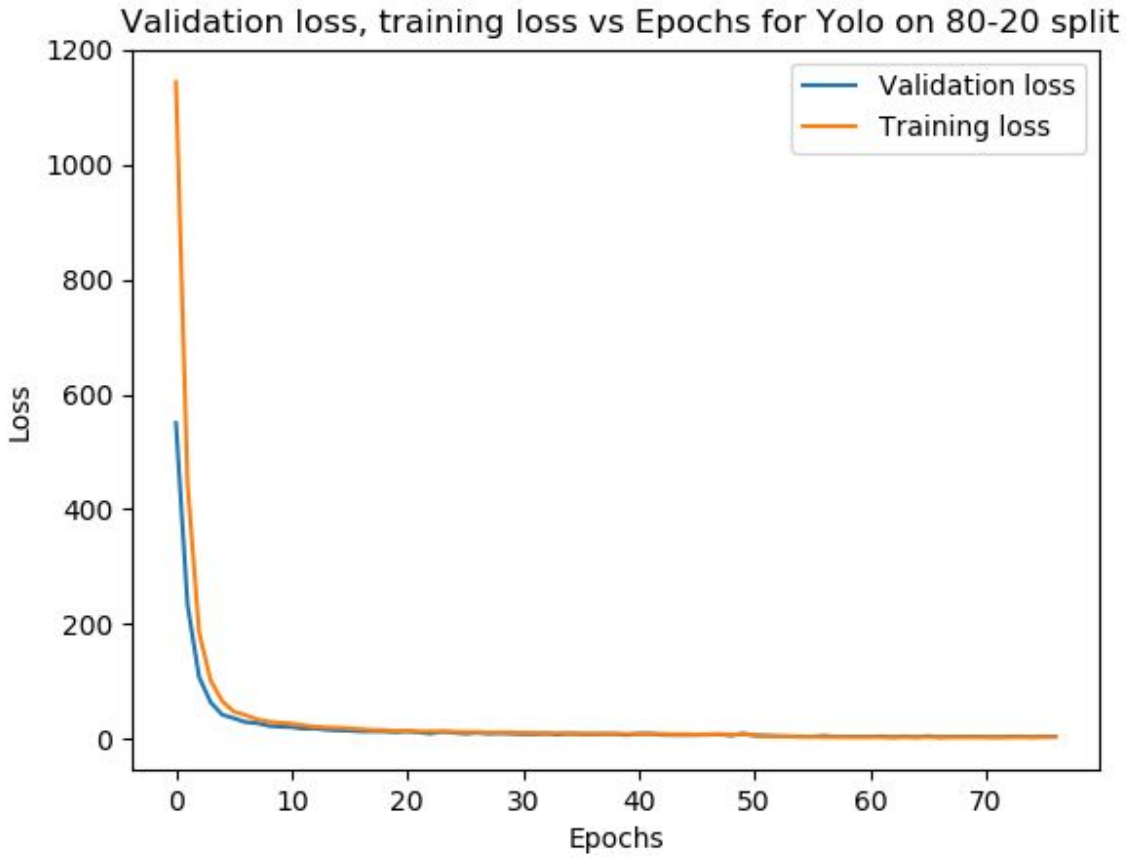

Figure 4.16: Training vs validation loss for 80-20 split

In this experiment, the validation loss over shoots the training loss at 48 epochs. Training and validation loss converge on 67 epochs and the training is done for next 10 epochs to check if it can be reduced. At the end of the training, validation loss is 4.58%

As it can be seen from the experiments above, the loss is less for 60-40 split. So 60-40 split has been used for yolov3 classifier.

Another experiment has been run to check how the custom dataset trained model works against the international dataset trained model. The detection result of model trained on both custom dataset and international dataset is shown below

4.8.4 International Dataset Prediction:

Prediction of international dataset training is given in Figure 4.17

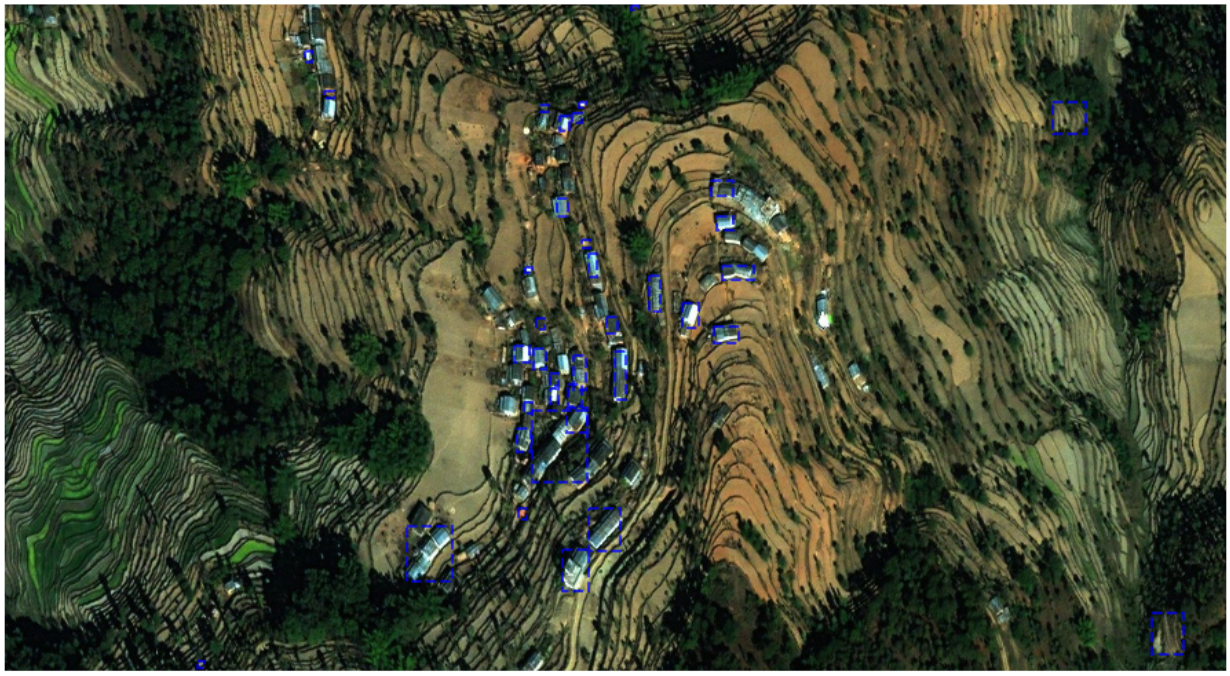

Figure 4.17: International dataset prediction

The model trained on international dataset gives the prediction as 37 buildings count. As it can be seen in the Figure 4.17, a lot of buildings are not detected by this model. Smaller buildings are not detected while there are multiple wrong predictions.

4.8.4 National Dataset Prediction

Prediction on dataset of nepal is given in Figure 4.18

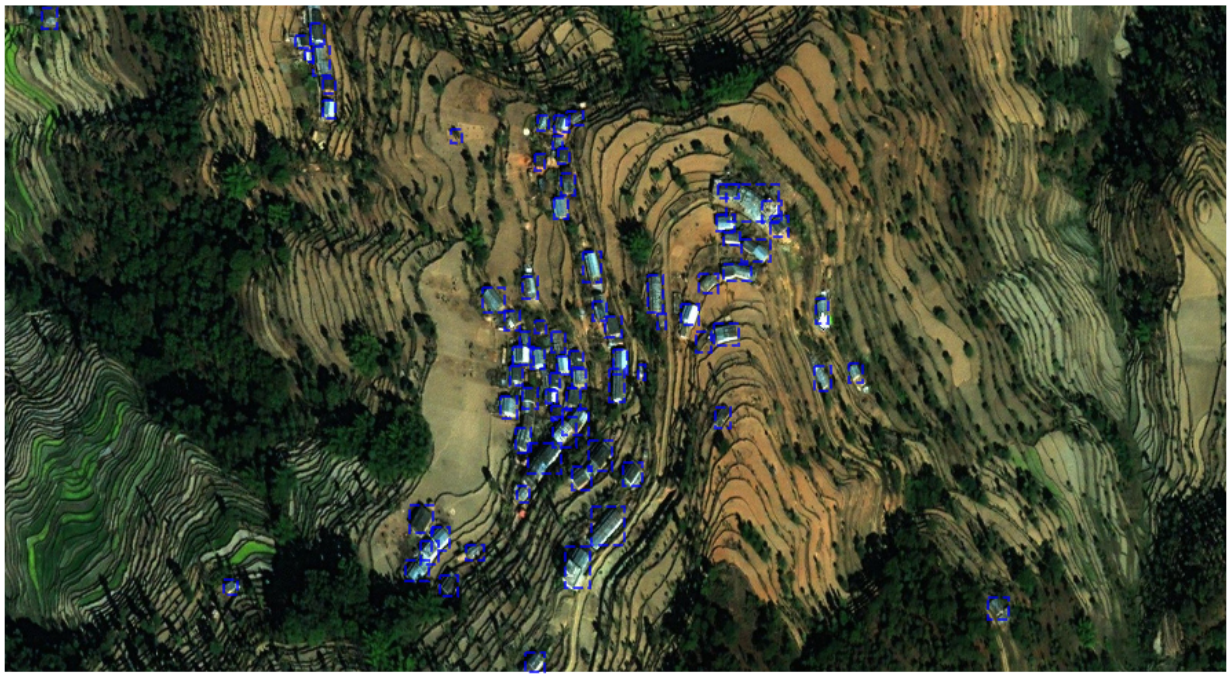

Figure 4.18: Prediction of model trained on Custom Dataset

The model trained on national dataset gives the predicted count of 77 buildings into same image.

4.9 Evaluation

Evaluation of the method was done via confusion matrix. Once the confusion matrix has been obtained, multiple scores were calculated i.e Precision, Recall and F1 score. This method achieved the overall F1 score of 0.89 as well as Precision of 0.94 and Recall of 0.86

	Predicted: No	Predicted: Yes	
Actual: No	6	5	11
Actual: Yes	6	75	81
	12	80	

Precision = 0.94

Recall = 0.86

F1 score = 0.89

5. Conclusion and Recommendation

The research was conducted for suitable method for building detection. Multiple methods like Morphological Operators, Mask R CNN and Yolo V3 were tested to find the suitable method. Morphological operator being dependent on the reflectance was not found suitable for the work. Mask R-CNN and Yolo V3 were the final candidates for the work. So, the backbone of R-CNN network family i.e the Region Proposal Network was used along with the Yolo v3. For the training, international dataset was used at the start of the training but Manually annotated Custom dataset for Nepal was used. Set of experiments were done to find the split ratio for training and validation. Once the training was complete, Detection was done on multiple satellite imagery of Nepal acquired from Google Earth Pro. Detections were fairly good.

For the assessment, Images of same area acquired over multiple time range was acquired. Detection was made on the acquired image to detect the number of absent buildings. On the basis of absent buildings the assessment was made.

At present, the custom dataset is small so the predictions are still missing. Increasing the data set will help solve this problem. Currently, assessment is done based on the number of missing buildings which is not a very good factor for assessment. Using the difference in the pixel values of building over the pre disaster and post disaster imagery will make a good assessment. The method doesn't predict well on the dark images thus giving the wrong predictions. Using gray scale images for the training could result in better predictions in dark illuminated images.

References

1. Dey, Sourav. (2015). A Devastating Disaster: A Case Study of Nepal Earthquake and Its Impact on Human Beings.

2. Remote sensing. (2019, September 23). Retrieved from https://en.wikipedia.org/wiki/Remote_sensing.

3. Ghandour, A., & Jezzini, A. (2018). Autonomous Building Detection Using Edge Properties and Image Color Invariants. Buildings, 8(5), 65. doi: 10.3390/buildings8050065

4. Liu, W., & Prinet, V. (n.d.). Building detection from high-resolution satellite image using probability model. Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS 05. doi: 10.1109/igarss.2005.1525759

5. Ankit, U. (2019, February 6). Semantic Segmentation of Aerial images Using Deep Learning. Retrieved from https://towardsdatascience.com/semantic-segmentation-of-aerial-images-using-deep-learning-90fdf4ad780

6. Abburu, S., & Golla, S. B. (2015). Satellite Image Classification Methods and Techniques: A Review. International Journal of Computer Applications, 119(8), 20–25. doi: 10.5120/21088-3779

7. Weng, L. (2018, December 27). Object Detection Part 4: Fast Detection Models. Retrieved from https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html

8. Glumov, N. I., Kolomiyetz, E. I., & Sergeyev, V. V. (2000, January 27). Detection of objects on the image using a sliding window mode. Retrieved from https://www.sciencedirect.com/science/article/pii/003039929593752D.

9. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. doi: 10.1145/3065386

10. Yamazaki, F., & Zavala, C. (2013). SATREPS Project on Enhancement of Earthquake and Tsunami Disaster Mitigation Technology in Peru. Journal of Disaster Research, 8(2), 224–234. doi: 10.20965/jdr.2013.p0224

11. Dellacqua, F., & Gamba, P. (2012). Remote Sensing and Earthquake Damage Assessment: Experiences, Limits, and Perspectives. Proceedings of the IEEE, 100(10), 2876–2890. doi: 10.1109/jproc.2012.2196404

12.Liu, W., & Prinet, V. (n.d.). Building detection from high-resolution satellite image using probability model. Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS 05. doi: 10.1109/igarss.2005.1525759

13. Amy Zhang, Xianming Liu, Andreas Gros, Tobias Tiecke (2017, July 07), Building Detection from Satellite Images on a Global Scale

14. L. Chiroiu,G. André, (2001), damage assessment using high resolution satellite imagery: application to 2001 bhuj, india, earthquake

15. Jenis, Archana. (2012). Earthquake Damage Assessment of Buildings Using Pre-event and Post- event Imagery

16. Worldhttp, G. (2018, March 16). Disaster Assessment Of Earthquake Using GIS and Remote Sensing.Retrieved from https://www.geospatialworld.net/article/disaster-assessment-of-earthquake-using-gis-and-remote-sensing /.

17. Huyck, C. K., Adams, B. J., Cho, S., Chung, H.-C., & Eguchi, R. T. (2005). Towards Rapid Citywide Damage Mapping Using Neighborhood Edge Dissimilarities in Very High-Resolution Optical Satellite Imagery—Application to the 2003 Bam, Iran, Earthquake. Earthquake Spectra, 21(S1), 255–266. doi: 10.1193/1.2101907

18. Parape, Chandana & Premachandra, Chinthaka & Tamura, Masayuki & Sugiura, Masami. (2012). Damaged building identifying from VHR satellite imagery using morphological operators in 2011 Pacific coast of Tohoku Earthquake and Tsunami. International Geoscience and Remote Sensing Symposium (IGARSS). 3006-3009. 10.1109/IGARSS.2012.6350793.

19. Hamaguchi, R., & Hikosaka, S. (2018). Building Detection from Satellite Imagery using Ensemble of Size-Specific Detectors. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). doi: 10.1109/cvprw.2018.00041

20. Rastiveis, Heidar & Samadzadegan, Farhad & Reinartz, Peter. (2013). A fuzzy decision

21. Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick (2017-2018). Mask R-CNN. arXiv:1703.06870

22. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2016.91

23. Datasets. (n.d.). Retrieved from https://www.crowdai.org/challenges/mapping-challenge.

24. Datasets. (n.d.). Retrieved from https://spacenetchallenge.github.io/datasets/datasetHomePage.html.

25. Ross, Sun, & Jian. (2016, January 6). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Retrieved from https://arxiv.org/abs/1506.01497.

26. Zhang, Ren, Sun, & Jian. (2015, December 10). Deep Residual Learning for Image Recognition. Retrieved from https://arxiv.org/abs/1512.03385.

27. Keras: The Python Deep Learning library. (n.d.). Retrieved from https://keras.io/.

28. Activation function. (2019, November 12). Retrieved from https://en.wikipedia.org/wiki/Activation_function.

29. Mask R-CNN [online] Available: https://github.com/matterport/Mask_RCNN

30. Redmon, Joseph and Farhadi, Ali (2018). YOLOv3: An Incremental Improvement

31. Bing, Microsoft, https://www.bing.com/maps/aerial.

32."VGG Image Annotator (VIA)." Visual Geometry Group - University of Oxford, http://www.robots.ox.ac.uk/~vgg/software/via/

33. Gkioxari, et al. "Mask R-CNN." ArXiv.org, 24 Jan. 2018, https://arxiv.org/abs/1703.06870.

34."GeoJSON."Wikipedia,Wikimedia Foundation, 21 Oct. 2019, https://en.wikipedia.org/wiki/GeoJSON.

35. Karmarkar, T. (2019, July 23). Region Proposal Network (RPN) - Backbone of Faster R-CNN. Retrieved from https://medium.com/egen/region-proposal-network-rpn-backbone-of-faster-r-cnn-4a744a38d7f9

36. Softmax function. (2019, November 7). Retrieved from https://en.wikipedia.org/wiki/Softmax_function.

37. Precision and recall. (2019, November 7). Retrieved from https://en.wikipedia.org/wiki/Precision_and_recall.