**APPLICATION OF MACHINE LEARNING TECHNIQUES ON**

**REMOTE SENSING DATA**

*by*

|  |  |  |
| --- | --- | --- |
| Name | Roll No. | Registration No: |
| Abhishek Agrahari | CSE2017/059 | **171170110002**of 2017-2018 |
| Subhadip Nandi | CSE2017/055 | **171170110107**of 2017-2018 |
| Rahul Raj | CSE2017/054 | **171170110068**of 2017-2018 |
| Rudrapratap Ghosh | CSE2017/033 | **171170110081**of 2017-2018 |

## Under the guidance of

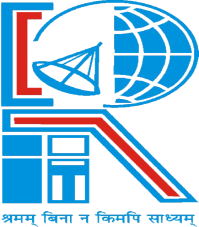
**Ms. Srirupa Das**

## Dept. of Computer Science & Engineering

## RCC Institute of Information Technology

## PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

## BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERINGRCC INSTITUTE OF INFORMATION TECHNOLOGY

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## RCC INSTITUTE OF INFORMATION TECHNOLOGY

## [Affiliated to West Bengal University of Technology] CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**RCC INSTITUTE OF INFORMATION TECHNOLOGY**

****

**TO WHOM IT MAY CONCERN**

I hereby recommend that the Project entitled

**“APPLICATION OF MACHINE LEARNING TECHNIQUES ON REMOTE SENSING DATA”** prepared under my supervision by **Abhishek Agrahari** (171170110002of 2017-2018), **Subhadip Nandi** (171170110107of 2017-2018), **Rahul Raj** (171170110068of 2017-2018), **Rudrapratap Ghosh** (171170110081of 2017-2018) of B. Tech (7th Semester), may be accepted in partial fulfillment for the degree of **Bachelor of Technology in Computer Science & Engineering** under Maulana Abul Kalam Azad University of Technology(MAKAUT).

…………………………………………

Project Supervisor

Department of Computer Science and Engineering

RCC Institute of Information Technology

Countersigned:

………………………………

Head

Department of Computer Sc. & Engg., RCC Institute of Information Technology Kolkata – 700015



# **CERTIFICATE OF APPROVAL**

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR EVALUATION OF PROJECT

**1.**

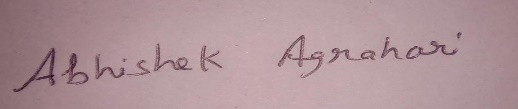
**2.**

**3.**

**(Signature of Examiners)**

ACKNOWLEDGEMENT

We acknowledge our overwhelming gratitude & immense respect to our revered guide, **Ms. Srirupa Das** (Assistant Professor, Dept. of Computer Science and Engineering, RCC Institute of Information Technology) under whose scholarly guideline, constant encouragement & untiring patience; we have proud privilege to accomplish this entire project work. We feel enriched with the knowledge & sense of responsible approach we inherited from our guide & shall remain a treasure in our life.



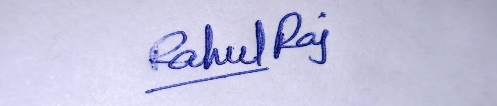
Name: Abhishek Agrahari

Roll No: 11700117141



Name: Subhadip Nandi

Roll No: 11700117036



Name: Rahul Raj

Roll No: 11700117075



Name: Rudrapratap Ghosh

Roll No: 11700117062

ABSTRACT

Remote sensing data processing deals with real-life applications with great societal values. For instance, urbanization monitoring, fire detection or flood prediction from remotely sensed images have great impact on economic and environmental issues.

Change detection based on remote sensing data is a very important method of detecting changes on the Earth’s surface and has a wide range of applications. It is a process of identifying changes of an object by using multi temporal RS data i.e. data from different time periods. Multi temporal RS data such as satellite imaginary provides a lot of information based on which changes occurred in an area or on the state of an object can be detected. For example by using satellite images of two time periods of an forest, the amount of deforestation can be estimated by detecting the changes occurred in the region. The project aims to apply change detection techniques for urbanization monitoring and/or agricultural land monitoring and/or deforestation.

CONTENTS

[**CERTIFICATEOFAPPROVAL I**](#_bookmark0)

[**ACKNOWLEDGEMENT II**](#_bookmark1)

[**ABSTRACT III**](#_bookmark2)

[**CONTENTS IV**](#_bookmark3)

[**LISTOFSYMBOLS V**](#_bookmark4)

[**LISTOFABBREVIATIONS VI**](#_bookmark5)

[**LISTOFFIGURES VII**](#_bookmark6)

[**LISTOFTABLES VIII**](#_bookmark7)

**CHAPTER-1**

**INTRODUCTION…………………………………………………………………….**

**LITERATURE REVIEW……………………………………………………………..**

**CHAPTER-2**

**WORK FLOW…………………………………………………**

**METHODOLOGIES OF IMPLEMENTATION………………………………….**

**SOFTWARE & HARDWARE REQUIREMENTS………………………………….**

**OUR APPROACH……………………………………………..**

**IMPLEMENTATION……………………………………**

**CHAPTER-3**

**CONCLUSION……………………………………………**

**FUTURE SCOPE………………………………………………………..**

**REFERENCES………………………………………………………………………….**

LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form / Meaning** |
| CD | Change detection |

LIST OF FIGURES

|  |  |
| --- | --- |
| Fig 1.1 | An example of semantic segmentation, where the goal is to predict class labels for each pixel in the image |
| Fig 1.2 | Binary change detection |
| Fig 1.3 | Multiclass change detection |
| Fig 1.4 | Change detection in long time series of images |
| Fig 2 | General flow of AI-based change detection |
| Fig 3 | dataset of different location of the world |
| Fig 4 | Unusable portions of the image |
| Fig 5 | building address changes |
| Fig 6 | building mask of an Image. |
| Fig 7 | General schematic diagram of change detection techniques |
| Fig 8 | Schematic diagram of single-stream framework structures of AI-based change detection |
| Fig 9 | Schematic diagram of double-stream framework structure of AI-based change detection |
| Fig 10 | Schematic diagram of multi-model integrated structure of AI-based change detection. |
| Fig 11 | Semantic segmentation using FCN |
| Fig 12 | Basic Unet model |
| Fig 13 | Mask RCNN Model |
| Fig 14 | post-classification structure |
| Fig 15 | Unet model structure |
| Fig 16 | Convolution Operation |
| Fig 17 | Max pooling operation |
| Fig 18 | Encoder and Decoder path of Unet model |
| Fig 19 | Input image, True Mask, Predicted Mask of basic convolution model |
| Fig 20 | Confusion matrix of basic convolution model |
| Fig 21 | Model accuracy of basic convolution model |
| Fig 22 | Model loss of basic convolution model |
| Fig 23 | Input image, True Mask, Predicted Mask of transpose convolution model |
| Fig 24 | Confusion matrix of transpose convolution model |
| Fig 25 | Model accuracy of transpose convolution model |
| Fig 26 | Model loss of transpose convolution model |

|  |  |
| --- | --- |
| Fig 27 | Input image and predicted mask location on 2018 |
| Fig 28 | Input image and predicted mask location on 2020 |
| Fig 29 | Changed map of different time period |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table 1.1 | Literature Review |  |

# **CHAPTER-1**

**1.1 Introduction**

**1.1.1. Remote Sensing**

## Remote sensing is the science of obtaining information about objects or areas from a distance, typically from aircraft or satellites. Remote sensors collect data by detecting the energy that is reflected from Earth. These sensors can be on satellites or mounted on aircraft. Remote sensors can be either passive or active. Passive sensors respond to external stimuli. They record natural energy that is reflected or emitted from the Earth's surface. The most common source of radiation detected by passive sensors is reflected sunlight.

In contrast, active sensors use internal stimuli to collect data about Earth. For example, a laser-beam remote sensing system projects a laser onto the surface of Earth and measures the time that it takes for the laser to reflect back to its sensor.

**1.1.2. Semantic Segmentation**

The goal of semantic image segmentation is to label each pixel of an image with a corresponding class of what is being represented. Because we're predicting for every pixel in the image, this task is commonly referred to as dense prediction.

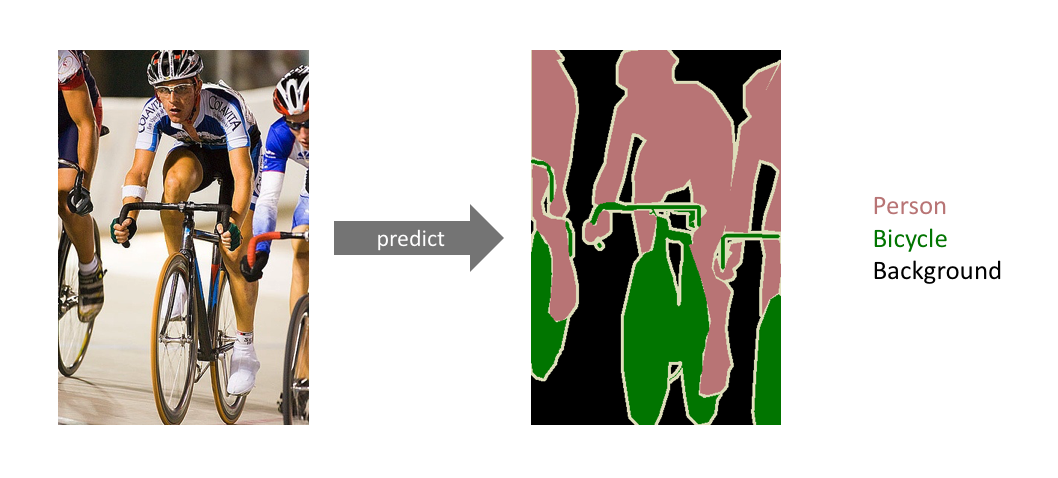


Figure 1.1: An example of semantic segmentation, where the goal is to predict class labels for each pixel in the image.

One important thing to note is that we're not separating instances of the same class; we only care about the category of each pixel. In other words, if you have two objects of the same category in your input image, the segmentation map does not inherently distinguish these as separate objects. There exists a different class of models, known as instance segmentation models, which do distinguish between separate objects of the same class.

**1.1.3. Change Detection**

Change detection is the process that analyzes multi temporal remote sensing images acquired on the same geographical area for identifying changes occurred between the considered acquisition dates.

We can define different change detection problems:

* Binary change detection.
* Multiclass change detection.
* Change detection in long time series of images.

1. Binary change detection

* **Goal:** production of binary maps in which changed and unchanged areas are separated.
* **Number of images:** 2 (or pairs of images extracted from a series).
* **Application domain:** detection of abrupt (step) changes.

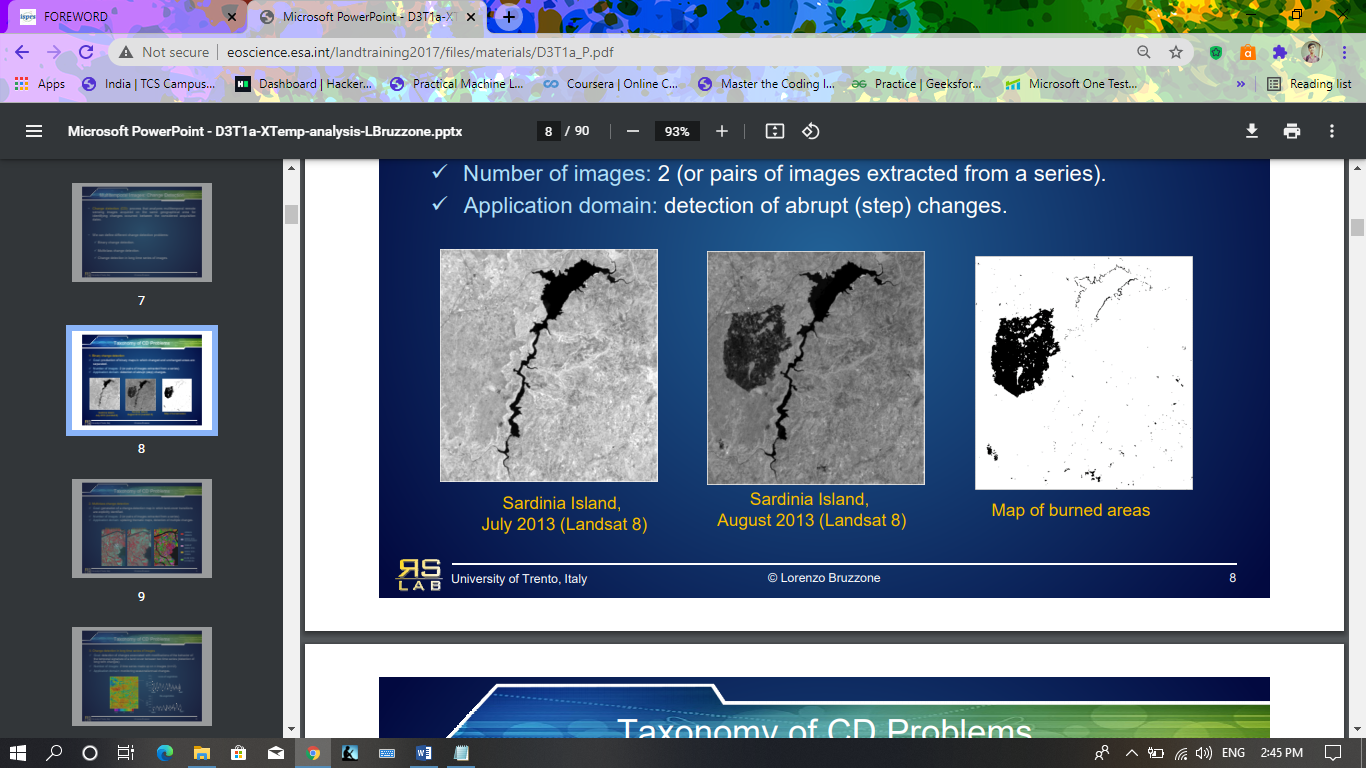


Figure 1.2 Binary change detection

2. Multiclass change detection

* **Goal:** Generation of a change-detection map in which land-cover transitions are explicitly identified.
* **Number of images:** 2 (or pairs of images extracted from a series).
* **Application domain:** updating thematic maps, detection of multiple changes.



Figure 1.3 Multiclass change detection

3. Change detection in long time series of images

* **Goal:** detection of changes associated with modifications of the behavior of the temporal signature of a land cover between two time series (detection of long term changes).
* **Number of images:** 2 time series made up on n images (n>>2).
* **Application domain:** monitoring seasonal/annual changes.

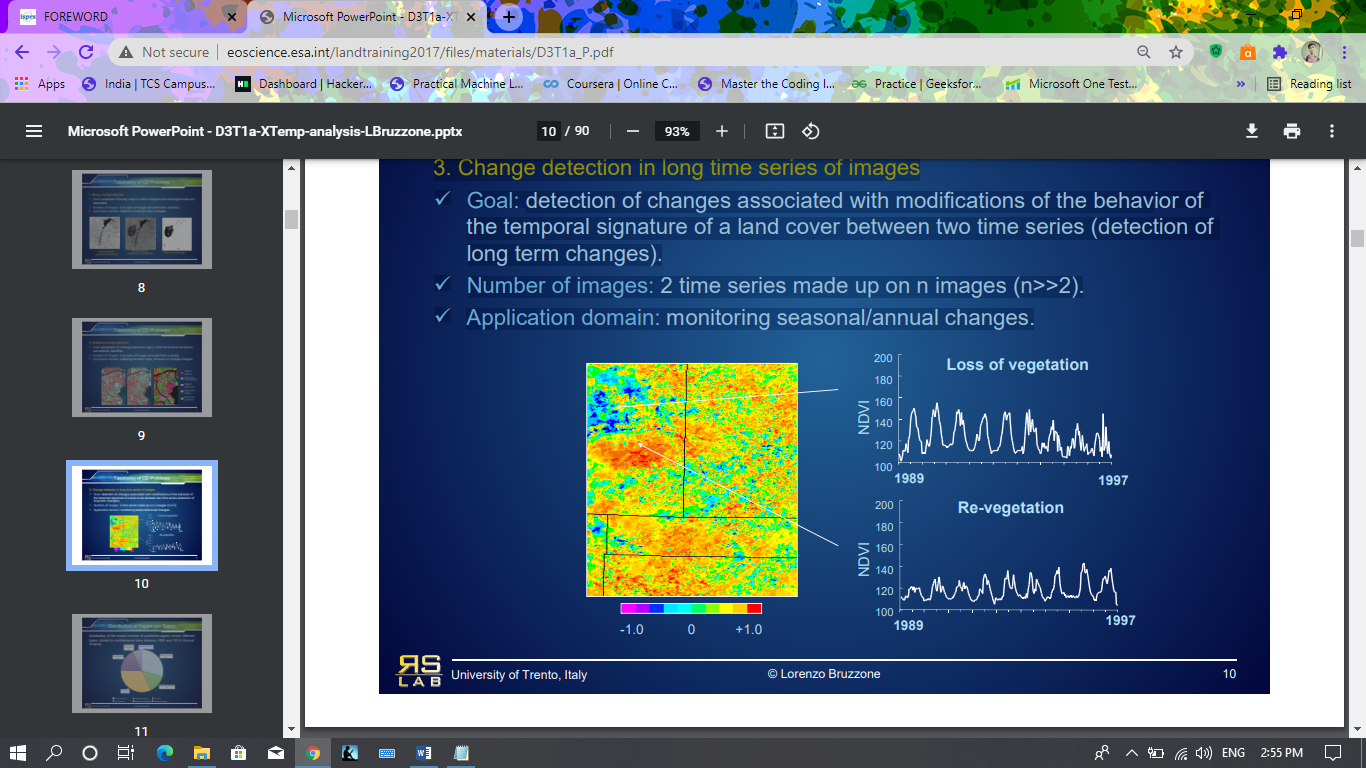


Figure 1.4 Change detection in long time series of images

**1.1.4. Urban Development Change Detection**

Satellite imagery analytics have numerous human development and disaster response applications, particularly when time series methods are involved. For example, quantifying population statistics is fundamental to 67 of the 232 United Nations Sustainable Development Goals, but the World Bank estimates that more than 100 countries currently lack effective Civil Registration systems. The project aims to identify and track buildings in satellite imagery time series collected over rapidly urbanizing areas. Then building construction over time is tracked, thereby directly assessing urbanization.

**1.2 LITERATURE REVIEW**

Several reviews of CD methods have discussed diverse aspects including approaches, algorithms, and applications. Singh [[**1**](https://www.mdpi.com/2072-4292/12/11/1781/htm#B1-remotesensing-12-01781)] and Nelson [**2**] classified CD techniques into

(1) Comparative analyses based on post-classified data and

(2) Simultaneous analyses of multitemporal images.

**Table 1.1**: Literature Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No** | **Author Name** | **Paper Name** | **Contribution** | **Challenges** |
| 1. | D.A Mouat,.; G.G Mahin,.;  J. Lancaster. | Remote sensing techniques in the analysis of change detection | focused on the use of CD to investigate ecosystem change by employing several sensors, including Advanced Very High Resolution Radiometer (AVHRR), Landsat Thematic Mapper (TM) and Multispectral Scanner (MSS), *Satellite pour l’Observation de la Terre* (SPOT), and aerial photographs.[3] | None of the reviews discussed CD from the perspective of users wishing to have an overview of possible data and suitable techniques for generating required change information. |
| 2. | P. Coppin;  I. Jonckheere.;  K. Nackaerts;  B. Muys,; E.Lambin. | Review Article Digital change detection methods in ecosystem monitoring: a review | Comprehensively discussed CD from three viewpoints, i.e., ecosystem properties, preprocessing procedures, and CD techniques.[4] |
| 3. | R.J. Radke,  S. Andra;  O. Al-Kofahi;  B. Roysam. | Image change detection algorithms: a systematic survey | Focused on CD algorithms. [**5**] |
| 4. | D. Lu;  P. Mausel;  E. Brondízio;  E. Moran. | Change detection techniques | grouped ecosystem explorations into ten classes, including general or specific aspects of land use, environment, forest, urban, and agricultural topics.[6] |

# **CHAPTER-2**

**2.1 WORK FLOW**

The implementation process of AI-based change detection includes the following four main steps:

1. **Homogenization:** Due to differences in illumination and atmospheric conditions, seasons, and sensor attitudes at the time of acquisition, multi-period data usually need to be homogenized before change detection. The former aims to geometrically align two or more given pieces of data, which can be achieved through registration or co-registration. Given two period data, only when they are overlaid can the comparison between corresponding positions be meaningful. The latter aims to eliminate radiance or reflectance differences caused by the digitalization process of sensors and atmospheric attenuation distortion caused by absorption and scattering in the atmosphere , which helps to reduce false alarms caused by these radiation errors in change detection. For heterogeneous data, a special AI model structure can be designed for feature space transformation to achieve change detection.
2. **Training set generation:** To develop the AI model, a large, high-quality training set is required that can help algorithms to understand that certain patterns or series of outcomes come with a given question. Multi-period data are labeled or annotated using certain techniques (e.g., manual annotation , pre-classification , use of thematic data ) to make it easy for the AI model to learn the characteristics of the changed objects. Figure 2 presents an annotated example for building change detection, which is composed of two-period RS images and a corresponding ground truth labeled with building changes at the pixel level. Based on the ground truth, i.e., prior knowledge, the AI model can be trained in a supervised manner.
3. **Model training:** After the training set is generated, it can usually be divided into two datasets according to the number of samples or the geographic area: a training set for AI model training and a test set for accuracy evaluation during the training process . The training and testing processes are performed alternately and iteratively. During the training process, the model is optimized according to a learning criterion, which can be a loss function in deep learning (e.g., softmaxloss , contrastive loss , Euclidean loss , or cross-entropy loss ). By monitoring the training process and test accuracy, the convergence state of the AI model can be obtained, which can help in adjusting its hyperparameters (such as the learning rate), and also in judging whether the model performance has reached the best (i.e., termination) condition.
4. **Model serving:** By deploying a trained AI model, change maps can be generated more intelligently and automatically for practical applications. Moreover, this can help validate the generalization ability and robustness of the model, which is an important aspect of evaluating the practicality of the AI-based change detection technique.

The above steps provide a general implementation process of AI-based change detection, but the structure of the AI model is diverse and needs to be well designed according to different application situations and the training data, which will be introduced in Sections 4 and 5. It is worth mentioning that existing mature frameworks such as TensorFlow ,Keras , Pytorch , and Caffe, help researchers more easily realize the design, training, and deployment of AI models, and their development documents provide detailed introductions.

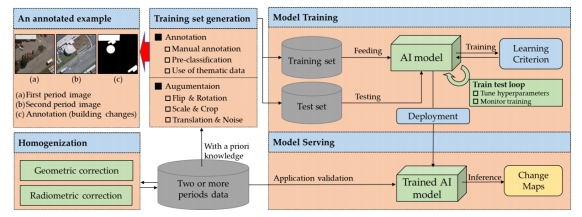
****

Figure 2 General flow of AI-based change detection

**2.1.1. Training dataset generation and Homogenization**

We are using the Spacenet 7 dataset.

This dataset consists of Planet satellite imagery mosaics, which includes 24 images (one per month) covering ~100 unique geographies. The dataset will comprise over 40,000 square kilometers of imagery and exhaustive polygon labels of building footprints in the imagery, totaling over 10 million individual annotations

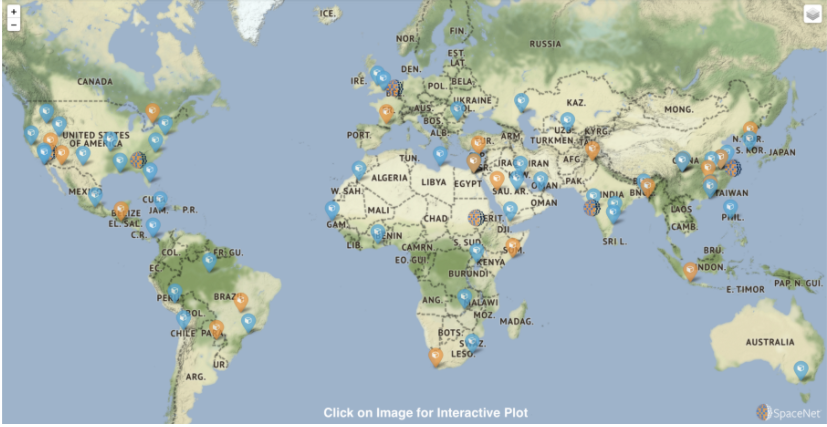
****

Figure 3 dataset of different location of the world

Imagery consists of RBGA (red, green, blue, alpha) 8-bit electro-optical (EO) monthly mosaics from Planet’s Dove constellation at 4 meter resolution. For each of the Areas Of Interest (AOIs), the data cube extends for roughly two years, though it varies somewhat between AOIs. All images in a data cube are the same shape, though some data cubes have shape 1024 x 1024 pixels, while others have a shape of 1024 x 1023 pixels. Each image accordingly has an extent of roughly 18 square kilometres.

Images are provided in GeoTiff format, and there are two imagery data types:

* images (training only) - Raw imagery
* images masked (training + testing) - Unusable portions of the image (usually due to cloud cover) have been masked out.

****

Figure 4 Unusable portions of the image

For each monthly mosaic there is a GEOJSON vector label present in the dataset.While building masks are useful for visualization (and for training deep learning segmentation algorithms) the precise vector labels of the SpaceNet 7 dataset permit the assignment of a unique identifier (i.e. address) to each building. Matching these building addresses between time steps is a central theme of the SpaceNet 7 challenge. The figure below displays these building address changes.

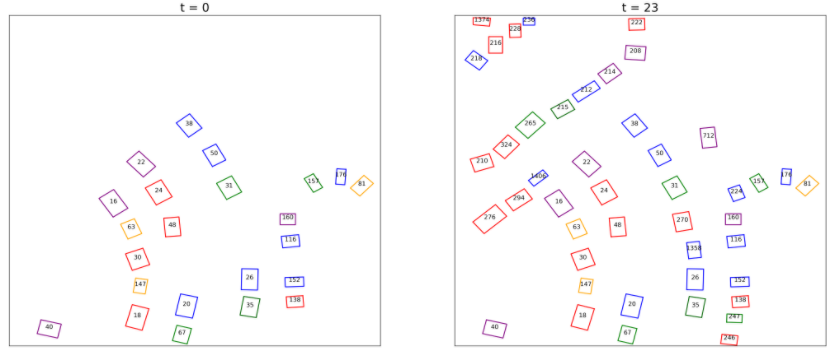
****

Figure 5 building address changes

**2.1.2. Model Creation and Training:**

Our dataset contains GEOtif images and the building labels in geojson format. At first we are creating the building masks from the geojson file for each image.

An example of a training data and its mask is shown bellow-

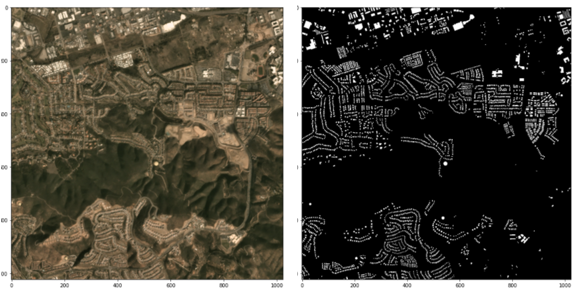


Figure 6: building mask of an Image.

We are creating a Neural Network model to do the semantic segmentation of our test dataset and predict the building mask for test images and then compare the pixel values of image from two different time period to generate the change map.

**2.2 METHODOLOGIES OF IMPLEMENTATION**

# **2.2.1. CHANGE DETECTION TECHNIQUES**

Change Detection Techniques are classified into two categories.

1. Traditional techniques.
2. AI based methods.

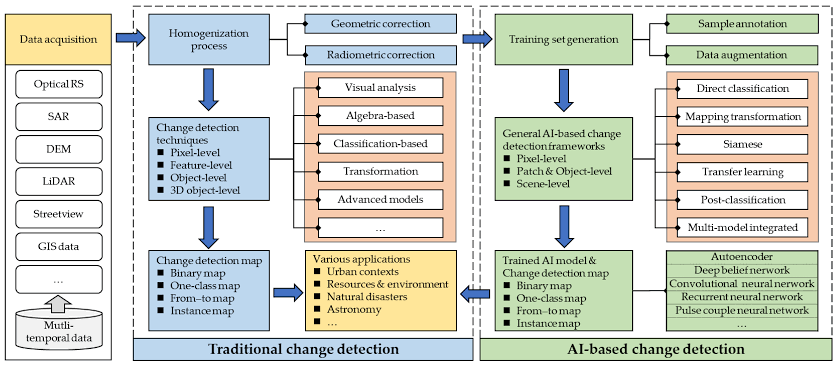


Figure 7. General schematic diagram of change detection techniques

The aim of change detection is to generate change maps for various applications. The first step of each approach is to prepare the data. The traditional approach uses two steps, one is homogenization process and the other is change detection process. In AI based approaches there is an extra step of training set generation and model training process.

Due to the ability of AI techniques to provide a better performance in various data processing tasks and the availability of advance classification methods, we aim to use AI techniques in our change detection process which is expected to give better results in comparison with the traditional techniques.

AI techniques are further classified into three types: single-stream, double-stream and multi-model integrated.

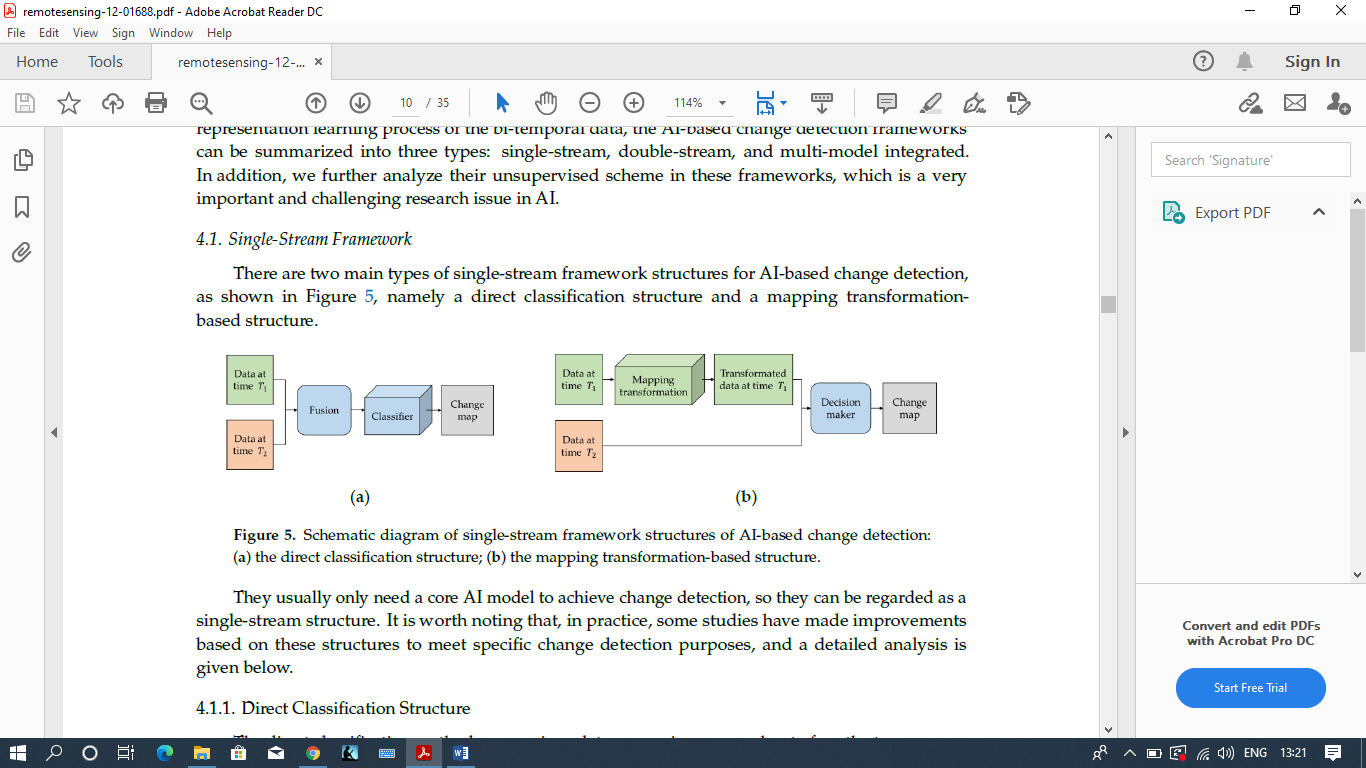


Figure 8. Schematic diagram of single-stream framework structures of AI-based change detection:(**a**) the direct classification structure; (**b**) the mapping transformation-based structure.

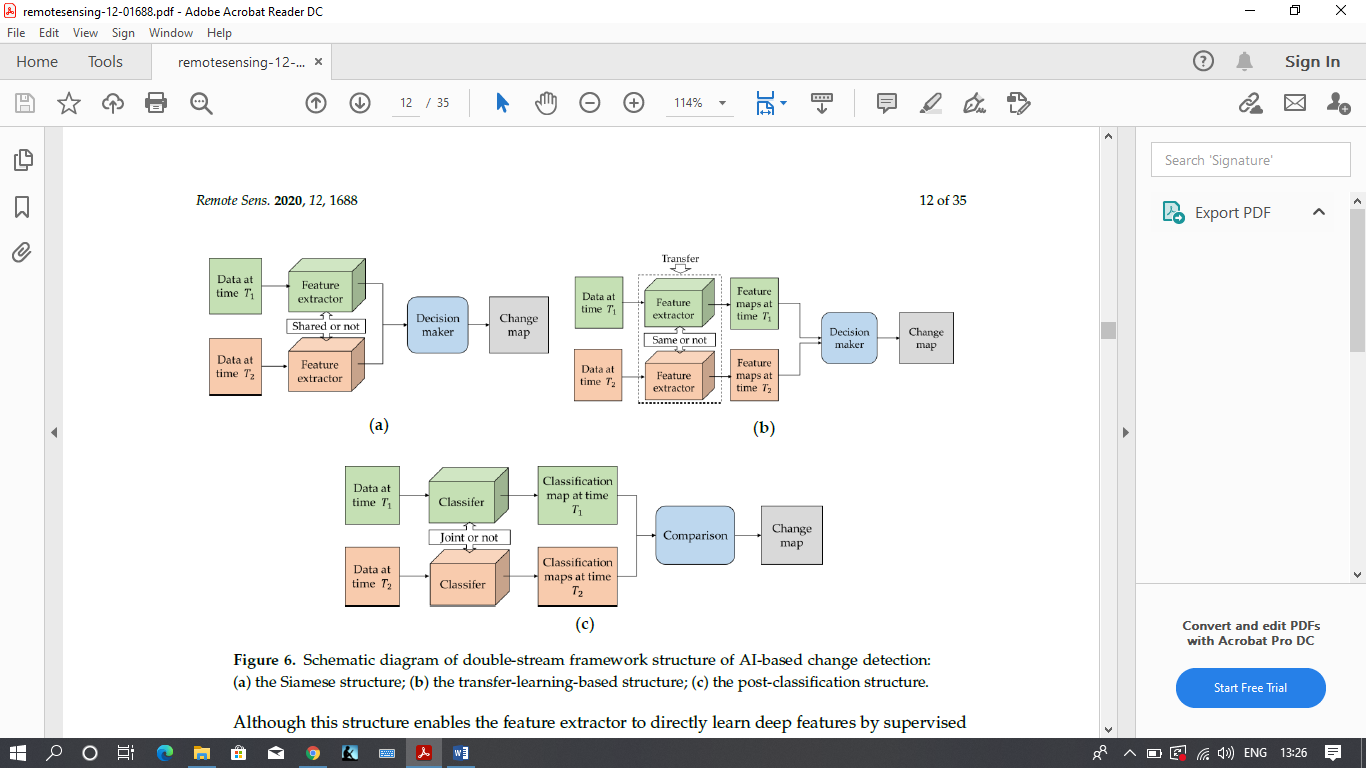


Figure 9. Schematic diagram of double-stream framework structure of AI-based change detection:(a) the Siamese structure; (b) the transfer-learning-based structure; (c) the post-classification structure.

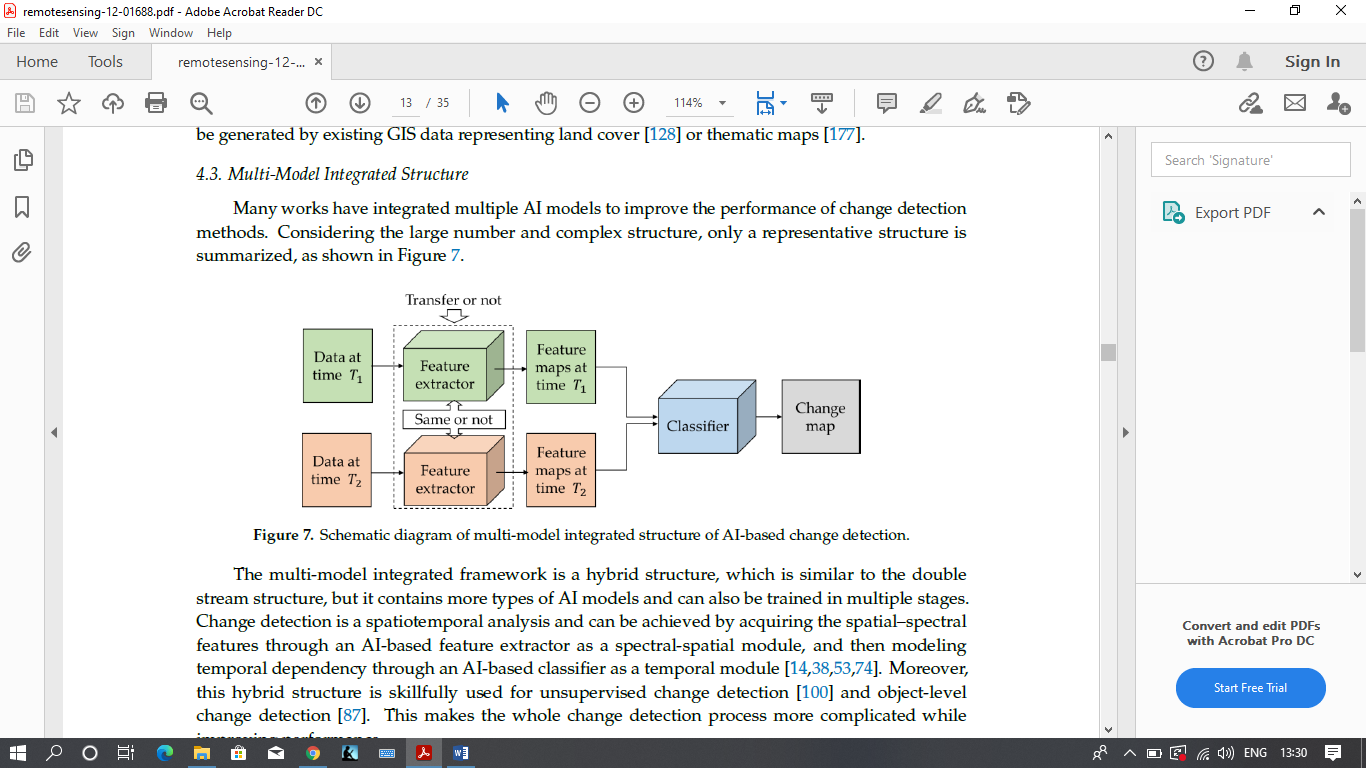


Figure 10. Schematic diagram of multi-model integrated structure of AI-based change detection.

There is also an unsupervised technique of change detection which is a challenging task and is a popular field of research.

**2.2.2. Deep Learning Model Architectures for Semantic Segmentation**

**2.2.2.1.**[**Fully Convolutional Network (FCN)**](https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf)

FCN is a popular algorithm for doing semantic segmentation. This model uses various blocks of convolution and max pool layers to first decompress an image to 1/32th of its original size. It then makes a class prediction at this level of granularity. Finally it uses up sampling and de convolution layers to resize the image to its original dimensions.

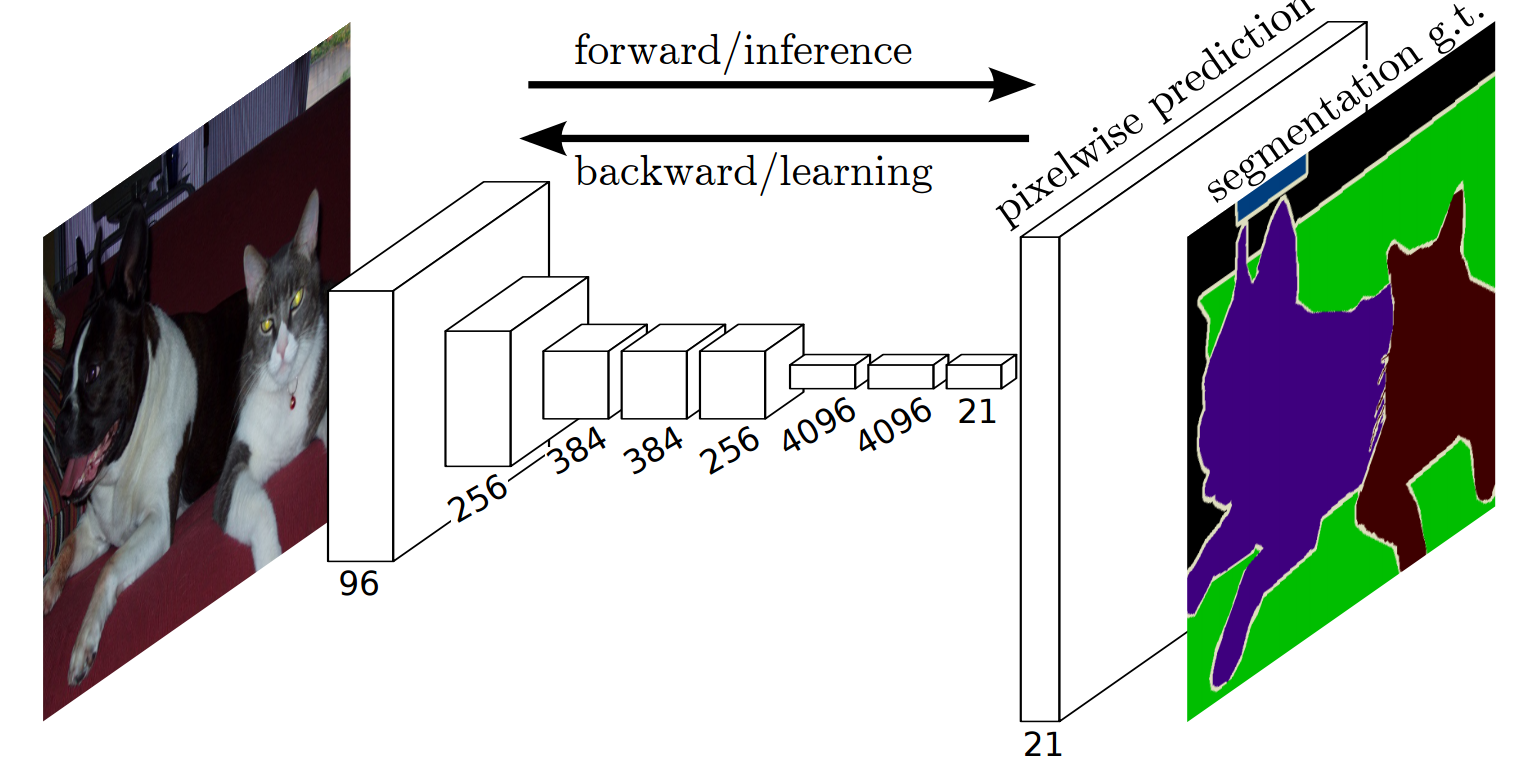


Figure 11. Semantic segmentation using FCN

These models typically don’t have any fully connected layers. The goal of down sampling steps is to capture semantic/contextual information while the goal of up sampling is to recover spatial information. Also there are no limitations on image size. The final image is the same size as the original image. To fully recover the fine grained spatial information lost in down sampling, skip connections are used. A skip connection is a connection that bypasses at least one layer. Here it is used to pass information from the down sampling step to the up sampling step. Merging features from various resolution levels helps combining context information with spatial information.

**2.2.2.2. U-Net**

The [U-Net](https://arxiv.org/abs/1505.04597)architecture is built upon the Fully Convolutional Network (FCN) and modified in a way that it yields better segmentation in medical imaging.

Compared to FCN-8, the two main differences are:

1. U-net is symmetric and
2. the skip connections between the downsampling path and the upsampling path apply a concatenation operator instead of a sum.

These skip connections intend to provide local information to the global information while upsampling. Because of its symmetry, the network has a large number of feature maps in the upsampling path, which allows to transfer information. B

The U-Net owes its name to its symmetric shape, which is different from other FCN variants.

U-Net architecture is separated in 3 parts:

1 : The contracting/downsampling path  
2 : Bottleneck  
3 : The expanding/upsampling path

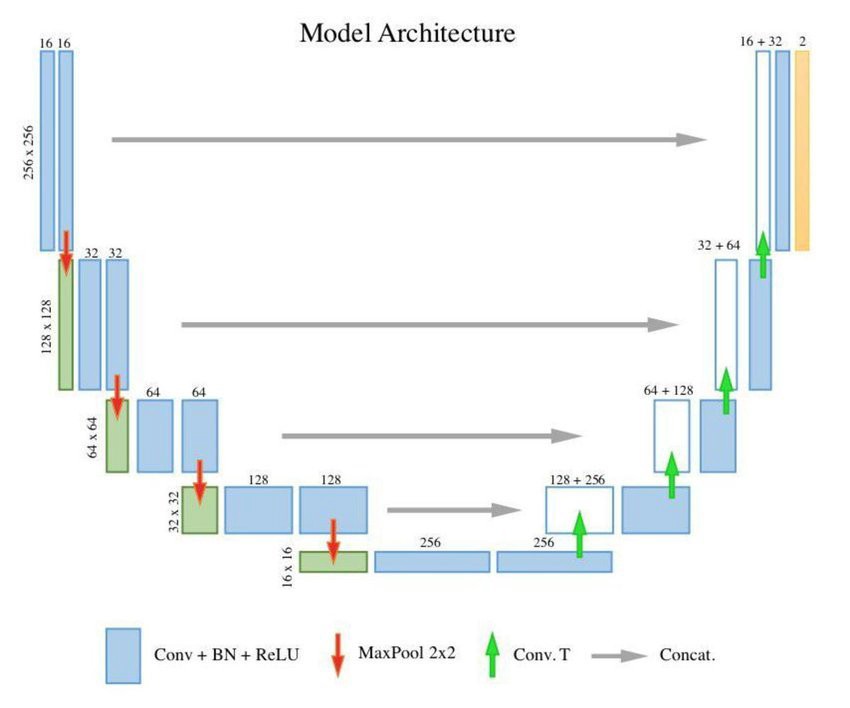


Figure 12. Basic Unet model

**2.2.2.3. Mask RCNN**



Figure 13.Mask RCNN Model

[Faster RCNN](https://arxiv.org/abs/1506.01497) is a very good algorithm that is used for object detection. Faster R-CNN consists of two stages. The first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes. The second stage, which is in essence Fast R-CNN, extracts features using RoIPool from each candidate box and performs classification and bounding-box regression. The features used by both stages can be shared for faster inference.

Mask R-CNN is conceptually simple: Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this we add a third branch that outputs the object mask — which is a binary mask that indicates the pixels where the object is in the bounding box. But the additional mask output is distinct from the class and box outputs, requiring extraction of much finer spatial layout of an object. To do this Mask RCNN uses the Fully Convolution Network (FCN).

**2.3 SOFTWARE & HARDWARE REQUIREMENTS**

We use Keras callbacks to implement:

* Learning rate decay if the validation loss does not improve for 5 continues epochs.
* Early stopping if the validation loss does not improve for 10 continues epochs.
* Save the weights only if there is improvement in validation loss.

There could be a lot of scope to tune these hyper parameters and further improve the model performance.

The model is trained on GPU google colab.

**2.4 OUR APPROACH**

We are using the **Post-Classification** method in our application of change detection as it is regarded as a very general and practical structure and it provides type change matrix and it works well for data acquired under different acquisition conditions and even different sensors.

For classifications we are building a semantic segmentation model using UNet.

The detail of our approach is discussed below.

* + 1. **Post-Classification Structure**

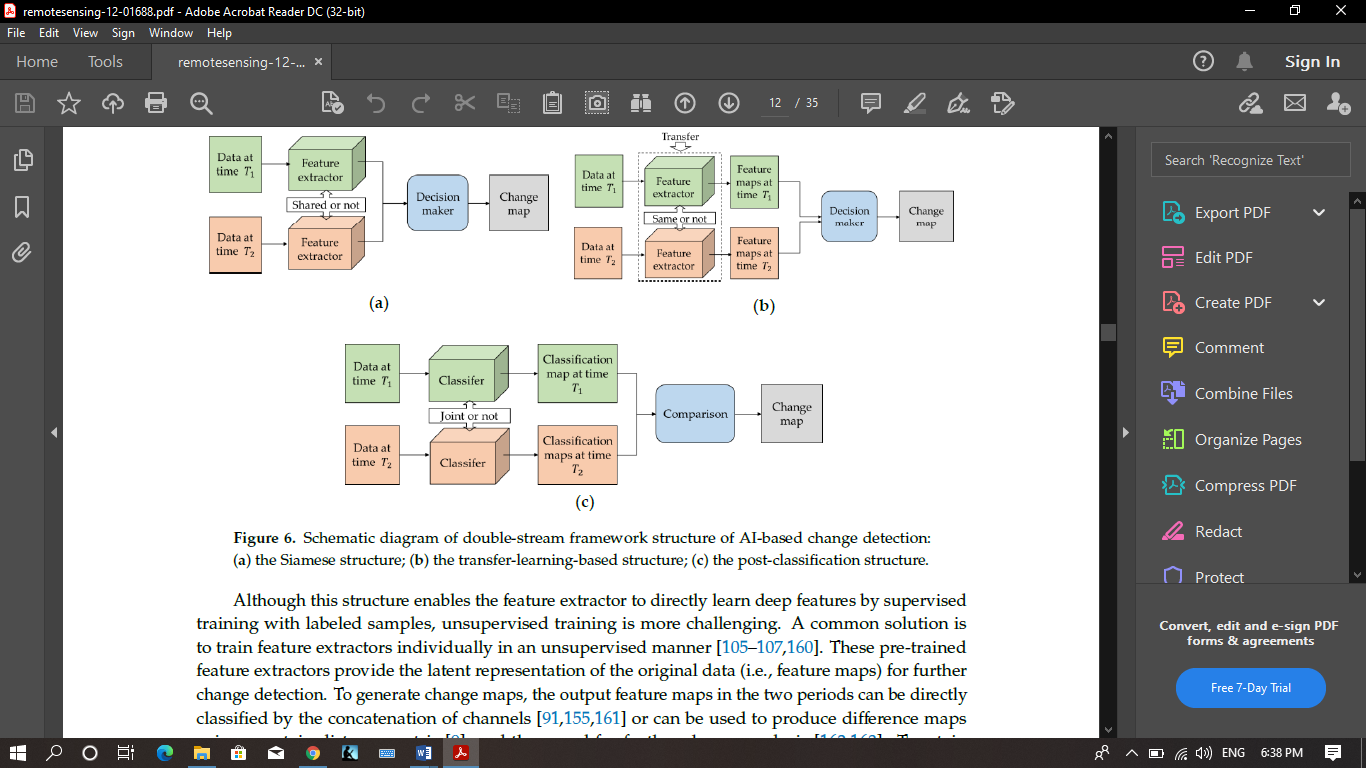
****

Figure 14.post-classification structure

As shown in the Figure above, the post-classification structure consists of two classifiers, which can usually be converted into classification tasks and trained in a joint or independent way. It provides a classification map for each period data and the change map with change directions can be obtained by comparing classification maps. Nevertheless, the accuracy of the change detection results of these methods depends on the performance of the classifier.

* + 1. **Classification/Semantic Segmentation using UNET**

The UNET is described as follows:

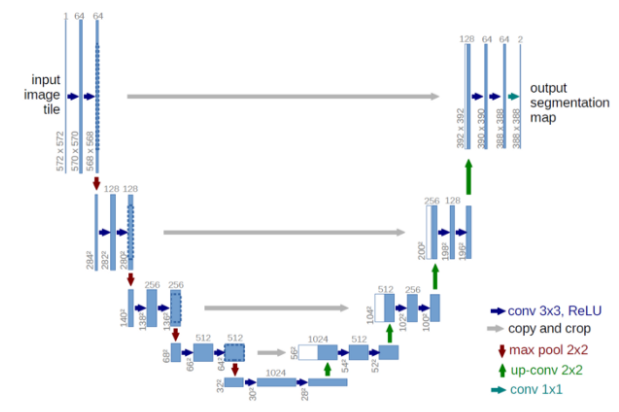
****

Figure 15.Unet model structure

* + 1. **Convolution Operation:**

There are two inputs to a convolutional operation

1. A 3D volume (input image) of size (nin x nin x channels)
2. A set of ‘k’ filters (also called as kernels or feature extractors) each one of size (f x f x channels), where f is typically 3 or 5.

The output of a convolutional operation is also a 3D volume (also called as output image or feature map) of size (nout x nout x k)

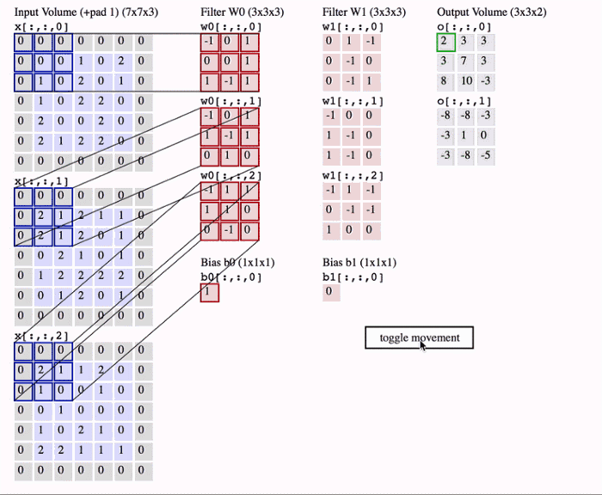
****

Figure 16.Convolution Operation

We have an input volume of size 7x7x3. Two filters each of size 3x3x3. Padding =0 and Strides = 2. Hence the output volume is 3x3x2.

## **Max pooling operation:**

In simple words, the function of pooling is to reduce the size of the feature map so that we have fewer parameters in the network.

Basically from every 2x2 block of the input feature map, we select the maximum pixel value and thus obtain a pooled feature map. Note that the size of the filter and strides are two important hyper-parameters in the max pooling operation.

The idea is to retain only the important features (max valued pixels) from each region and throw away the information which is not important. By important, I mean that information which best describes the context of the image.

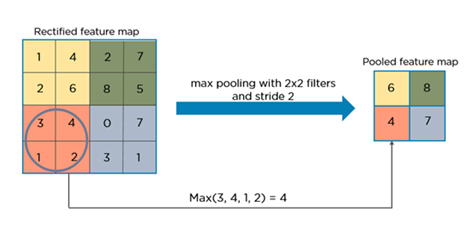
****

Figure 17.Max pooling operation

## **Need for up sampling:**

The output of semantic segmentation is not just a class label or some bounding box parameters. In-fact the output is a complete high resolution image in which all the pixels are classified

## **Transposed Convolution:**

Transposed convolution (sometimes also called as deconvolution or fractionally strided convolution) is a technique to perform up sampling of an image with learnable parameters.

It is nicely explained how a normal convolution can be expressed as a matrix multiplication of input image and filter to produce the output image. By just taking the transpose of the filter matrix, we can reverse the convolution process, hence the name transposed convolution.

Below is the detailed explanation of the architecture:

Fig 15 has the size of the input image is 572x572x3, however, we will use input image of size 128x128x3. Hence the size at various locations will differ from that in the original paper but the core components remain the same.

## **Points to note:**

* 2@Conv layers means that two consecutive Convolution Layers are applied
* c1, c2, …. c9 are the output tensors of Convolutional Layers
* p1, p2, p3 and p4 are the output tensors of Max Pooling Layers
* u6, u7, u8 and u9 are the output tensors of up-sampling (transposed convolutional) layers
* The left hand side is the contraction path (Encoder) where we apply regular convolutions and max pooling layers.
* In the Encoder, the size of the image gradually reduces while the depth gradually increases. Starting from 128x128x3 to 8x8x256
* This basically means the network learns the “WHAT” information in the image, however it has lost the “WHERE” information
* The right hand side is the expansion path (Decoder) where we apply transposed convolutions along with regular convolutions
* In the decoder, the size of the image gradually increases and the depth gradually decreases. Starting from 8x8x256 to 128x128x1
* Intuitively, the Decoder recovers the “WHERE” information (precise localization) by gradually applying up-sampling
* To get better precise locations, at every step of the decoder we use skip connections by concatenating the output of the transposed convolution layers with the feature maps from the Encoder at the same level:

u6 = u6 + c4  
u7 = u7 + c3  
u8 = u8 + c2  
u9 = u9 + c1

After every concatenation we again apply two consecutive regular convolutions so that the model can learn to assemble a more precise output

* This is what gives the architecture a symmetric U-shape, hence the name UNET
* On a high level, we have the following relationship:  
  Input (128x128x1) => Encoder =>(8x8x256) => Decoder =>Ouput (128x128x1)

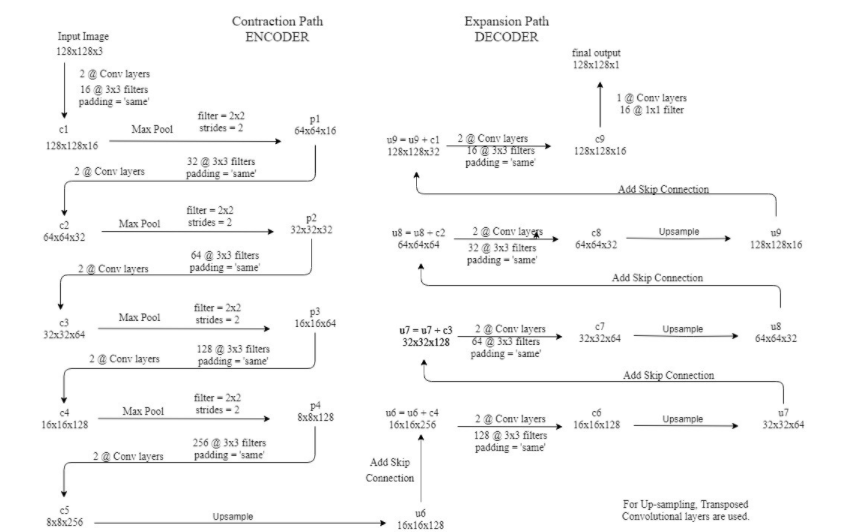
****

Figure 18.Encoder and Decoder path of Unet model

* 1. **IMPLEMENTATION**
     1. **About the Training and Validation split and Test split**

We have to total 56 images of randomly chosen different locations . For training we have used first 50 images in the training dataset and for validation 6 images are used in validation dataset. We use only one image or testing

**Training Dataset**: The sample of data used to fit the model.

**Validation Dataset**: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration.

**Test Dataset**: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

* + 1. **Data Processing during training**
       1. **Patch Generator:**

To have sufficient training data from the given high definition images cropping is required to train the classifier which are parameters of our U-Net implementation. The crop size of 160 x 160 we find under-representation of the binary classes.

Using a cropping window of 160x160 pixels with a stride of 160 resultant of 80% training and 20% validation images.

* + - 1. **Image Dimensions:**

Before cropping, the dimensions of training images are converted into multiples of stride for convenience during strided cropping.

For the cases where the no. of crops is not the multiple of the image dimensions we initially tried zero padding , we realised that adding padding will add unwanted artefacts in the form of black pixels in training and test images leading to training on false data and image boundary.

Alternatively we have correctly changed the image dimensions by adding extra pixels in the right most side and bottom of the image. So we padded the difference from the left most part of the image to it’s right deficit end and similarly for the top and bottom of the image.

* + 1. **Model:**
       1. **Unet model with transpose convolution:**

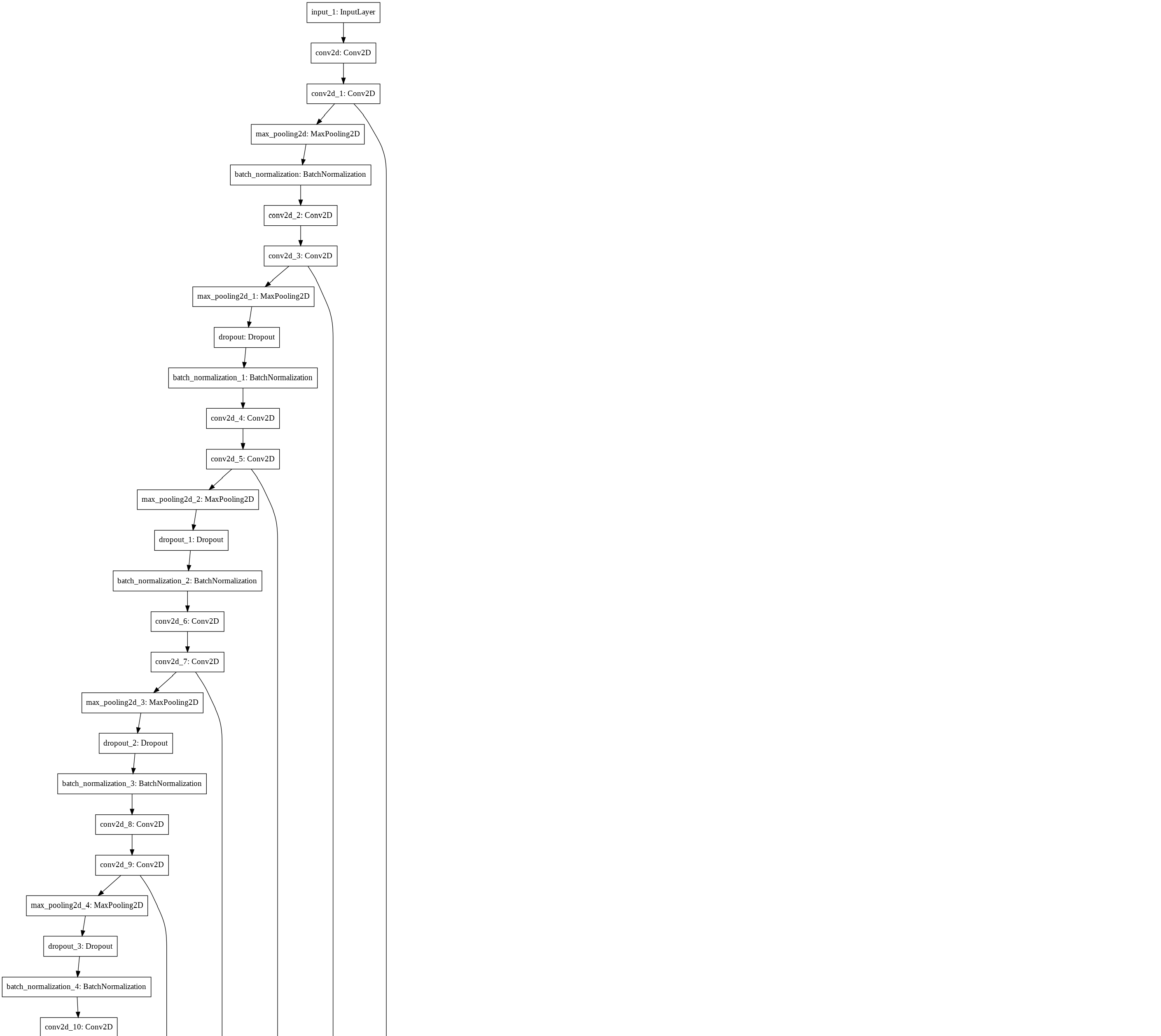
We have plotted the model architecture . The plot image is given below in two parts.

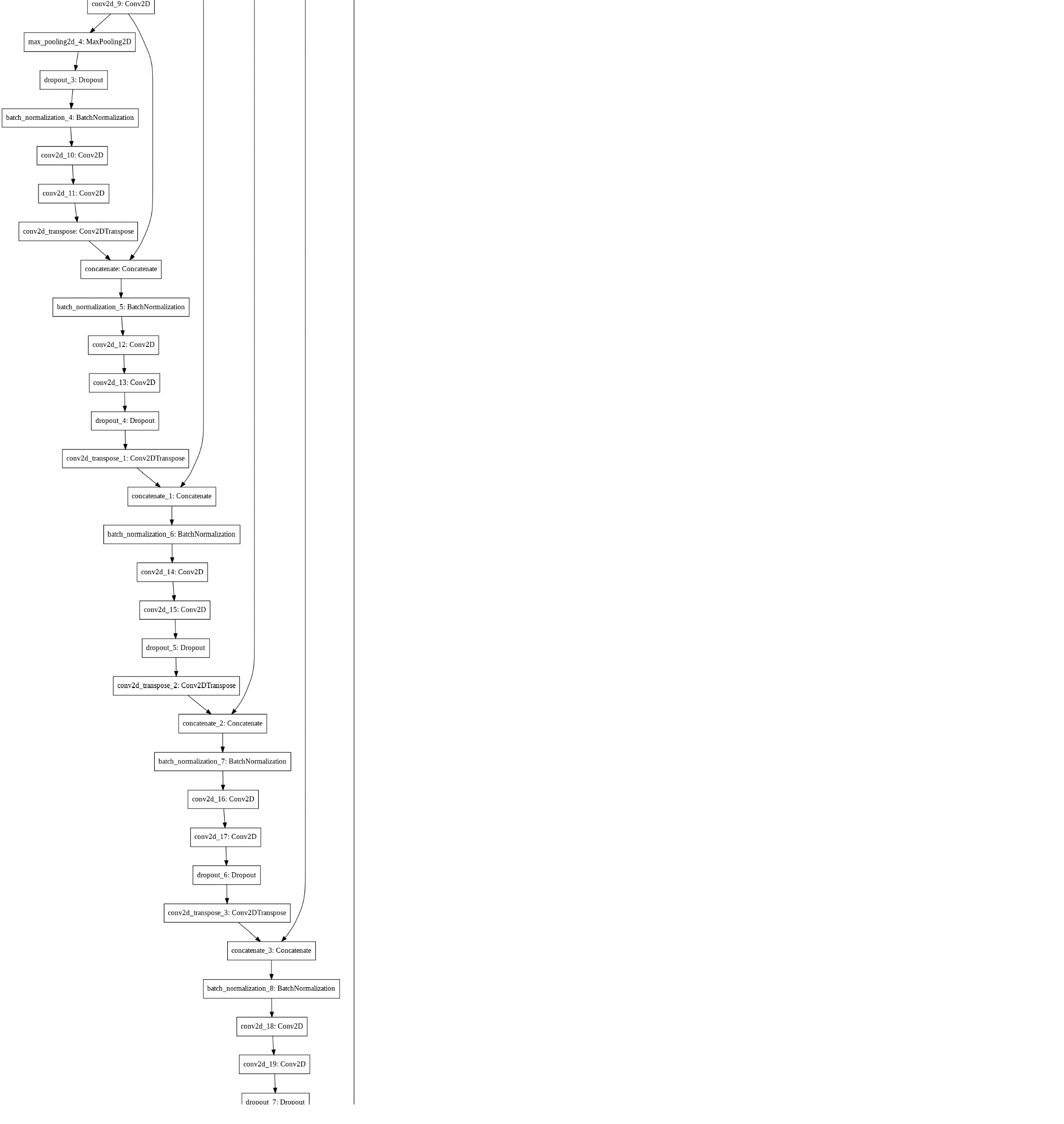
1st part describes Encoding .

2nd part describe Decoding.

Input of the model is 160x160x3 images and label sixe is 160x160x1.

The model is binary classification model.





Model summary:



**Trainable parameters** are the number of, well, trainable elements in your network; neurons that are affected by back propagation. For example, for the Wx + b operation in each neuron, W and b are trainable – because they are changed by optimizers after back propagation was applied for gradient computation.

In keras, **non-trainable parameters** ,means the number of weights that are not updated during training with back propagation.

There are mainly two types of non-trainable weights:

* The ones that you have chosen to keep constant when training. This means that keras won't update these weights during training at all.
* The ones that work like statistics in Batch Normalization layers. They're updated with mean and variance, but they're not "trained with back propagation".

Performance and Accuracy:

We have train with 56 location and from each location we have used 10+ images.

Patches ware generated from the images using the process described in **2.5.2.1 .**

**Patch size =** 160 x 160 x3

**Stride =**160

**Training set =** 11052 patches

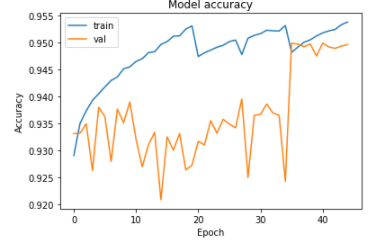
**Validation set=** 3060 patches

**Batch size =** 64

**Epochs =** 45

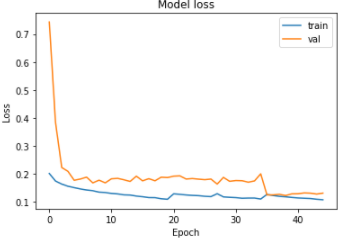
**Filter size=** 64

Accuracy vs epoch :

****

So, In this model validation accuracy not follow training accuracy, so model is over-fitting model.

Loass vs epoch:

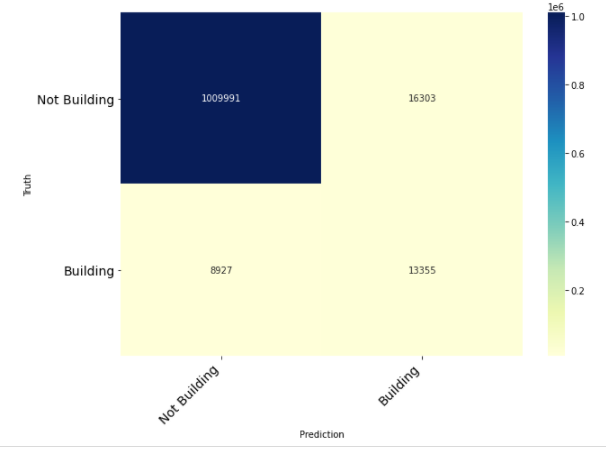
****

**Training accuracy:** 0.9538

**Validation accuracy:** 0.9496

**Validation kappa score :** 0.5250

Confusion matrix:



**Test accuracy:** 0.98

**Test kappa score :** 0.5021

**Output:**



Figure 23 Input image, True Mask, Predicted Mask of transpose convolution model

**Other things we have tried:**

We have changed training and validation set images , like we have assigned 70% of location in training set and 30% of location in validation set.

And we have changed karnel initializer, like Random\_normal , uniform\_normal, random\_uniorm.

But did not find major improvement.

* + 1. **Result**
       1. **Test image and prediction**

**1**. Location on 2018

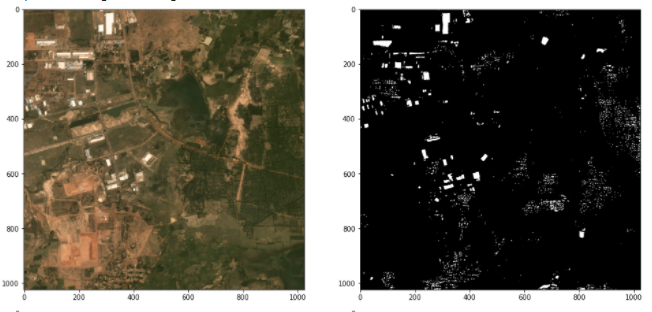


Figure 27 Input image and predicted mask location on 2018

**2.** Location on 2020

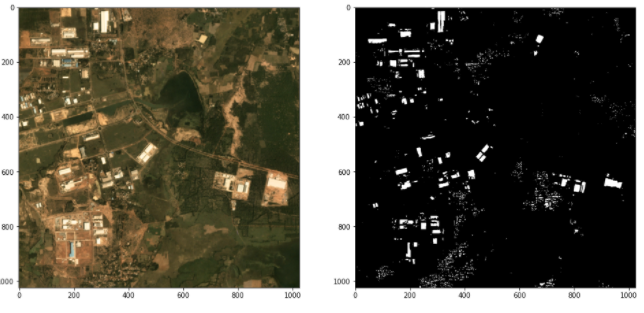


Figure 28 Input image and predicted mask location on 2020

* + 1. **Generating Change Map**

The change map is generated by comparing the pixel values of the classified images of the images from different time period.

The working of the project is described in the picture below with an example.

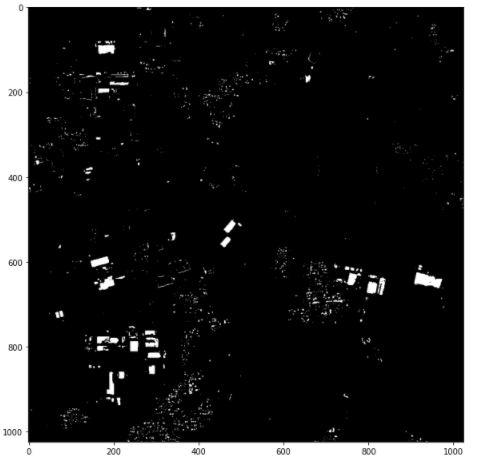


Figure 29.Changed map of different time period

**CHAPTER-3**

3.1 CONCLUSION

In this study, the techniques for detecting and measuring changes in urban development have been discussed in detail. Among various techniques we are using a post classification approach of change detection. For that purpose the Spacenet 7 dataset is used. A semantic segmentation model(U-Net) is considered to be used for classification purposes. The complete procedure of building the model and performing the change detection has been discussed in detail with examples. The testing and evaluation of our model is under progress. Along with that we are doing our research to improve the performance of our approach.

3.2 FUTURE SCOPE

Till now we are done with our model selection for this change detection. and detailed study of Unet model. We have created labeled mask of different time series data of Spacenet 7 dataset. Our next step is to feed the image and label mask into the model and train it and classify image pixel by pixel and compare with mask. Our main intention to create a optimize loss function and predict the mask for test image .then we have to detect the changes between different time series data.

References

1. Singh, A. Review Article: Digital Change Detection Techniques Using Remotely-Sensed Data. *Int. J. Remote Sens.* **1989**, *10*, 989–1003.
2. Nelson, R.F. Detecting Forest Canopy Change Due to Insect Activity Using Landsat Mss. *Photogramm. Eng. Remote Sens.* **1983**, *49*, 1303–1314.
3. Mouat, D.A.; Mahin, G.G.; Lancaster, J. Remote sensing techniques in the analysis of change detection. *Geocarto Int.* **1993**, *8*, 39–50.
4. Coppin, P.; Jonckheere, I.; Nackaerts, K.; Muys, B.; Lambin, E. Digital change detection methods in ecosystem monitoring: A review. *Int. J. Remote Sens.* **2004**, *25*, 1565–1596.
5. Radke, R.J.; Andra, S.; Al-Kofahi, O.; Roysam, B. Image change detection algorithms: A systematic survey. *IEEE Trans. Image Process.* **2005**, *14*, 294–307.
6. Lu, D.; Mausel, P.; Brondízio, E.; Moran, E. Change detection techniques. *Int. J. Remote Sens.* **2004**, *25*, 2365–2401.
7. About Unet model (<https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>)