Chapter 3
Methodology

3.1 Introduction

In recent eras, the consumption of land resources has become a serious problem. Remote sensing (RS) is the art of finding and understanding data from a long distance, using sensors without communication with the object being observed (Goslee, 2020). Land Use land cover classification aims to organize space-born images into an exact class, which were reliant on the distribution of recognized land use land cover classes. In the last few years variety of applications like urban development, monitoring of natural tragedies, and land use land cover analysis (Cheng et al., 2016) (Gmez-Chova et al., 2015) (Zhang et al., 2016) (Jaiswal et al., 1999). The remote Sensing image consists of residential areas, agricultural land, uncultivated land, water bodies, and other open areas.

Deep learning (DL) has seen a growing trend over the earlier span due to its commanding capability to represent learning. Deep learning permits models that are composed, based on multiple layers, to learn illustrations of data samples with several varieties of abstraction (Yann et al., 2015). DL algorithms are considered a methodology of choice for remote sensing image analysis over the past few years. Due to its effective applications, deep learning has also been introduced for automatic change detection and achieved great success (Khelifi & Mignotte, 2020). DL has been applied to several areas, such as computer vision (Voulodimos et al., 2018), speech recognition (Deng et al., 2013), and information retrieval (Palangi et al., 2016).

The applications of deep learning models and computer vision in the current period are rising by hurdles and restrictions. Computer vision is a field of artificial intelligence where we train our models to interpret actual graphic images. Using deep learning architectures like U-Net can secure superior results on computer vision datasets to perform tedious tasks. While computer vision is a huge field with many dissimilar and exclusive types of problems to solve.

The job in image segmentation is to input an image and split it into numerous lesser fragments. These segments will help with the calculation of image segmentation tasks. For image segmentation, another essential requirement is the creation and use of masks. With the help of masking, we can get the desired outcome essential for the segmentation task. Once we describe the most essential elements of the image found during image segmentation with the help of images and their corresponding masks, we can achieve a gathering of coming tasks with them.

DeepLab is a semantic segmentation model invented and open-sourced by Google. The dense prediction is achieved by simply up-sampling the output of the last convolution layer and computing pixel-wise loss.

DeepLabv3+ is an extension of DeepLabv3 by adding a decoder module to additional improve the segmentation outcomes, especially along object boundaries.

3.2 Land use land cover classification model

The proposed model is for the classification and segmentation of land use land cover using remote sensing images or space-born multispectral IRS LISS- III images. Indian remote sensing satellite images will be acquired from archives of ISRO and pre-processed. After geometric corrections, GCPs will be laid down on images to perform supervised classification. Remote Sensing plays an important role in providing the land coverage mappings and classification of land cover features. Characteristics of land cover land use, the difference of spectral reflectance of different land use, and differences in feature characteristics such as shape and texture are important parameters that should be considered while working land cover land use areas with remote sensing. Therefore, image classification is an important tool for examining and assessing satellite images. Images were taken with the help of Space born remote sensing platforms (Satellites) can be very helpful for the Identification and measurement of land cover land. Furthermore, this method is cost-effective and consumes a lesser amount of time. Figure 3.1 shows the land use land cover classification model.

Figure 3.1: Land use land cover Mapping & Classification Model

The Model is intended to perform the following task.

- Data Acquisition
- Data Pre-processing
- Identification and classification

3.2.1 Data Acquisition

It is a process of gathering information or a procedure of collecting related information. An extensive field survey was done to record ecological features and distribution patterns of different land use. Eight distinct land use classes have been identified in the study area. Quadrats of 30m * 30m sizes (corresponding with a spatial resolution of satellite sensor 30m) were laid down across the marked study area. The numbers of points taken for each class were dependent on the distribution of identified land use classes within the study area. Ground Control Points (GCP) locations of all the quadrats were recorded within an error of ±4m. Images were taken from IRS (LISS III) platform (Spatial Resolution 30 m). The numbers of points taken for each class were dependent on the distribution of identified land use classes within the study area. Figure 3.2 shows the GCPs collection.

Multispectral space-born image (LISS – III) was used for the study. It has more than 100 nm resolution and less the 10 bands. The LISS_ III image contains a total of 4 bands; the spatial resolution is 30 meters. Quadrats of 30m * 30m size were laid down across the study area. Data was collected from the website https://bhuvan-app3.nrsc.gov.in. The data were processed using the software ENVI4.7. A wide field

study was completed to collect environmental landscapes and circulation patterns of different land use and land cover. And the Longitude and Latitude of the location for the respective class are recorded (GCPs). To record GCPS, the GPS device Garmin – eTrex 30 is used for GCPs collection. The GCPs reserved for the respective category were reliant on the allocation of recognized land use classes within the study area. A LISS-III multispectral remote sensing image consists of 4 different bands in separate .tiff files and the number of bands is Band – 2,3,4 and 5(Blue, Green, Red, and near Infrared)

Figure 3.2: GCPs Collection

3.2.2 Preprocessing

3.2.2.1 FCC creation

For the land use land cover Classification in the study, the LISS - III space-born multispectral images were merged into a False Colour Composite (FCC) image. False-color composites (FCC) are created by stacking these multiband.TIFF images are on top of each other and take a stacked grouping of Band-4, Band-3, and Band-2 to generate FCC. After creating FCCs for each image, ground truth masks were created and used to train the deep learning model. These masks are created using the maximum likelihood algorithm on a small region of interest for each class.

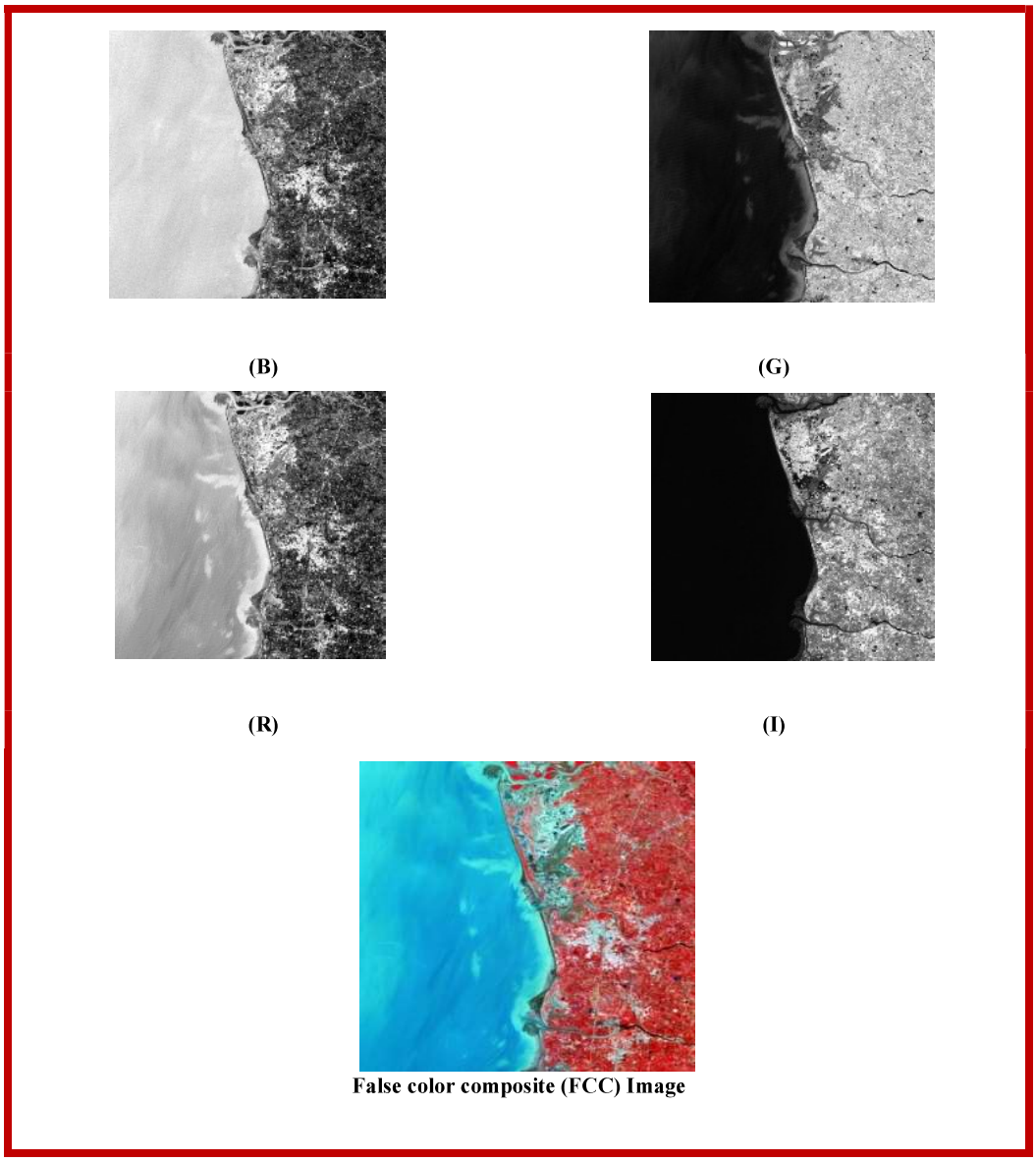

Figure 3.3: FCC Creation

Figure 3.3 represents the creation of the FCC image. Both the FCCs and their corresponding masks are resized to 1024 * 1024 pixels and then divided into patches of size 256 * 256 pixels, 128 * 128 pixels with a striding of 128 and 64.

3.2.2.2 Pre-processing of FCC:

1. False-color composites (FCC) are created by stacking these multiband TIFF images on top of each other. Stacked Band-4, Band-3, and Band-2 to generate FCC.

2. After creating FCCs for each image, ground truth masks were generated which were used to train the deep learning model. These masks are created using the maximum likelihood algorithm on a small region of interest for each class.

3. Both the FCCs and their corresponding masks are resized to 1024 * 1024 pixels and then divided into patches of size 256 * 256 and 128 * 128 pixels with a striding of 128 and 64. Figure 3.3 represents the FCC image divided into 256 * 256 with a striding of 128. Figure 3.4 shows the FCC divided using patches.

4. These images and masks are separated into two subgroups, one for training and another for validation.

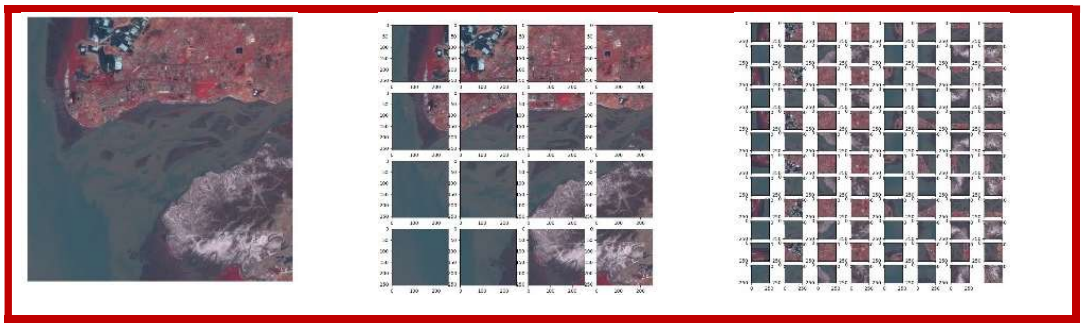

Figure 3.4: FCC (1024 X 1024) (256 x 256)

3.2.2.3 Creation of Mask

The maximum likelihood (ML) classifier was used with LISS- III multispectral image data, where every pixel with the maximum likelihood is classified into the matching class. In ML, a pixel is nominated for a class based on its chance of fitting. Mean vector and covariance metrics are the main essentials of the maximum likelihood that can be improved from training data (Schowengerdt, 2006). The ground truth masks were created using the maximum likelihood (ML) algorithm on the region of interest

for each class. Figure 3.5 shows the basic concepts of maximum likelihood (Fan et al., 2019). Ground truth masks were created after the creation of FCC images and used for model training.

Following is a Discriminant Functions Calculated for Each Pixel:

$$\boldsymbol{g_i(x) - \ln p(\omega_i) - 1/2\ln|\Sigma_i| - 1/2(x - m_i)^t\Sigma_i^{-1}(x - m_i)} \quad (1)$$

Where i is class, x is n-dimensional data in which n represents the total number of bands. $p(\omega_i)$ represents the chance that class ω_i occurs in the image, $|\Sigma i|$ is the determinant of the covariance matrix, $\Sigma i\text{-}1$ is an inverse matrix the mean vector represents by m_i.

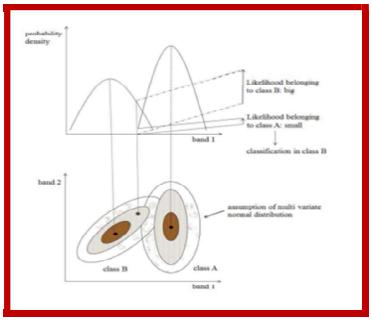

Figure 3.5: Basic concept of ML

After creating FCCs for each image, ground truth masks were created that will be used to train a model. These masks are created using the maximum likelihood algorithm on a small region of interest for each class. Figure 3.6 shows the ground truth masks.

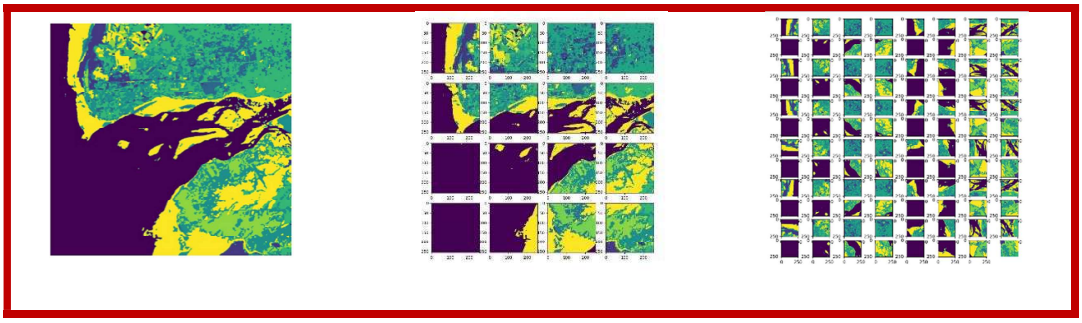

Figure 3.6: Ground Truth Mask

3.2.2.4 Dataset

Dataset 1: FCCs and their corresponding masks are resized to 1024 x 1024 pixels and then divided into patches of size 256 x 256 pixels with a striding of 128. The dataset is of a total of 1470 images. 1,255 images are used for training, while the rest 215 images are reserved for validation and evaluation.

Dataset 2: FCCs and their corresponding masks are resized to 1024 x 1024 pixels and then divided into patches of size 128 x 128 pixels with a striding of 64. The dataset is 13502 images.11250 used for training, while the rest 2250 validation images are reserved for validation and evaluation.

Dataset 3: FCCs and their corresponding masks are resized to 1024 x 1024 pixels and then divided into patches of size 256 x 256 pixels with a striding of 64. The dataset contains 960 images, out of which 800 are for training and 160 for validation.

3.2.3 Identification and classification

This phase of the model is related to the identification and classification of land use land cover classes for IRS LISS -III multispectral remote sensing images. For the identification and classification of multispectral remote sensing images, classifiers U-Net, Tiramisu, and Deeplabv3+ were used.

3.2.3.1 Models

3.2.3.1.1 U-Net :

Semantic segmentation, also known as pixel-based classification, is an important job in which we classify each pixel of an image to a particular class. The Aim of semantic segmentation is the same as traditional image classification in remote sensing, which is usually conducted by applying traditional machine learning techniques such as random forest and maximum likelihood classifier and Deep

Learning techniques like U-net, Mask R-CNN, and Feature Pyramid Networks (FPN), etc.

U-Net was first proposed in the year 2015. U-Net is a convolutional neural network (CNN) that was created for biomedical image segmentation. The network is built on a fully convolutional network (FCN) whose architecture was altered and extended to work with fewer training images and yield more accurate segmentation. U-Net has features like learning segmentation in an end-to-end setting. It inputs a raw image and gets a segmentation map as the output. U-Net performs classification on every pixel so that the input and output share the same size. And it uses very few annotated images.

U-Net architecture can be generally assumed as an encoder network followed by a decoder network. semantic segmentation not only needs discrimination at the pixel level but also a process to project the distinct features study at different phases of the encoder onto the pixel space.

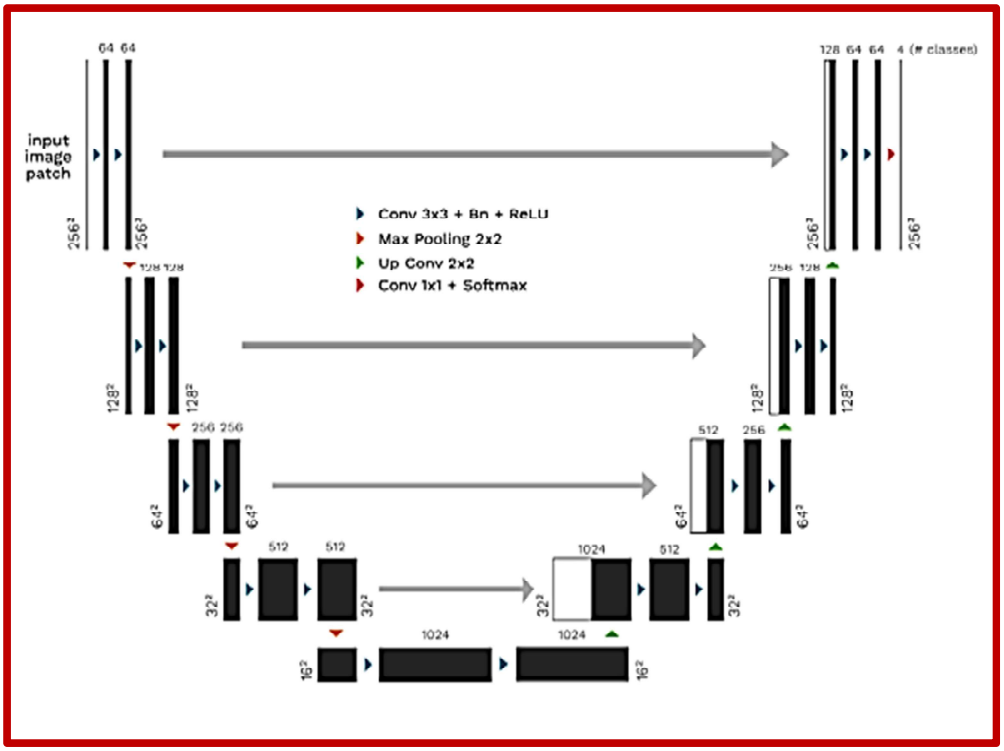

Figure 3.7: U-net Architecture

By taking a brief look at the architecture revealed in the image, we can notice why it is stated to as U-Net architecture. The shape of the architecture is in the form of a 'U' and hereafter the following name. Just observing the structure and the several elements involved in the procedure of the creation of this architecture, we can understand that the network built is fully convolutional. Figure 3.7 shows the U-net architecture.

The arrows denote the various processes, the black containers denote the feature map and the gray containers denote the cropped feature maps from the contracting path.

$$E = \sum w(x)\log\left(P_{k(x)}(x)\right) \quad (2)$$

Where p_k is the pixel-wise SoftMax function applied over the final feature map.

$$P_k(x) = \frac{e^{ak(x)}}{\sum^{k}_{k!=1} eak(x),} \quad (3)$$

And a_k (x) denotes the activation in channel k.

Semantic Segmentation is defined as a pixel-level classification of images where a class is allotted to each pixel of the image. In the present study, there are 4 classes - Water Bodies, Vegetation, Uncultivated Land, and Residential areas. A deep neural network was used to handle this task. A fully-convolutional network (FCN) with skip connections is trained to take an image input of size 256 * 256 * 3, 128 * 128 * 3 and outputs a matrix of shape 256 * 256 * 4, 128 * 128 * 4 i.e., a one-hot encoded version of the mask. The FCN is a U-Net architecture that contains an encoder part and a decoder part. The encoder part contains 5 blocks and each block is 2 (convolution + batch normalization + relu) layers stacked on top of one another and trailed by a max-pooling except for the last block. The output of this encoder part is then inputted into the decoder containing 4 blocks. Each block in the decoder starts with an upsampling of the input followed by a 1 * 1 convolution operation. A skip connection is also used

that concatenates the output of the corresponding encoder block to the output of the upsampling and convolution operation. The concatenated tensor is then again passed to two convolution layers similar to that of the corresponding encoder block. The output of the decoder part is finally fed to a 1 * 1 convolution with the number of filters equivalent to the number of classes which is 4.

3.2.3.1.2 Deeplabv3+

Deeplabv3+ was invented by google and open-sourced back in 2016. DeepLabv3+ is a semantic segmentation architecture that builds on DeepLabv3 by adding an actual decoder module to improve segmentation results.

DeepLabv3+ is a deep learning model for semantic image segmentation where the goal is to assign semantic labels to every pixel in the input image. The DeepLabv3+ model has an encoding phase and a decoding phase. The encoding phase extracts the important data from the image using a convolutional neural network (CNN) while the decoding phase rebuilds the output of suitable dimensions based on the data obtained from the encoder phase. The decoder module was added to give better segmentation outcomes along object boundaries. DeepLab supports various network backbones like MobileNetv2, Xception, ResNet, PNASNet, and Auto-DeepLab.

DeepLabv3+ is an extension of DeepLabv3 by adding a simple yet effective decoder module to further refine the segmentation results, especially along object boundaries. Figure 3.8 shows the Deeplabv3+ architecture.

Encoder: it uses Aligned Xception in place of ResNet-101 as its key feature extractor (encoder) but with an important alteration. All max pooling operations are substituted by depth-wise separable convolution.

Decoder: The encoder is based on an output stride of 16. Instead of using bilinear up-sampling with a factor of 16, the encoded features are first up-sampled by a factor of 4 and concatenated with equivalent low-level features from the encoder module having the same spatial dimensions. Before concatenating, 1 x 1 convolutions are

applied to the low-level features to reduce the number of channels. After concatenation, a few 3 x 3 convolutions are applied and the features are up-sampled by a factor of 4. This gives the output of the same size as that of the input image.

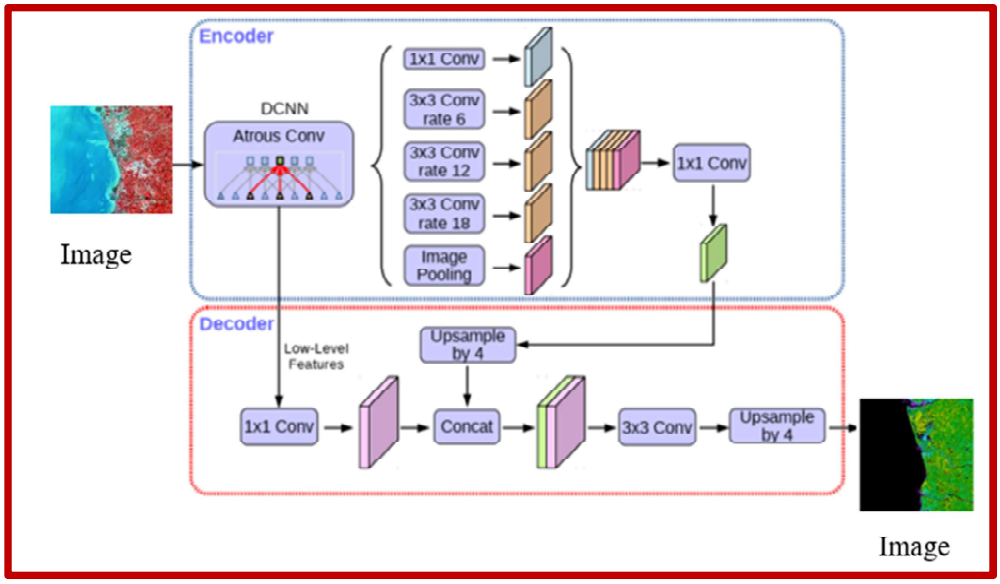

Figure 3.8: Deeplabv3+ model. [ref: https://arxiv.org/ab s/1802.02611]

3.2.3.1.3 Tiramisu

Tiramisu is a polyhedral compiler for dense and sparse deep learning and data-parallel algorithms and directs a huge set of loop optimizations and data design alterations. It is the only open-source DNN compiler that optimizes sparse DNNs and marks distributed architectures. It can perform complex loop transformations and uses dependence investigation to assure the accuracy of optimizations. Tiramisu has also demonstrated its performance on various standards like deep learning operations (Convolution, ReLu, MaxPool, Sparse Neural Networks, etc.) and linear algebra. However, the Tiramisu network, which itself is a modified U-Net, is much larger and took longer to train. Figure 3.9 shows the tiramisu architecture.

$$xl = Hl\ (xl - 1) \qquad (4)$$

in standard convolution, xl is computed by applying a non-linear transformation Hl to the output of the previous layer xl-1.

$$xl = Hl\ (xl - 1) + xl - 1 \qquad (5)$$

ResNet introduces a residual block that sums the identity mapping of the input to the output of a layer

.

$$xl = Hl\ ([\ xl\ -1,\ xl\ -2,\ \ldots\ldots\ldots,\ X0]) \qquad (6)$$

DenseNet input concatenates all previous feature outputs in a feedforward fashion for convolution.

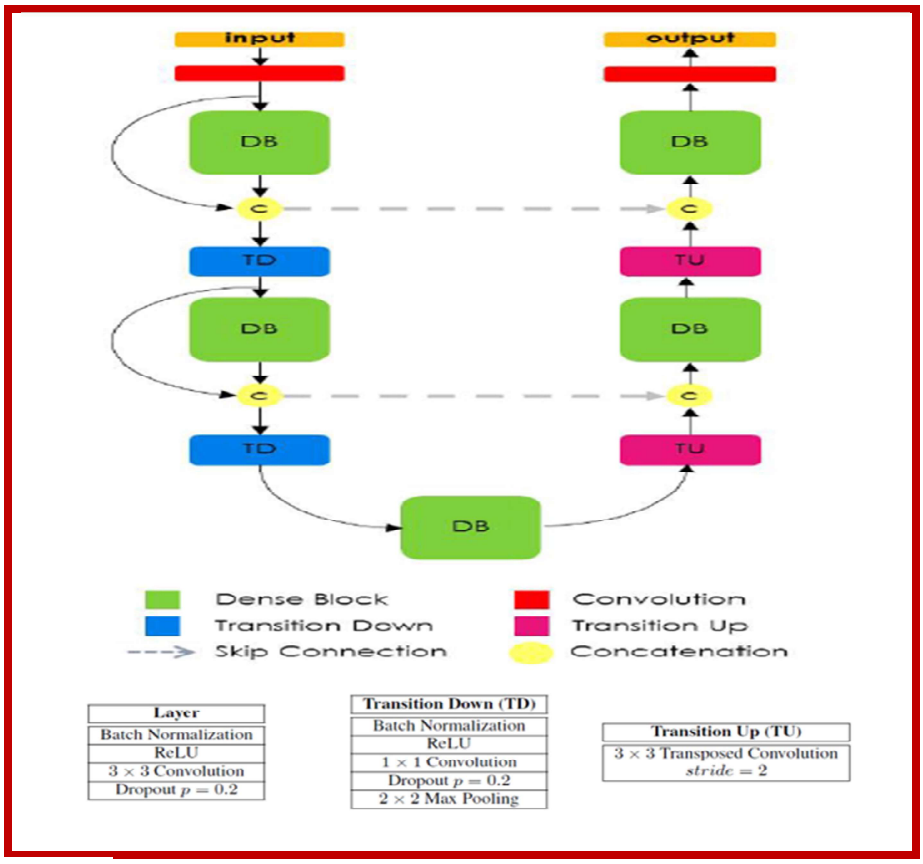

Figure 3.9: Tiramisu Architecture [Ref https://towardsdatascience.com/review-fc-densenet-one-hundred-layer-tiramisu-semantic-segmentation-22ee3be434d5]

3.2.3.2 Training Sample Selection

It is the most important component of remote sensing classification and measuring the quality of the region of interest (ROI). Accurate classification accuracy is dependent upon good training sample selection. Classification correctness is mainly determined by ROI separability. High-quality classification training samples (with high ROI reparability) define the classification accuracy to a firm extent.

3.2.3.3 Training Configuration

While data ingestion i.e. before passing the images and masks to the model for training, we normalize the input images by clipping them to [0.0, 255.0] while the masks are one-hot encoded according to the total number of classes i.e. 4. In addition, random augmentations are also applied to the batch of images and masks before passing them to the model for training. This expands our dataset and makes the model robust enough to encounter different orientations than just the training data. Data augmentation is a factor that when done correctly, prevents overfitting. A custom image data generator is created to fulfill the requirements of this data ingestion pipeline.

Table 3.1 is the table of hyperparameters and other configurations used for training the model.

Hyperparameters & Configurations	Values
Train Batch Size	16
Validation Batch Size	16
Input Image Shape	(256, 256, 3), (128,128,3)
Number of classes	4
Epochs	50
Loss Categorical	Focal Loss*
Optimizer	Adam, RMSprop

Metrics	Dice Coefficient*
Class Weights	[1.69941, 0.53043, 1.23977, 1.38949]

Table 3.1: Hyperparameters and other configurations used for training the model.

3.2.3.4 Algorithm steps

The steps of the algorithm are as follows:

(1) The pre-processing steps include remotely sensed image alteration, registration, and the masking of the image. The images containing band- 2 to band – 4.

(2) For training and testing, a total of four types of classes were selected, such as Water Bodies, Vegetation, Uncultivated Land, and Residential areas. This study used the ENVI image processing software (ROI Tool) to pick the four types of classes.

(3) A total of four types of classes were used for the building and training of the model. The model structure and parameters were kept for successive image segmentation or classification after training.

(4) The model which was trained in step -3 was applied for classification.

(5) Find whether all the classifications were finished or not. If all is done then the classification outcome will be displayed and terminate the algorithm.

3.2.3.5 Final classified image

The final classified image contains different classes of land use/land cover classification. The final image is classified into the classes like water bodies, agricultural land, residential areas, and uncultivated land.

3.3 Research Methodology

Figure 3.10 represnts the reseach methodology to perform Land use land cover classification on IRS LISS – III multispectral image , which includes several basic steps such as data collection, pre processing,dataset development,model development ,training and testing and final classified image.

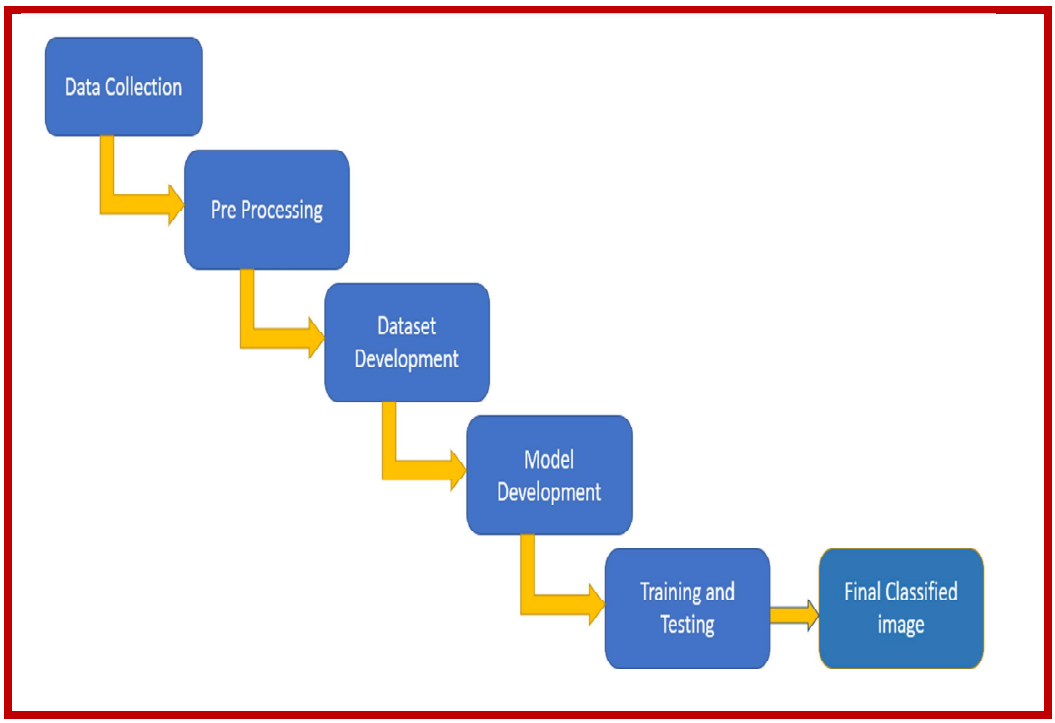

Figure 3.10: Research Methodoloy

References:

[1] Cheng, G., Zhou, P., & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(12), 7405-7415.

[2] Deng, L., Hinton, G., & Kingsbury, B. (2013, May). New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8599-8603). IEEE.

[3] Fan, J., Cao, X., Yap, P. T., & Shen, D. (2019). BIRNet: Brain image registration using dual-supervised fully convolutional networks. *Medical image analysis*, *54*, 193-206.

[4] Gómez-Chova, L., Tuia, D., Moser, G., & Camps-Valls, G. (2015). Multimodal classification of remote sensing images: A review and future directions. *Proceedings of the IEEE*, *103*(9), 1560-1584.

[5] Goslee, S. C. (2011). Analyzing remote sensing data in R: the landsat package. *Journal of Statistical Software*, *43*, 1-25.

[6] Jaiswal, R. K., Saxena, R., & Mukherjee, S. (1999). Application of remote sensing technology for land use/land cover change analysis. *Journal of the Indian Society of Remote Sensing*, *27*(2), 123-128.

[7] Khelifi, L., & Mignotte, M. (2020). Deep learning for change detection in remote sensing images: Comprehensive review and meta-analysis. *Ieee Access*, *8*, 126385-126400.

[8] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(4), 694-707.

[9] Schowengerdt, R. A. (2006). *Remote sensing: models and methods for image processing*. Elsevier.

[10] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*.

[11] Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[12] Zhang, L., Zhang, L., & Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and remote sensing magazine*, *4*(2), 22-40.

[13] https://arxiv.org/ab s/1802.02611