

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NORTH SOUTH UNIVERSITY



CSE499

Senior Design Project Report

Effect of Label Noise in Patch-wise Multi-class Semantic Segmentation of Sedimentation from High-resolution Satellite Image.

Group Members

<u>Name</u>	<u>NSU ID</u>
Tahmid Hasan Pranto	1711697042
Abdulla All Noman	1711793042
Asaduzzaman Noor	1631231042
Ummeh Habiba Deepthy	1621136042

Supervisor

Dr. Rashedur M. Rahman

Professor, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh.

Spring 2021

DECLARATION

We hereby declare that this is our original work. This work, fully or partially have not been submitted anywhere for any other purpose. Every material and reproduced works used for the purpose of this study has been properly acknowledged.

Declared By:

.....
Name: Tahmid Hasan Pranto

ID: 1711697042

.....
Name: Abdulla All Noman

ID: 1711793642

.....
Name: Asaduzzaman Noor

ID: 1631231042

.....
Name: Ummeh Habiba Deepy

ID: 1621136042

Approval

This senior design capstone project entitled “Patch-wise semantic segmentation of sedimentation from high-resolution satellite images using deep learning.” by Tahmid Hasan Pranto (NSU ID 1711697042), Abdulla All Noman (NSU ID 1711793642), Asaduzzaman Noor (NSU ID 1631231042) and Ummey Habiba Deepy (NSU ID 1621136042) is approved in partial fulfillment of the requirement of the Degree of Bachelor of Science in Computer Science and Engineering on May and has been accepted as satisfactory.

Approved By:

.....
Supervisor

Dr. Rashedur M. Rahman
Professor, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh.

.....
Department Chair

Dr. Rezaul Bari
Associate Professor, Department of Electrical and Computer Engineering
North South University, Dhaka, Bangladesh.

Acknowledgement

Towards the accomplishment of this project, the constant support, encouragement and guidance of Dr. Rashedur M. Rahman, our honourable supervisor for this project was indispensable. We would like to express our profound gratitude to our honorable supervisor, Dr. Rashedur M. Rahman, for his diligent guidance, valuable feedbacks, patience and encouragement towards the completion of this project. At times we were in a state of great doubt about the completion of this project but Dr. Rashed always encouraged, directed and supported us with conceivable solutions. Without his help, it would not have been possible for us to bring our project to its current state. Finally, we would like to thank everybody who supported us and provided us with counsel for the completion of this project.

Abstract

The coastline of a country plays innate role in a country's geo-political, socio-economical and development aspect. Sediment load has direct impact to the biodiversity deep under the sea as well as on the agricultural economy of a country. Constant monitoring and data analysis of sedimentation load thus becomes very important. Remotely sensed satellite information is one of the most efficient and reliable way to get data and in recent times, satellite data availability has increased significantly, helping researchers worldwide to explore, analyze and approach different problems using the most recent computer vision techniques. In this work, we created dataset of Bangladesh marine area sedimentation for segmenting sediment load into 5 different classes (Land, High Sediment, Moderate Sediment, Low Sediment and No Sediment). We used a modified U-Net architecture to segment our training images using corresponding label masks. As high-resolution satellite images cannot be directly used as inputs of deep learning models, we applied a patch-wise learning approach. Among the four datasets, our model performed best with Dec-2019 dataset with validation dice coefficient of 87.81%, validation loss of 0.60 and validation pixel accuracy of 78.97%. We further tested our models performance under three kinds of label noise (NCAR, NAR and NNAR). We found that for NCAR, performance of the model decreased slightly as we increase the percentage of noise. For NAR, after a certain level of rotation the models performance changes drastically and for NNAR model completely gets confused and makes random guesses for most of the pixels.

Contents

1	Introduction	10
1.1	Background & Motivation	10
1.2	Study Objective & Aspiration	11
1.2.1	Study Objective	11
1.2.2	Study Aspiration	13
1.3	Cognitive Challenges	14
2	Related Works	15
2.1	Deep Neural Networks for Image Segmentation	16
2.2	Effect of Label Noise	18
3	Project Plan	20
4	Study Area	22
5	Data Collection & Preparation	24
5.1	Labelling Satellite Images in Earth Engine Code.	24
5.2	Data Preparation	25
6	Methodology : Modified U-Net	28
6.1	Contracting Part	29
6.2	Expansive Part	29
7	Experiment setup	30
7.1	Hardware Requirement	30

7.2	Software Requirement	30
7.3	Hyper Parameter	31
7.4	Code optimization	31
8	Evaluation metrics and loss function	33
8.1	SoftMax	33
8.2	Dice Coefficient	33
8.3	Pixelwise accuracy	34
8.4	Categorical Cross-Entropy	34
9	Result & Analysis	35
9.1	Analysis of Result under noiseless labels	35
9.2	Analysis of Result Under Noisy Labels	39
9.2.1	NCAR - Gaussian Random Noise	39
9.2.2	NAR - Rotation with Nearest Fill Mode Label Noise	42
9.2.3	Label Flip Noise	46
10	Project Impact	49
10.1	Economic Impact	49
10.2	Impact on research community	50
10.3	Social Impact	51
11	Conclusion	52
11.1	Project Findings	52
11.2	Future Work	54

List of Figures

1	Statistical Taxonomy of Different Label Noise. Figure Inspired by [1]. (a) NCAR - Noisy Completely at Random. (b) NAR - Noisy at Random. (c) NNAR - Noisy not at Random.	18
2	Project Plan. Numbers in the Figure Denotes the Chronology of the Tasks During Our Study.	20
3	(a) Bangladesh Geographical Location. (b) High-Resolution Satellite Image of Target Region. (c) Targeted Marine Region.	23
4	(a) Original Satellite Image having 12 Multi-Spectral Band but only RGB Bands Selected for the Purpose of this Study. (b) Labelling using in-built Labelling Tool of Google Earth Engine. (c) Processing the Labelling File using QGis Software. (d) Adding the Shapefile as a Band to the Original Satellite Image.	24
5	Image Retrieval and Band Separation. (a) Original Satellite Image With Label Added as the 4th Band. (b) Band Stack of Downloaded Tiles. (c) Retrieved Image. (d) Retrieved Mask.	26
6	Two Kinds of Regions in the Inputs (a) Class Join Regions. (b) Single Class Regions.	27
7	Modified U-Net Architecture Used in this Study.	28
8	Training and Validation Dice Coefficient of Four Datasets Over 30 Epochs.	36
9	Training and Validation Loss of Four Datasets Over 30 Epochs.	37
10	Image and their Corresponding True Mask and Predicted Mask for Class Join Regions.	38

11	Image and their Corresponding True Mask and Predicted Mask for Single Class Regions.	38
12	Example of Gaussian Label Noise in a Range of 1% to 25%.	39
13	Class-Join region prediction Under Gaussian Label Noise Ranging from 1% to 25%.	40
14	Single Class Region Prediction Under Gaussian Label Noise Ranging from 1% to 25%.	41
15	Example of Rotational Label Noise (10° to 40°) with Nearest Fill Mode.	43
16	Class-Join Region Prediction Under Rotational Label Noise Ranging from 10° to 40°	44
17	Single Class Region Prediction Under Rotational Noise Ranging from 10° to 40°	45
18	Example of Horizontal and Vertical Flip Label Noise.	46
19	Class-Join Region Prediction Under Horizontal and Vertical Flip Label Noise.	47
20	Single-Class Region Prediction Under Horizontal and Vertical Flip Label Noise.	47

List of Tables

1	Hardware Specifications for this Project.	30
2	Metric, Accuracy and Loss for Four Year Dataset.	36
3	Dice Coefficient, Pixel Accuracy and Loss for Gaussian Label Noise.	42
4	Dice Coefficient, Pixel Accuracy and Loss for Rotational Label Noise.	43
5	Dice Coefficient, Pixel Accuracy and Loss for Label Flip Noise.	46

Chapter 1 **Introduction**

Bengal delta is the largest delta in Asia and second largest in the world in terms of its size. On the other hand, in terms of population, Bengal delta is the most populated delta in the world [2]. A huge number of rivers cross Bangladesh on their hydrologic flow path, which ends in the Bay of Bengal. Thus, sediment load in this particular area holds significant attention both for maritime diversity as well as country's economy. This section vividly describes the background motivation and challenges of our study.

1.1 Background & Motivation

Originated from mountains, rivers across the world flows towards the sea. Big rivers crosses more than one country in their path making themselves inter-country rivers. There are inter country rivers too. According to the data of Water Development Board Bangladesh, there are a total of 405 rivers flowing across the country among which 48 are inter-country and the other 357 river are intra-country [3]. Bangladesh being a land of rivers, has a significant tie between its economy, agriculture, and people's life with its rivers.

Bangladesh is a low land which is situated in lower stream of its intra-country rivers which finally falls into the Bay of Bengal. On the other hand, the intra-country rivers also falls into the Bay of Bengal creating a huge load of sedimentation in the region where these rivers meet the sea. According to CEGIS study, [4] conducted by the Center for Environmental and Geographic Information Services of Bangladesh, an estimated

amount of 1.2 BT(billion-tonnes) of slit is being carried and finally discharged into the Bay of Bengal by the GBM (Ganga-Brahmaputra-Meghna) river system of Bangladesh.

1.2 Study Objective & Aspiration

1.2.1 Study Objective

The overall maritime boundary of Bangladesh expands up-to 354 NM which includes 12 NM of sovereign right over the resources and an economic marine zone of 200 NM [5]. This marine boundary of Bangladesh has a great impact on the country's overall economy [6]. Thus, understanding the load of sedimentation in this huge marine region, holds a great impact in several related sectors like marine biology, blue economy and aquaculture. Conducting field research in this enormous areas can both be effort-taking and costly. Satellite image can aid the purpose of diverse study in marine area using the most recent techniques in machine learning paradigm. Crucial cases such as soil erosion, flood management, oceanology and biodiversity under the sea can be greatly benefited by using these techniques.

Remotely sensed satellite information is a great source of multi-temporal geo-spatial data, especially images that can be used in deep learning algorithms for accomplishing a wide range of tasks [7]. The reason that satellite images are being used in these studies is because of its representation of high valued information which can only be seen by the sensors of a satellite. Satellite images can often be huge in size but, consisting of compact high level information in small pixels. This study is directed towards a specific patch-wise approach to learn five different sedimentation type from large satellite im-

ages using a deep learning technique known as image segmentation.

The process of splitting a visual input into different segments is known as image segmentation. Image segmentation has a wide range of application [8] but it can be divided into two fundamental category, instance segmentation and semantic segmentation. In instance segmentation, a segment is a parts of an object or a whole object where the bare minimum threshold to be considered as a segment is a pixel. Each segment can be consisting of sets of pixels or "super-pixels". But in case of semantic segmentation every pixel becomes classified or predicted as a class or segmentation type. Deep learning techniques have been well known for learning different patterns from data inputs to predict object classes from each pixel thus creating segments [8].

Deep learning is going through major advancements in the past few years. Convolutional Neural Network (CNN) has already achieved great success in segmentation task [9]. But for the last three decades, one of the most difficult problems in computer vision has been accurate image segmentation [10]. Recent progress in semantic image segmentation using transfer learning has shown some promising scenarios and have significant improvement over previous semantic segmentation approaches [11, 12, 13, 14]. In this study, we will be using high resolution satellite imagery with deep neural networks to segment sedimentation in the entire marine region of Bangladesh.

Segmentation of remotely sensed images has further greater implications as these images contain high-level information captured from the top that can be of great help to understand certain aspects of the world. Satellite images contain more uniformed and

compact meaningful data than conventional images as remotely sensed images capture spatial data from top where it can cover larger regions at high resolution. There are so many worthwhile applications of remotely sensed images. Road extraction[15], building detection[16], land cover classification[17], urbanization[18] and slum detection[11] etc.

Noise is a phenomena where data is either corrupted or not in its actual form or values. Satellite images are often contained with various kinds of noises among which salt and paper noise [19], speckle noise [20], stripping noise [21] etc are commonly seen and found. Various methods of removing these noises from image have also been shown by the research community [19, 20, 21]. But, noise can also exist in labels and masks that we use to train our deep learning models. In this study, we will be using high resolution satellite imagery with deep neural networks to segment sedimentation in the entire marine region of Bangladesh with a profound focus on label noise. Using patch-wise learning method, our focus will be to evaluate the performance of deep learning segmentation algorithm for different type of noise as well as different magnitude and volume of these noises.

1.2.2 Study Aspiration

Our study is targeted towards the following aspirations.

- Creating dataset using satellite image.
- Segmenting sedimentation type into five different classes.

- Analysis of label noise on performance of segmentation model.

1.3 Cognitive Challenges

First of all, all acquirable satellite images are not usable as inputs of deep learning models. Only refined and properly cleaned images had to be filtered for the final use. On the other hand, satellite image used in our case is enormous in size, which cannot be used directly as model input, patch-wise learning technique had to be taken. Moreover, these images are often affected with values that are hard to be dealt with (e. g. infinite values, 'Nan' values). These values does not cause any error while reading the images, but affected the error metrics substantially. This kind of concealed flaws are hard to find which caused us great difficulty to fine tune the models performance. Dataset size was very large (450 GB) and to run the model we needed a high configuration computational setup which we did not have. But, through code optimization (details given on subsection 7.4), we were able to do our experiments finally.

Chapter 2 Related Works

Extraordinary improvement of deep learning methods has created more advanced and efficient techniques for different computer vision tasks over the last decade [22]. Tasks like medical image classification, [23] object detection [24] and segmentation [25] etc. are now being done with remarkable performance. Traditional segmentation models like measurement space guided spatial clustering, single and hybrid linkage region growing scheme, spatial clustering scheme and split-and-merge scheme were used before the year 2000 [26]. After year 2000, with great improvement of artificial intelligence, segmentation tasks were done using clustering techniques, classification algorithms and mixture of both [27] [28]. But, the reign of deep learning techniques has bought remarkable improvement of segmentation tasks [25].

The field of deep learning saw a paradigm shift when Alex Krizhevsky outperformed all other state-of-the-art deep learning techniques in the ImageNet LSVRC-2010 contest [29]. A successful combination of 650,000 neurons and 60 million parameters which was trained on 1.2 million high resolution images and classifies them in 1000 different classes. Since then, researchers have shown numbers of different deep learning techniques and approaches for innumerable use cases. Image segmentation has been greatly improved by using different deep learning approaches with remarkable performance [30]. The fundamental essence or the most important aspect of image segmentation is segmenting/classifying/mapping different section of an image into specific classes sometimes recognizing objects from an unknown image and sometimes classifying a single pixel. Section 2.1 contains the study on related literature focusing on

the application of deep learning techniques applied in image segmentation.

2.1 Deep Neural Networks for Image Segmentation

As the deep learning development progresses over the last decade, convolutional neural network (CNN) has achieved a great success in almost all the computer vision tasks [31]. But, for the last three decades, one of the most difficult problems in computer vision has been image segmentation [11]. Image segmentation holds a major position in the domain of computer vision for its further implications in many other sectors. Crucial sectors like medical [32] [33], geo-spatial data analysis [34] and autonomous driving [13] uses segmentation technique as solution to several other implementation.

Recent progress in semantic image segmentation [10] has shown some promising outputs in terms of their accuracy and performance. And recent advancement in Convolutional Neural Networks have significant improvement over previous semantic segmentation techniques [35] [36]. Wu et. al. [37] came up with CasFCN, a less complex but high-performing model for ultrasound maternity image segmentation. Long et. al. [36] proposed a segmentation architecture that uses classification models like AlexNet, GoogleNet or VGG as full convolution part of their model. They uses a combination of upsampling and patch-wise training mechanism and to fuse coarse, they added skip connections between the layers for a better feature extraction. Although deep learning performs well for segmentation tasks but vanishing gradient and overfitting still becomes a problem while training deep neural networks [38].

Complex images are challenging to segment as localized information can be confusing for the model to interpret. For example, satellite images are collection of high-level squeezed information which can be very difficult to segment. This is where more improved and advanced deep learning models like UNet [39], DeepLabV3 [40] comes handy. UNet is a FCNN (Fully Connected Neural Network) which uses skip connection between the 'contracting' and 'expansive' parts of the network for the best possible feature extraction [32]. So far, the most impactful application of U-Net based segmentation has been seen on the medical sector [30] [33]. It has been used for nuclei segmentation in histology images [32] liver and tumor segmentation [30], chronic stroke lesion segmentation [34], retina-vessel segmentation [26] as well as for heart-conditions from ultrasound images (echo-cardiography) [27].

The promising performance of U-net like architecture in medical sector encouraged so many researchers to use in other scenarios where segmentation can be proven beneficial. Though, it was initially developed for biomedical image segmentation, several other implementations of UNet have been seen for many other use cases like satellite image segmentation [41], street tree segmentation [42], tomato leaf segmentation [43], real-time hair segmentation [44], hand segmentation in complex background [28], pedestrian segmentation [45] and sea-land segmentation [46]. In this study, we used a modified U-Net architecture for segmenting sedimentation type in Bangladesh Marine region from high resolution satellite images. Section 2.2 contains the related literature study on effect of label noise in the performance of deep learning models.

2.2 Effect of Label Noise

The fundamental aspect in almost all kind of neural network models is learning from the input data and comparing the predicted output with some corresponding labels. Without the labels being accurate, no matter how perfect the model predicts, evaluation metrics will take a hit. There are many ways these datasets are made. And the process of building a dataset can often produce label noise which can directly affect the model's accuracy. Three major kinds (NCAR, NAR and NNAR) of label noise is seen in machine learning paradigm [1]. The statistical taxonomy and increasing complexity dependency of the three type of label noise is shown in the diagram below.

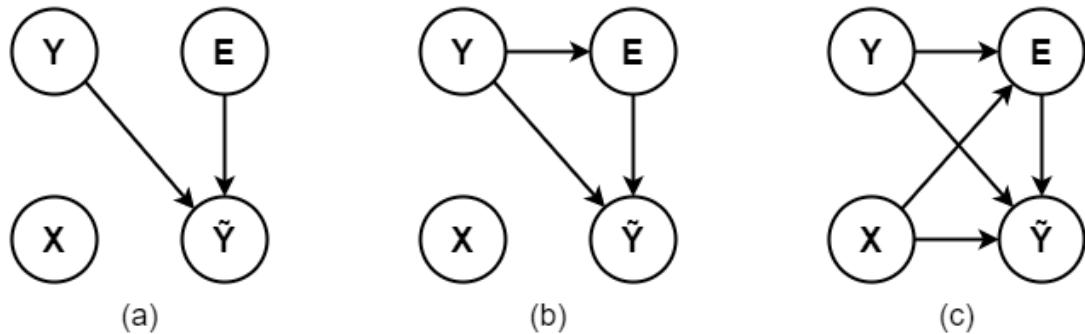


Figure 1: Statistical Taxonomy of Different Label Noise. Figure Inspired by [1]. (a) NCAR - Noisy Completely at Random. (b) NAR - Noisy at Random. (c) NNAR - Noisy not at Random.

In the figure above, Y represents the true labels, E represents the possibility of occurrence of error, X represents the input feature vector and \tilde{Y} represents the predicted output. From the figure we can see that occurrence of a NCAR type of label noise is not dependant on the other variables, including the true labels itself with a error probabil-

ity of $p_e = P(E = 1) = P(Y \neq \tilde{y})$ where the incorrectness in the label is chosen at random [1] [47] [48]. In the case of NAR, the possibility of error in labels is dependent on the true label while the error probability is $p_e = P(E = 1) = \sum_{y \in Y} P(Y = y)P(E = 1|Y = y)$ [1]. NNAR is the most realistic case of noise as the source of noise can both be the input images and the label images [49] [50]. This representation of label noise is the most complex label noise relation with the probability of error which follows the equation, $p_e(x, y) = P(E = 1|X = x, Y = y)$ [1].

It has been already shown by many researchers that deep learning models are robust to label noise [49] [50] [51], however up-to a certain level of label noise, this statement holds. Görkem Algan and İlkkay Ulusoy showed that feature dependant label noise which is a NAR type of label noise, affect the models performance by reducing the test accuracy [49]. On another study, Sornkitja. et. al. showed that, the more the presence of label noise, the more the accuracy is reduced in the case of satellite image classification but proper distribution of high-intensity noise into the study area tends to increase the correctness of the model [52].

Chapter 3 Project Plan

Starting from labelling and collecting the images from google earth engine (GEE), preparing them for deep neural network model and finally analysing the output, several intermediate resolutions were undertaken. Approaching the tasks with an earlier plan was proven very effective to us during the implementation of that task for our study. The project plan is shown in the figure below (figure 2).

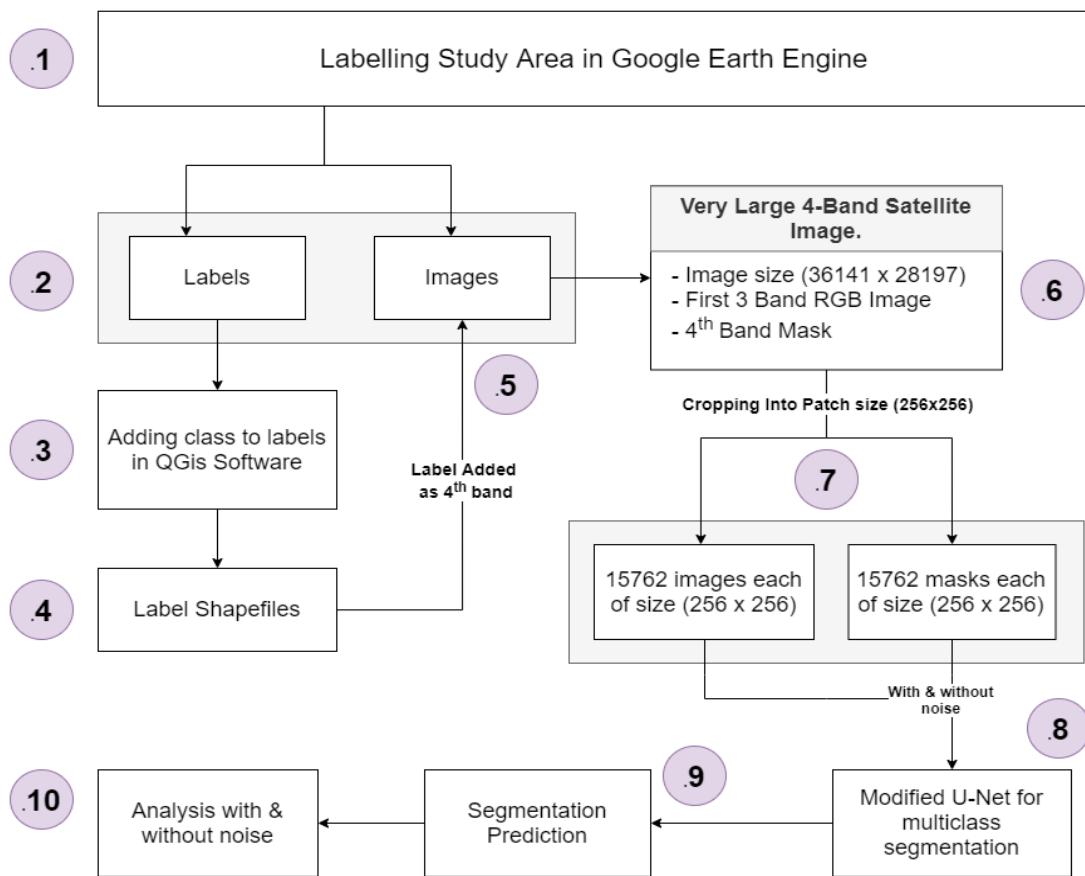


Figure 2: Project Plan. Numbers in the Figure Denotes the Chronology of the Tasks During Our Study.

The figure shows the sequence of the tasks approached by us during this study. First we labelled the study area and collected the images and labels from google earth engine. Then we processed the labels to create shapefile in QGis software. The the shapefile was again added the original images as the fourth band of it. After that, the image and labels were separated using python GDAL library. The input image is the combination of first three bands which are the red, green and blue band respectively. The last band in our labelled band which acts as the mask to deep neural network for a segmentation task. Then the final image and mask was cropped into 15762 patches. These patches are the images and masks which was divided into training and validation set for our final analysis. Noise was pushed into the labels for analyzing the effect of different kinds of noise in the performance of the model. Detailed discussion about these steps are shown in the following sections.

Chapter 4 Study Area

Bangladesh's marine waters occupy an area of 165,887-kilometer square, which is greater than the total ground region of this country. The latitude and longitude of this area is 20.99°N, 90.73°E [53]. The total coastal zone of Bangladesh consists of 19 districts, which covers an area of 47,201 kilometers square [54]. The country also has a 580 km long coastline, which is vindicated by many rivers [55]. The regional coastal districts are Jessore, Narail, Gopalganj, Shariatpur, Chandpur, Satkhira, Khulna, Bagerhat, Pirozpur, Jhalakati, Barguna, Barisal, Patuakhali, Bhola, Lakshmipur, Noakhali, Feni, Chittagong, and Cox's Bazar [54]. In the Bay of Bengal, Bangladesh's coastal marine regions are spilled into three zones which are 12 nautical miles(NM) of territorial waters, 200 NM of exclusive economic zone, and 350 NM of the sea bed from Bangladesh baseline [39]. The geographic features of Bangladesh divided the coastal zone of this country into three parts: eastern zone, central zone and western zone from which the central zones are considered more vulnerable in case of soil erosion as the river Meghna is included in this region [54]. Our study area has been depicted in the figure below (figure 3).

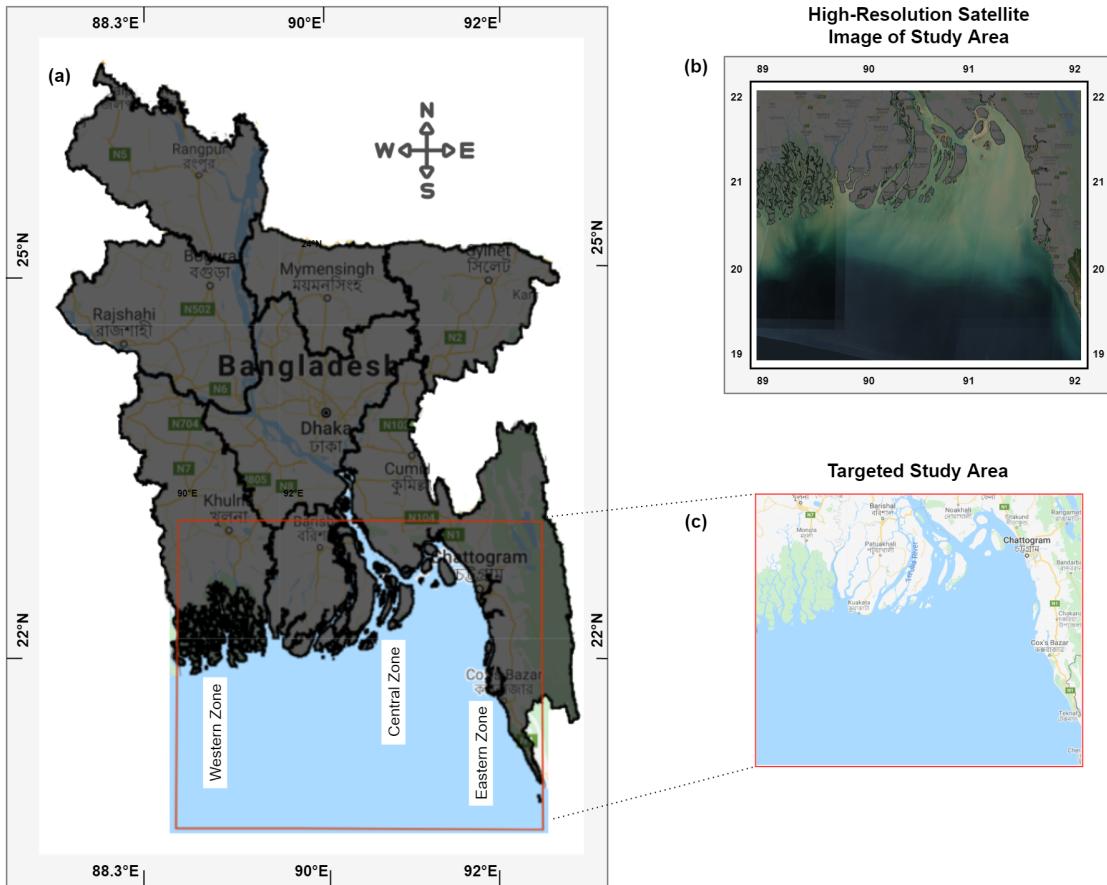


Figure 3: (a) Bangladesh Geographical Location. (b) High-Resolution Satellite Image of Target Region. (c) Targeted Marine Region.

Figure 3 shows the exact targeted marine region that we're working with in this study. We chose this specific area because all the rivers flowing across the country opens up to the Bay-of-Bengal in our targeted area. High-resolution satellite image tiles have been clipped on this region on google earth engine. But these image tiles need further pre-processing which has been depicted in the following section.

Chapter 5 Data Collection & Preparation

5.1 Labelling Satellite Images in Earth Engine Code.

Google earth engine code provides an in-built tool for labelling satellite images into different classes and also provides JavaScript API to conduct different pre-processing and uploading the images directly to google drive. The process of labelling is shown in the figure below (figure 4).

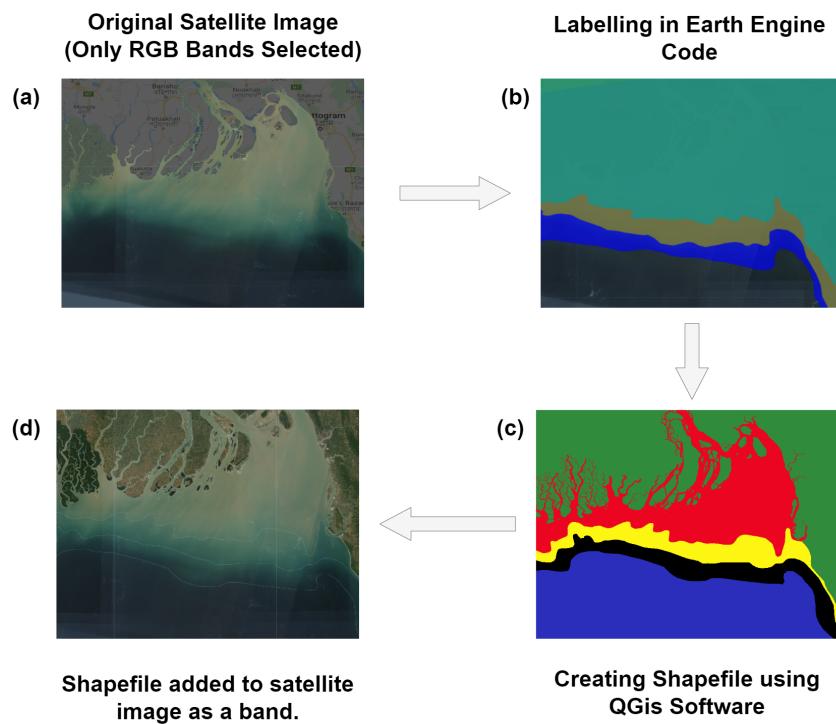


Figure 4: (a) Original Satellite Image having 12 Multi-Spectral Band but only RGB Bands Selected for the Purpose of this Study. (b) Labelling using in-built Labelling Tool of Google Earth Engine. (c) Processing the Labelling File using QGis Software. (d) Adding the Shapefile as a Band to the Original Satellite Image.

We're segmenting the image into five different classes where the mask should be categorical. Clear image is a must as satellite images are often distorted by cloud shadows. JavaScript API allows filtration for best possible image extraction. We chose images which had cloud cover less than 1%. We also checked if the image tile passes geometric, radiometric and sensor tests that google earth engine (GEE) API provides for better image filtration. For labelling we used polynomial tool of GEE. Then we downloaded these labels, processed them spatially and categorized them into 5 classes using QGis software. Shapefile created using QGis will be used as our original mask image. This mask has been added to the original image as the fourth band in the corresponding coordinate of our target study area. The final image is the image that we will use for our study.

5.2 Data Preparation

We used google earth engine's JavaScript API to accumulate images of our study area for four different time frames. They are as follows: November-December 2018, January-February 2019, November-December 2019 and January-February 2020. Image of each time-frame was 36141 x 28197 pixels in width and height correspondingly which was captured in a rectangular zone that covers the area located at $(89.09, 22.91)^\circ$ in North-West, $(92.34, 22.91)^\circ$ in North-East, $(92.34, 20.37)^\circ$ in South-East and $(89.09, 20.37)$ in South-West. For collecting the image data, we used the Copernicus Sentinel-2 satellite imagery at 10m resolution. Each image consists of a total of 4 spectral bands where the first three bands represent RGB channels and the last band represents the corresponding labels. By separating the first three bands, we got our input image and we got the mask image by separating the last band. Image extraction and band separation has

been shown in the figure below.

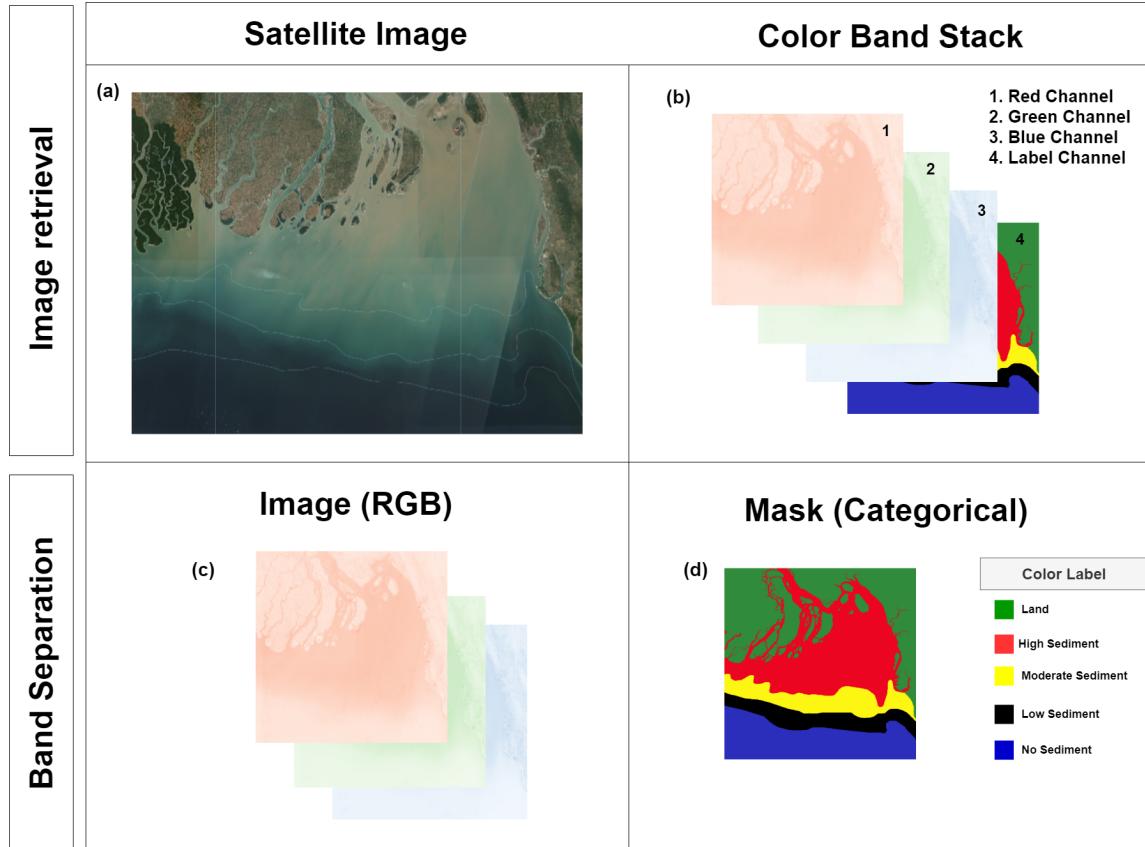


Figure 5: Image Retrieval and Band Separation. (a) Original Satellite Image With Label Added as the 4th Band. (b) Band Stack of Downloaded Tiles. (c) Retrieved Image. (d) Retrieved Mask.

But both images and mask contains two kinds of regions that the model has to distinguish. The figure below shows the two kinds of regions which are (1) where two class joins, (2) single class regions. The figure below depicts these two regions in the image. This enormous images cannot be used to train the model. That is why, using the image translation method of python GDAL library, we further created 15762 patches both for

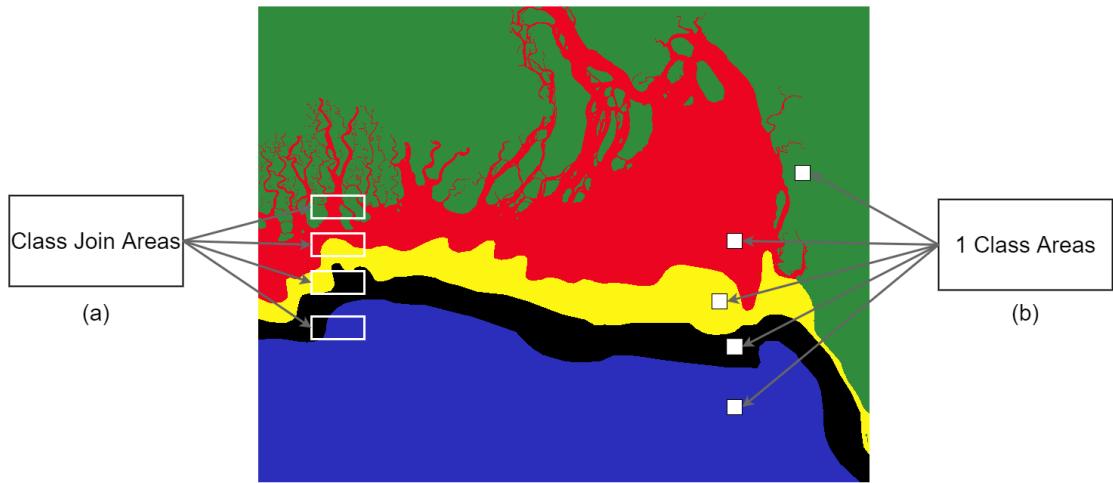


Figure 6: Two Kinds of Regions in the Inputs (a) Class Join Regions. (b) Single Class Regions.

the images and the masks. The size of each image and mask patch is 256x256 pixel which can be used to train image segmentation models.

Chapter 6 Methodology : Modified U-Net

U-Net architecture has two parts, the left part is called contracting part and the right side of the model is called the expansive part. Both these parts together forms a shape like the English letter "U" that is why it has been named U-Net. We used the same form of architecture with modification in the shapes across the model while sub-dividing the model in 11 groups for easier explanation. The activation function used in between the layers is ReLU and the output activation function is SoftMax. The model architecture is shown in the figure below (figure 7).

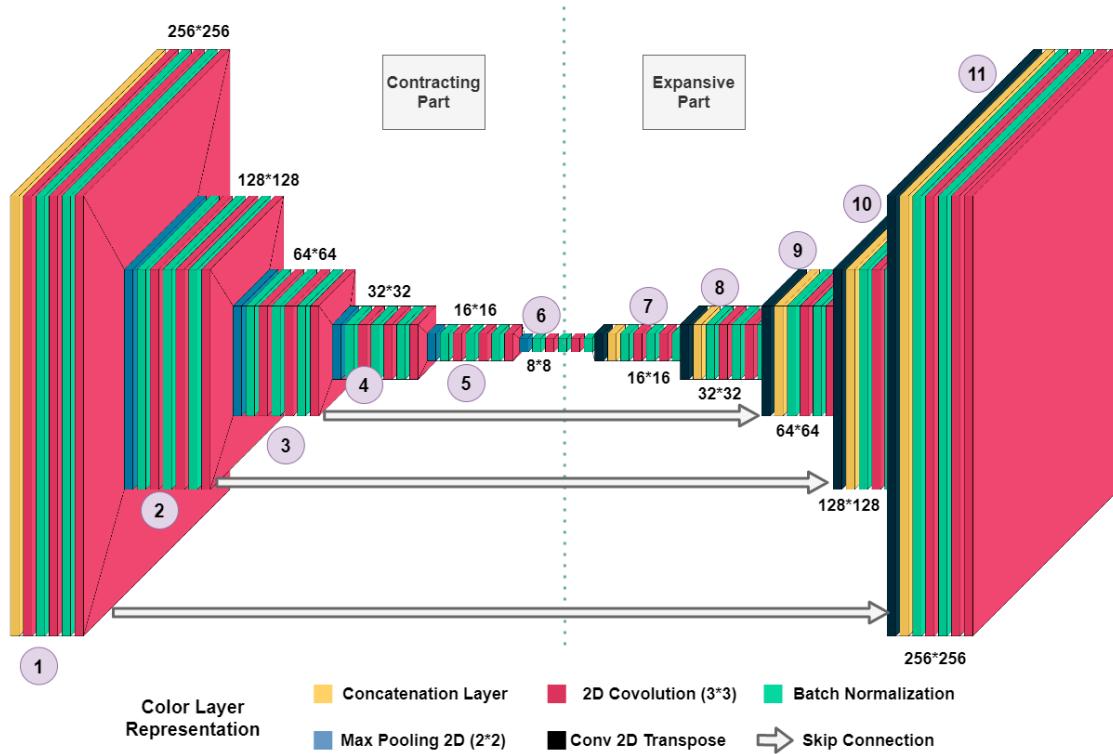


Figure 7: Modified U-Net Architecture Used in this Study.

6.1 Contracting Part

In group 1, The input image of 256*256 first goes through a concatenate layer and then through a layers of unpadded 2d-convolution with 64, (3*3) kernels followed by a batch normalization with 2*2 kernel, stride size of 2 and momentum of 0.01. The next two layers are exactly the same. In this stage, values are then copied or passed to the expansive part of the model via skip connection. Then the image passes through a 2d-max-polling layer with kernel size (2*2) and stride 2 shrinking the image size to 128*128. The arrangement of layers in group 3 is exactly like group 2. Group 4,5 and 6 has the same layer configuration as group 2 expect there is no skip connection in this layers.

6.2 Expansive Part

From group 7, the expansive part of the model starts. A group in expansive part starts with a Conv2d-transpose layer with 64 kernels of size (3*3) and stride of 2. This layer does the exact opposite of 2d-Convolution. This layer is followed by a concatenation layer, then a batch normalization and a 2d-convolution layer for twice. This arrangement is followed till the 11'th group and finally the last layers image has a shape of 256*256*5.

Chapter 7 Experiment setup

7.1 Hardware Requirement

Deep neural network can often be hardware expensive as millions of parameters are learning or updating in each step. Often times, Datasets are also immense. So, hardware specification in most cases are quite substantial. Our hardware specifications have been shown in the table below. (Table 1)

Component	Description
CPU	Intel(R) Core(TM) i7-8565U CPU 1.80GHz - 4.20GHz 4 Core 8 Logical Processor
GPU	NVIDIA GeForce GTX 1050 Max-Q 2GB GDDR5
RAM	16GB 2400MHz DDR4
Operating System	Microsoft Windows 10 Home

Table 1: Hardware Specifications for this Project.

7.2 Software Requirement

The core of our project is Tensorflow (2.4.1). Although Keras (2.4.3) backend have been used for training, Keras backend has been shifted to Tensorflows API. Other common

python package like Numpy(1.19.2), Pandas(1.2.0), pillow(8.1.0), Rasterio(1.2.1), gdal(3.2.1) and glob(2.0.0) has been used for different purpose based on their usability. All matrix representation and manipulation has been done using numpy and pandas while all the raster processing has been handled by rasterion and gdal.

7.3 Hyper Parameter

Most of the hyper parameters have been chosen empirically. The best performing parameters have been finally selected for this project. Adam optimizer was used while training the model with a learning rate of e^{-4} and decay rate of e^{-5} . The original U-Net was built for 1-channel image and we modified it for 3-channel. Based on our output (five class), we had to change the last layer activation function from "Sigmoid" to "softmax" for the last layer of U-Net. "ReLU" activation were used in between the layers. For performance measurement, we used "Dice coefficient" and "Pixel-wise Accuracy" function. We used "categorical cross entropy" for our loss function. This setup was gained by trial and error. The model was trained for 30 epochs while 305 images were used in every step of the training epoch and 93 images for validation. Batch size was set to 32. For evaluation metrics, we used Dice Coefficient and pixel-accuracy score. Both of these matrices measure the similarity between ground-truth and predicted mask.

7.4 Code optimization

The dataset we are using is enormous in size, compared to the dataset, we had computational power that meets the requirements at the margin. For completing the project,

we had to optimize our code to the highest level so that we could run it on our computational setup. Code optimization in this case means batch-wise learning and only loading the data that is currently being used for training. U-Net itself has 1.617 Million parameters in which 1.612 Million parameters are trainable. This huge model itself takes a larger portion of the usable memory. In addition to our RAM, we created another paging file that moves the memory used by python back and forth between the RAM and SSD card during training. Each time only 32 images are loaded on the memory for training and after being used, these 32 images are removed from the current memory and another set of 32 images are loaded.

8.1 SoftMax

The modified U-Net that we're using uses a SoftMax activation function in the last layer of the model. SoftMax activation will give a vectorized probability distribution of a pixel belonging to a particular class [56]. As we are doing a semantic segmentation task, pixel-wise use of SoftMax is ensured by the structure of the U-Net model. SoftMax activation function uses the equation below. (equation 1)

$$\sigma(\vec{z}_i) = \frac{e^{(z_i)}}{\sum_{k=1}^n e^{(z_k)}} \quad (1)$$

Where e represents standard exponential function, z is the input vector and n are the number of classes. σ is the Softmax function that is applied in the input vector according to equation 1.

8.2 Dice Coefficient

Dice coefficient is a performance evaluation metric. It shows the similarity percentage between the true image and predicted image [57]. The equation of dice coefficient is as follows. (equation 2)

$$DiceCoefficient = \frac{2 * intersection(A * B)}{(A + B)} \quad (2)$$

Where A is the true image and B is the predicted image by the model.

8.3 Pixelwise accuracy

As this study deals with the problem of semantic segmentation, pixel accuracy is another important metric. Pixel accuracy determines the percentage of pixel being classified correctly by the segmentation [58]. In other words, pixel accuracy shows the percentage of true positive rate attained by the model. Pixel accuracy follows the equation below. (equation 3)

$$Pixel\ Accuracy = \frac{\sum_n TP_n}{\sum_n (TP_n + FP_n)} \quad (3)$$

Here in equation 3, n represents the number of class and TP means true positive and FP means false positive.

8.4 Categorical Cross-Entropy

Categorical Cross-entropy also known as negative log loss is a modification of binary cross-entropy for multi-class scenario [59]. The equation for calculating negative log loss is as follows. (equation 4)

$$L(y, \hat{y}) = - \sum_{j=0}^M \sum_{i=0}^N (y_{ij} * \log(\hat{y}_{ij})) \quad (4)$$

In equation 4, y is the true class and \hat{y} is the predicted class.

Chapter 9 Result & Analysis

We have analyzed the results and predictions of our model from two perspectives, with and without noisy labels. In satellite image, noise is a common phenomena [60] where on the other hand, labelling by human experts also increases the amount of noise [1] as different human being labels an image from their own confidence (deciding upon the label). We have tested our best performing model that is, Dec2019, under three kinds of label noise.

Despite of some other types of noise, the best image over a region is taken on when the satellite is in its NADIR position [61] (satellite perpendicular to a point on earth). Off-NADIR position creates and angle between the sensor of satellite and the region. Moreover, sunlight also has an angle of reflectance over a region in which satellite image is being taken. So, analysis of models robustness under different level of rotation and flip has been analyzed in this study. On the other hand, salt and paper [62] is another common noise seen in satellite images. So, we also experimented with gaussian noise which represents salt and pepper noise.

9.1 Analysis of Result under noiseless labels

During the model training 70% data has been used for training and the other 30% data has been used for testing. The models were trained for 30 epochs. This criteria has been followed for all four datasets. The results and metric scores has been shown in table 2.

	Dice Coefficient(%)	Validation Dice Coefficient(%)	Loss	Validation Loss	Pixel Accuracy(%)	Validation Pixel Accuracy(%)
Dec-2018	85.51	85.42	0.695	0.667	74.88	73.34
Jan-2019	86.72	86.76	0.627	0.614	77.47	77.07
Dec-2019	86.86	87.81	0.628	0.600	77.00	78.97
Jan-2020	85.76	86.22	0.672	0.661	75.25	75.83

Table 2: Metric, Accuracy and Loss for Four Year Dataset.

Figure 8 shows the change of training and validation dice co-efficient and figure 9 shows the change of training and validation loss over 30 epochs.

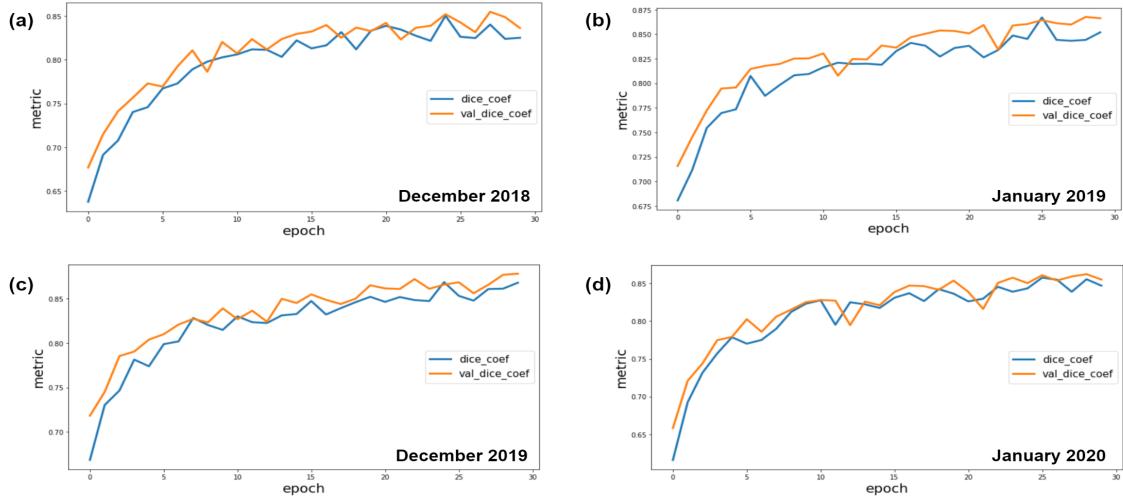


Figure 8: Training and Validation Dice Coefficient of Four Datasets Over 30 Epochs.

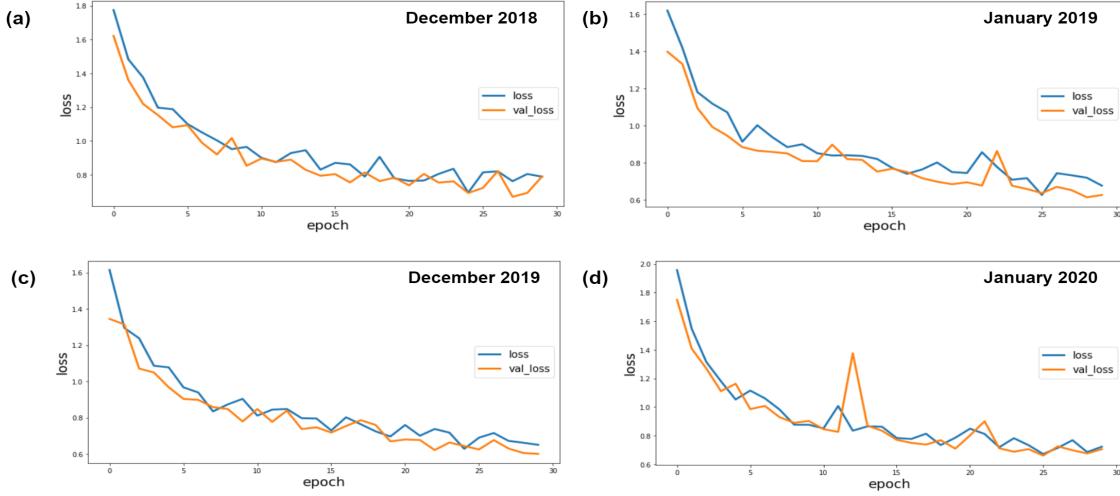


Figure 9: Training and Validation Loss of Four Datasets Over 30 Epochs.

By the pixel accuracy, loss and dice coefficient values and graphs, it appears that the model is learning some pattern in the data as the loss is smoothly decreasing in accordance with the validation loss while on the other hand, same scenario can be seen in the case of dice coefficient. As dice coefficient is a performance evaluation metrics, it seems to perform well with a highest coefficient of 85% both for training and validation.

The model predicts for 5 classes but as already shown, the total image has two kind of regions, one where it has one of the five classes and the other is where two separate class joins. The next two figure below shows the prediction of our model for class join regions (Figure 10) as well as for single class regions (Figure 11).

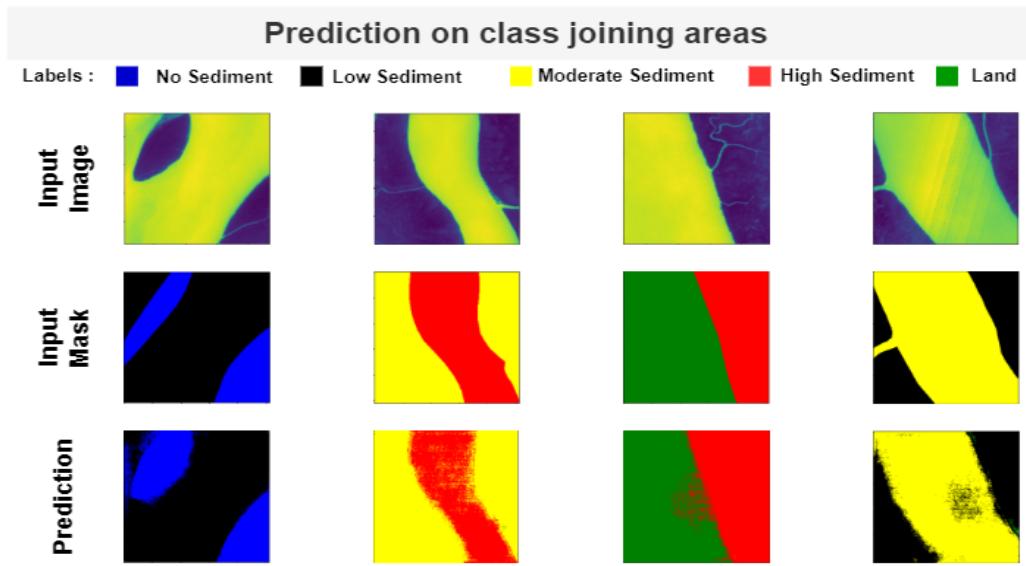


Figure 10: Image and their Corresponding True Mask and Predicted Mask for Class Join Regions.

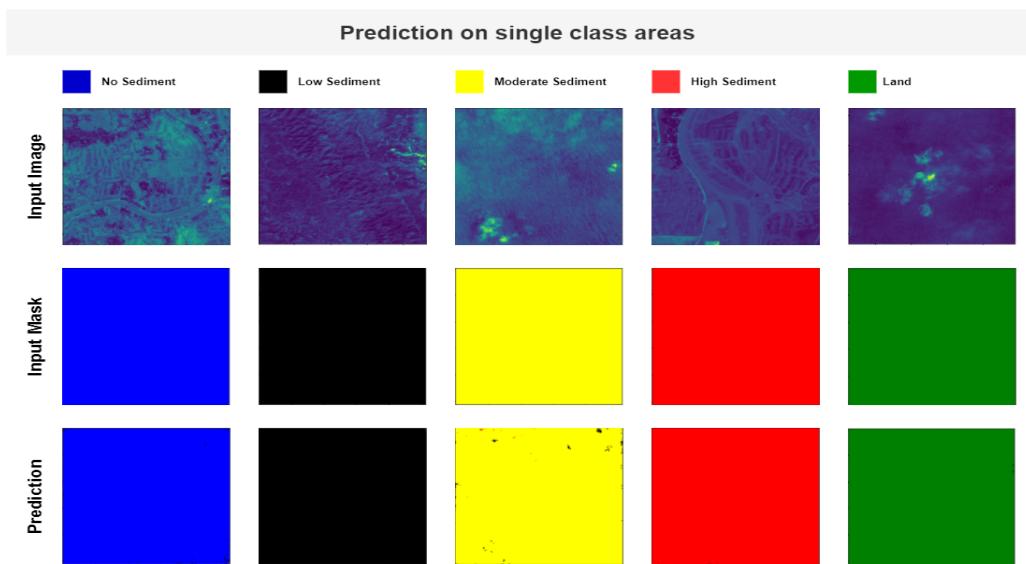


Figure 11: Image and their Corresponding True Mask and Predicted Mask for Single Class Regions.

9.2 Analysis of Result Under Noisy Labels

In the previous section we saw how the models perform for different year dataset under noiseless labels. The same model performed almost similarly for all these datasets, but the Dec-2019 model comparatively performed better than the other datasets. In this section, we will present the result and metric as well as model predictions for our best performing model.

9.2.1 NCAR - Gaussian Random Noise

Gaussian random noise is a NCAR (Noise Completely at Random) type of noise in which noise in the label appears completely randomly and class independently. We pushed random noise in a range of 1% to 25% with gaps in between. Random pixels were chosen based on the noise percentage and a randomly picked class was pushed in that random pixel location. The percentage we chose are 1,3,5,10,15,20 & 25 percent. The figure below shows the gradual increase of Gaussian label noise.

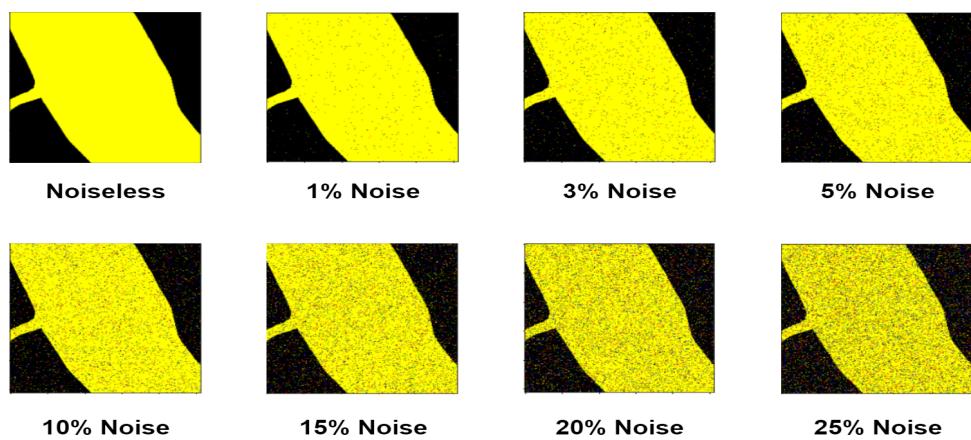


Figure 12: Example of Gaussian Label Noise in a Range of 1% to 25%.

The following figure shows prediction on class join regions under different percentage of Gaussian label noise.

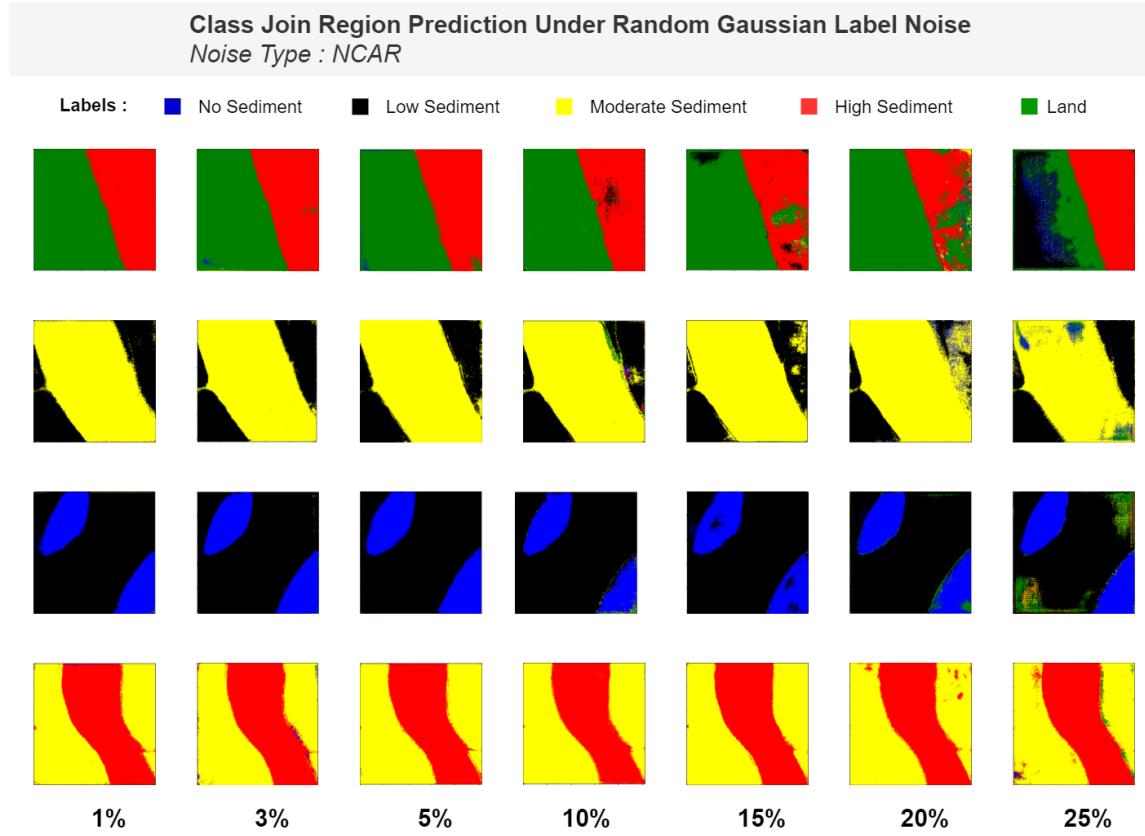


Figure 13: Class-Join region prediction Under Gaussian Label Noise Ranging from 1% to 25%.

The figure shows that the model performs almost similar to noiseless models upto 10% noise. After that, some slight drop is seen for 15% noise. But, as we get closer to highest noise, that is 25%, performance changes drastically. The same characteristic of performance drop is also seen in predictions of single class region areas (Figure 14).

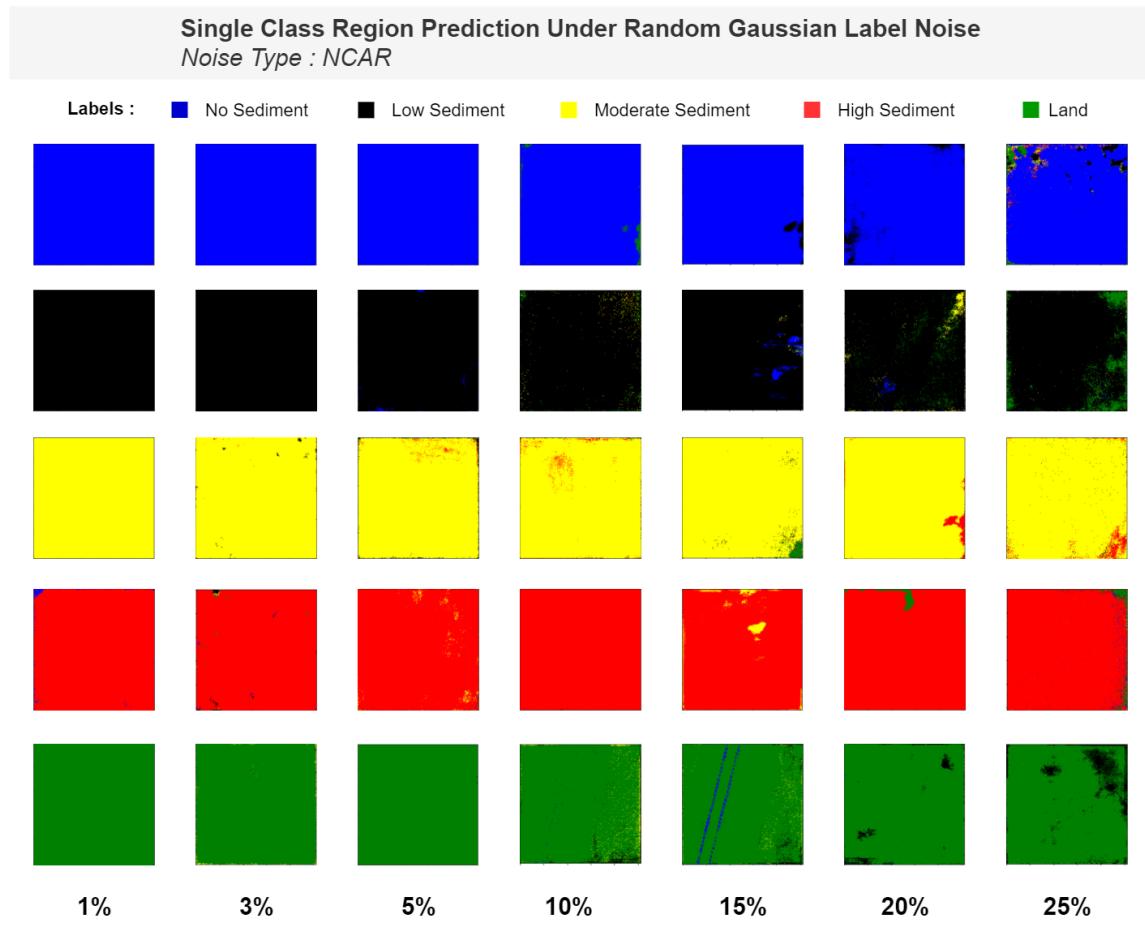


Figure 14: Single Class Region Prediction Under Gaussian Label Noise Ranging from 1% to 25%.

The table below (Table 3) shows the metric and accuracy measurements of the model under different percentage of Gaussian noise.

	Dice Coefficient(%)	Validation Dice Coefficient(%)	Loss	Validation Loss	Pixel Accuracy(%)	Validation Pixel Accuracy(%)
1%	87.51	87.42	0.595	0.607	78.88	78.34
3%	85.02	86.15	0.734	0.703	74.99	76.44
5%	83.94	85.33	0.796	0.752	72.16	75.11
10%	82.85	83.94	0.874	0.843	71.48	73.35
15%	81.14	81.70	0.954	0.952	69.07	69.40
20%	79.18	79.76	1.03	1.02	65.11	65.85
25%	78.65	79.20	1.07	1.05	64.11	65.21

Table 3: Dice Coefficient, Pixel Accuracy and Loss for Gaussian Label Noise.

From the table (Table 3), we can clearly see that, as we increase the percentage of Gaussian label noise, the model's performance seems to drop. Where the dice coefficient for 1% noise was 87.51% and pixel accuracy was 78.88%, for 25% label noise, dice coefficient drops to 78.65% and pixel accuracy drops to 64.11%. Same scenario is seen in case of loss. From 0.595 for 1% Gaussian noise, loss increases to 1.07 for 25% Gaussian noise.

9.2.2 NAR - Rotation with Nearest Fill Mode Label Noise

Rotation is a type of random noise but not completely at random. A image can only be rotated in a certain angle which is not entirely random. Our approach to rotation is slightly different, labels were ranging between 10° to 40° with an interval of 5° and the gap created due to rotation was filled with "Nearest" mode of python SciPy library to avoid artificial pixel filling. The figure below (Figure 15) contains some example of

rotation with nearest fill noise.

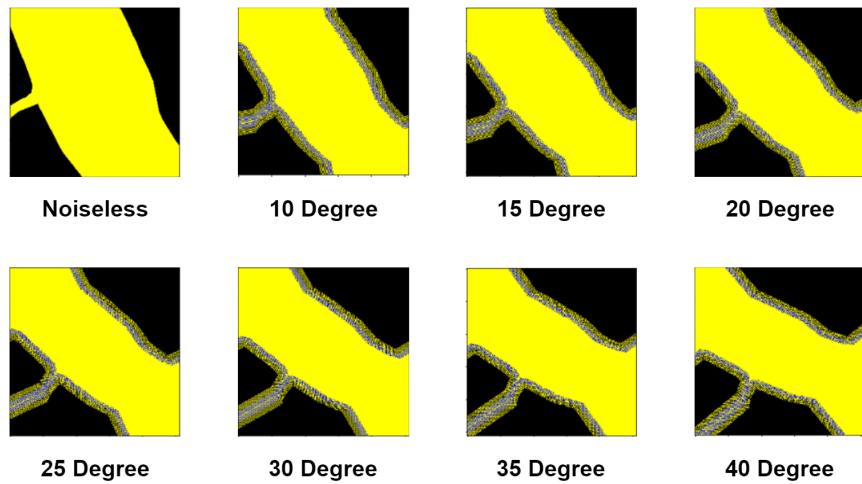


Figure 15: Example of Rotational Label Noise (10° to 40°) with Nearest Fill Mode.

The table below contains the accuracy and metrics of model performance under rotation noise.

	Dice Coefficient(%)	Validation Dice Coefficient(%)	Loss	Validation Loss	Pixel Accuracy(%)	Validation Pixel Accuracy(%)
10°	88.46	87.54	0.562	0.610	80.78	78.21
15°	87.53	87.65	0.616	0.613	79.09	79.01
20°	85.43	86.76	0.705	0.649	74.49	77.14
25°	80.78	81.14	0.717	0.726	72.28	71.84
30°	71.23	73.26	0.958	0.918	67.35	68.16
35°	67.45	66.09	1.598	1.480	60.12	59.14
40°	65.71	64.38	1.687	1.662	59.10	58.45

Table 4: Dice Coefficient, Pixel Accuracy and Loss for Rotational Label Noise.

As we can see from the table, as we increase rotation performance of the model decreases notably. Dice co-efficient of 88.46% for 10° rotation reduces down to 65.71% for 40° rotation. Same type of scenario is seen in the case of validation dice coefficient. On the other hand, loss of 0.562 for 10° rotation increases up-to 1.68 for 40° rotation. These values of evaluation metrics clearly depicts that up-to 25° , the model's performance slightly drops but after that the performance drops significantly. The next two figures (Figure 16,17) show the models prediction capability under different level of rotational noise.

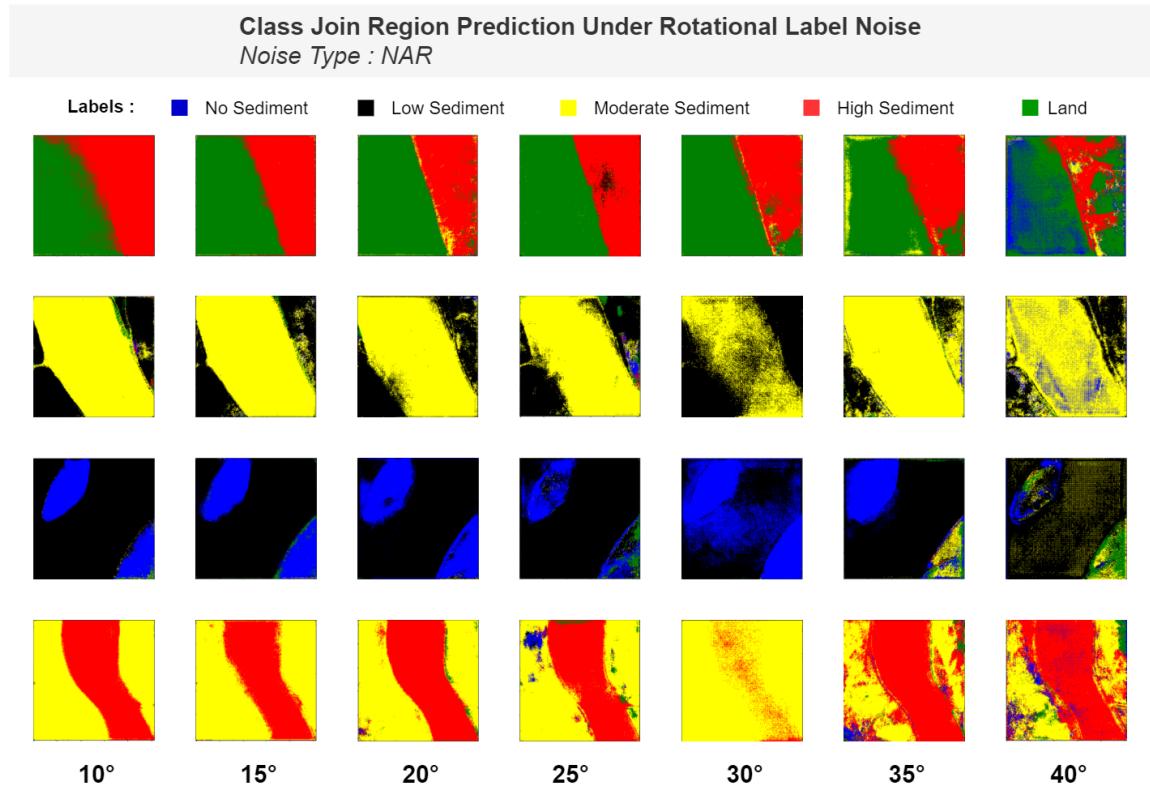


Figure 16: Class-Join Region Prediction Under Rotational Label Noise Ranging from 10° to 40° .

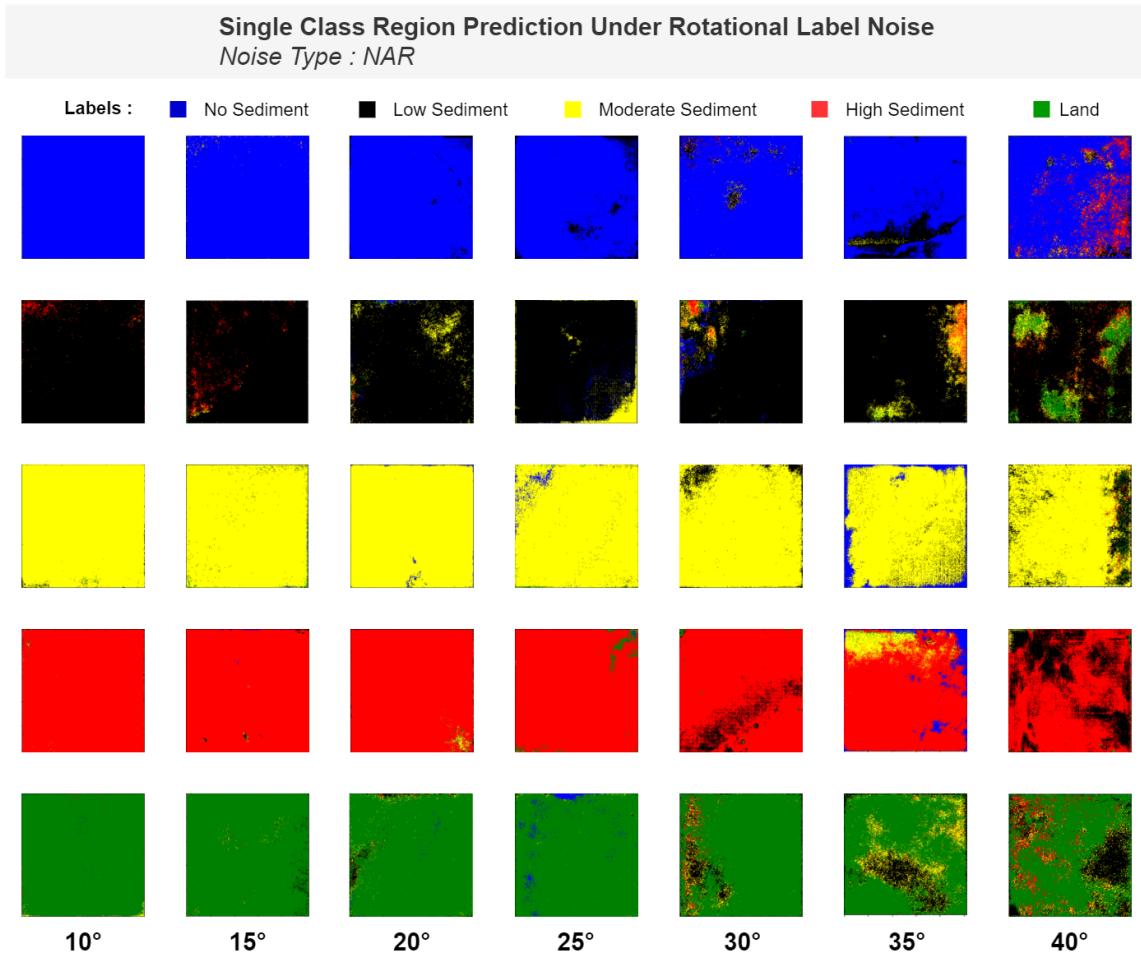


Figure 17: Single Class Region Prediction Under Rotational Noise Ranging from 10° to 40° .

The figures also depicts that upto 25° rotation, the model predicts almost similar to the noiseless model while 30° rotation causes performance drop. From the metric-accuracy table, we can also see that rotation more than 30° brings down both pixel accuracy and dice coefficient drastically. The prediction image also shows the same characteristic as we can clearly see the model guessing randomly both for single class and class join regions.

9.2.3 Label Flip Noise

In label flip noise, we flipped the labels vertically and horizontally. This is a NNAR (noise not at random) type of noise. The figure below shows an example of label flip noise on an mask image.

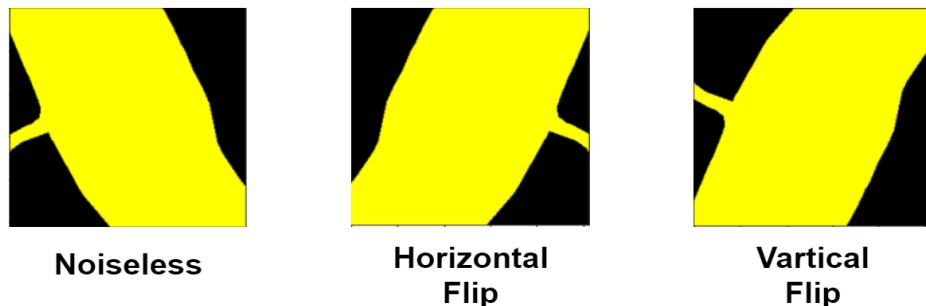


Figure 18: Example of Horizontal and Vertical Flip Label Noise.

The table below shows the pixel accuracy and metrics results under horizontal and vertical label flip noise.

	Dice Coefficient(%)	Validation Dice Coefficient(%)	Loss	Validation Loss	Pixel Accuracy(%)	Validation Pixel Accuracy(%)
Horizontal	65.94	63.83	1.365	1.647	61.50	63.23
Vertical	67.53	64.65	1.416	1.513	63.09	62.01

Table 5: Dice Coefficient, Pixel Accuracy and Loss for Label Flip Noise.

The table shows that, for label flip, U-Net does not perform well for segmentation task. The loss is high and accuracy and dice coefficient is considerably low which leads the model to random guesses for pixels. The next two images show the model's performance under label flip noise for both class join regions and single class regions.

Class Join Region Prediction Under Label Flip Noise Noise Type : NNAR

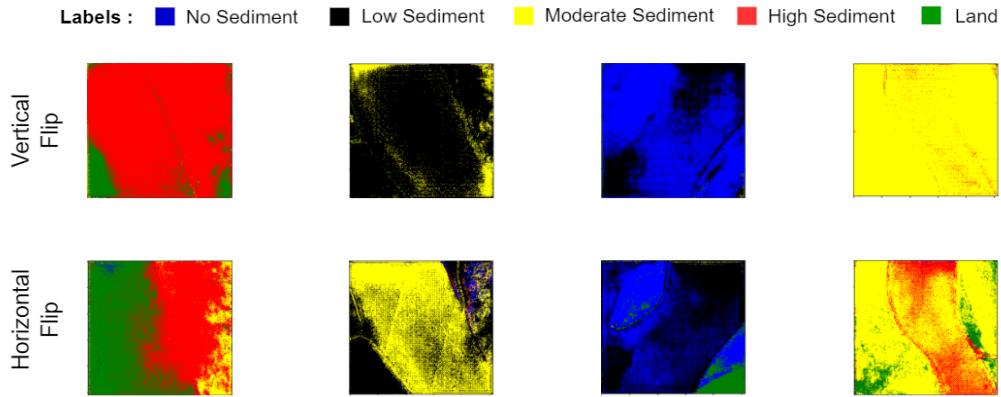


Figure 19: Class-Join Region Prediction Under Horizontal and Vertical Flip Label Noise.

Single Class Region Prediction Under Label Flip Noise Noise Type : NNAR

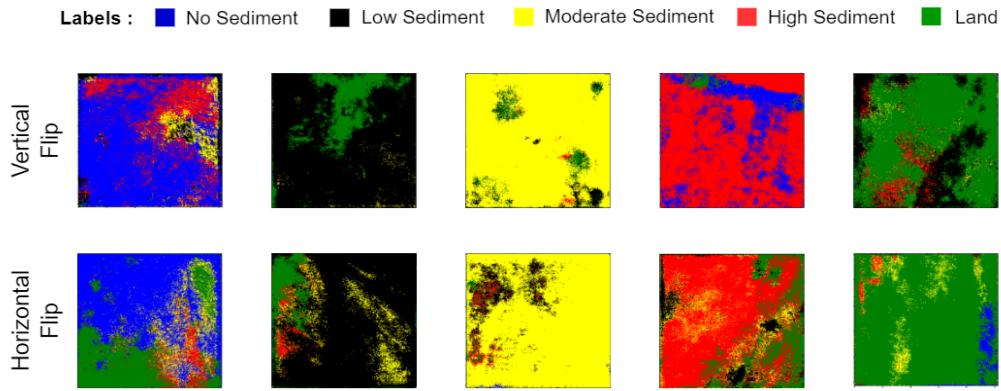


Figure 20: Single-Class Region Prediction Under Horizontal and Vertical Flip Label Noise.

Figure 19 and 20 which shows the prediction of our deep neural model under label flip noise. We can see that under label flip, the models performance is considerably poor. The model for some labels cannot segment properly let along a clean segmentation. For both horizontal and vertical flip, the model seems to loose its segmentation capability.

Chapter 10 Project Impact

This section depicts the impacts of our study in different sectors and analyzes how this impact can save human effort, time and money while solving real-world problems.

10.1 Economic Impact

Funding field researches not only involves the fund, but also needs a lot of other hassles like arranging manpower, travelling and organizing the project for a long time depending on the topic. But, in the recent years, machine learning has been bringing tremendous revolution in almost every sector. Using ML for solving real-world problems does not require a lot of human effort but requires data related to that specific problem. In traditional machine learning problems, data is an issue as open-source data specific to a problem is scarce.

As we have already showed that vast amount of data is being recorded by a ample amount of satellites, it is easier to find data for most of the studies related to geo-spatial analysis. Using remotely sensed satellite data is a cost-effective way to investigate and analyse sedimentation in the marine area without reaching into the study field while wasting valuable time and resource. This process will lessen additional efforts for the

purpose and will require minimal computation power. In our work, we showed the usability and effectiveness of UNet architecture to segment the sedimentation of the Bangladesh bay area into five separate classes, which can assist the legitimate authorities in further investigations.

10.2 Impact on research community

It came to our knowledge while conducting this study that, research with satellite data is still lagging behind in Bangladesh with compared to other countries. But, satellite data is easily collectable by nearly anyone, it is vast in sample size and several filtration can be done to get more accurate data for a effective research. We think that study like ours will create a Chanel for the future researches and will also encourage them to look deep into this type of subject matter.

The backbone of our project is a modified U-Net which was originally built for medical image segmentation for only gray-scale images. Successful modification of most recent techniques for any another use case has always bought significant innovation. Moreover, We think that our project also delivers the message of experimentation with newer algorithms in computer science and solve different social problems by fine tuning those technologies.

U-Net is generic enough to generate segmentation of other use cases. We showed a combination of patch-wise learning, categorical-crossentropy loss, Adam optimization and softmax prediction while measuring the performance with dice co-efficient and pixel-wise accuracy. We investigated the effect of different label noise as noise in satellite imagery is a common scenario. Our model can be further used for predicting the sediment type in the river basins of Bangladesh.

10.3 Social Impact

The land of Bangladesh is full of rivers and canals which eventually ends in the Bay of Bengal creating huge amount of sediment load in the estuaries (where the river's meet the sea) areas. A lot of people loses their home and land due to river erosion, flood and other river-sea related natural disasters. Sedimentation has direct impact on also all of these calamities. So study related to sedimentation becomes a must for our country. Our study can be a future solution to perform early decisions of the relevant authorities. The segmentation of sedimentation in the marine area can also be helpful in the case of a country's sustainable development.

Chapter 11 Conclusion

Sediment load analysis can perform a crucial part in densely populated countries like Bangladesh because there are numerous vulnerable areas here where soil erosion, river erosion, and frequent flood are well-known phenomena which can affect a large number of people. This study was conducted as a part of senior design project by the ECE department of North South University. During the experimentation and analysis we came across some findings and outcomes which might be beneficial to future research on this topic.

11.1 Project Findings

Before getting into the actual implementation, we had to collect data that are usable as satellite imagery are contained with various kinds of noise as well as images are sometimes distorted for technical reasons. In our project, before downloading the images from GEE, we filtered them from a geometrical, radiometrical, sensor and general perspective and all these images were cloud free (presence of cloud less than 1%). Another empirical finding was the model-metric setup. The original U-Net was built for 1-channel image and we modified it for 3-channel. Based on our output (five class), we had to change the last layer activation function from "Sigmoid" to "softmax" for the last layer of U-Net. "ReLU" activation were used in between the layers. For perfor-

mance measurement, we used "Dice coefficient" and "Pixel-wise Accuracy" function. We used "categorical crossentropy" for our loss function. This setup was gained by trial and error. We used other alternatives for all these metrics but the ones performing best, optimizing well, minimizing cost and finally giving similar outcome segmentation were chosen.

Noise is a inseparable part of satellite images data. Understanding noise thus aid the purpose of building robust solutions to specific problems under label noise. Without proper understanding of how deep learning segmentation model can react to different kinds of noises, it is difficult to build precise solutions. In this study, we conducted a in depth analysis of the performance of deep learning model under three prevalent kinds of noise (NCAR - noise completely at random, NAR - noise atrandom and NNAR - noise not at random) that can be present in the labels. For Gaussian random label noise, which is a NCAR type of noise, seven magnitudes of noise percentage was used. As we increase the percentage of Gaussian random noise, Dice Coefficient falls, loss increases and pixel-wise accuracy drops at a equidistant trend compared to the previous value of each other. There is no sudden dramatic change in these values. But, for rotation noise, which is a NAR type of noise the trend is different. Up-to 25° , the model Dice Coefficient stays above 80% and drops in a equidistant trend. But for 30° , rotation we see a sudden downfall of every parameter. More rotation after that shows the same trend. And for NNAR, horizontal or vartical flip of noise shows significant negative impact on

the performance. Dice Coefficient drops in the vicinity of 65% and pixelwise accuracy to 62% which leads the model to random guess.

11.2 Future Work

Using remotely sensed satellite data is a cost-effective way to investigate and analyse sedimentation in the marine area without reaching into the study field while wasting valuable time and resource. For the future work of this study, we have the desire to work with the delineation of sediment in the marine region of Bangladesh as well as in the river banks. For improving the models output, we might try some other approach and other models performance of these datasets. As noise is a common phenomena for satellite images, we also have the desire of de-noising the prediction by using homogenization algorithms. Our implementation has been made generic enough that can aid other related research. For encouraging future researches, our implementation has been made public and can be found in <https://github.com/Tahmid1406/Sediment-Load-Performance-Under-Label-Noise>.

References

- [1] B. Frénay and M. Verleysen, “Classification in the presence of label noise: a survey,” *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.
- [2] J. P. Ericson, C. J. Vörösmarty, S. L. Dingman, L. G. Ward, and M. Meybeck, “Effective sea-level rise and deltas: Causes of change and human dimension implications,” *Global and Planetary Change*, vol. 50, no. 1-2, pp. 63–82, 2006.
- [3] “Bwdb,” *Bangladesh Water Development Board | On Going Project*, Jan. 2021.
- [4] “Cegis,” *Comprehensive Resource Database*, Feb. 2021.
- [5] M. M. Islam and M. Shamsuddoha, “Coastal and marine conservation strategy for bangladesh in the context of achieving blue growth and sustainable development goals (sdgs),” *Environmental science & policy*, vol. 87, pp. 45–54, 2018.
- [6] M. G. Hussain, P. Failler, A. A. Karim, and M. K. Alam, “Major opportunities of blue economy development in bangladesh,” *Journal of the Indian Ocean Region*, vol. 14, no. 1, pp. 88–99, 2018.
- [7] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS journal of photogrammetry and remote sensing*, vol. 152, pp. 166–177, 2019.

- [8] S. S. Chouhan, A. Kaul, and U. P. Singh, “Soft computing approaches for image segmentation: a survey,” *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28483–28537, 2018.
- [9] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.
- [10] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” *International journal of multimedia information retrieval*, vol. 7, no. 2, pp. 87–93, 2018.
- [11] M. Wurm, T. Stark, X. X. Zhu, M. Weigand, and H. Taubenböck, “Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks,” *ISPRS journal of photogrammetry and remote sensing*, vol. 150, pp. 59–69, 2019.
- [12] T. Panboonyuen, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, “Semantic segmentation on remotely sensed images using an enhanced global convolutional network with channel attention and domain specific transfer learning,” *Remote Sensing*, vol. 11, no. 1, p. 83, 2019.
- [13] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, “Semantic segmentation with transfer learning for off-road autonomous driving,” *Sensors*, vol. 19, no. 11, p. 2577, 2019.
- [14] B. Cui, X. Chen, and Y. Lu, “Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection,” *IEEE Access*, vol. 8, pp. 116744–116755, 2020.

- [15] Z. Xu, Z. Shen, Y. Li, L. Xia, H. Wang, S. Li, S. Jiao, and Y. Lei, “Road extraction in mountainous regions from high-resolution images based on dsdnet and terrain optimization,” *Remote Sensing*, vol. 13, no. 1, p. 90, 2021.
- [16] J. Li, J. Cao, M. E. Feyissa, and X. Yang, “Automatic building detection from very high-resolution images using multiscale morphological attribute profiles,” *Remote Sensing Letters*, vol. 11, no. 7, pp. 640–649, 2020.
- [17] S. Talukdar, P. Singha, S. Mahato, S. Pal, Y.-A. Liou, A. Rahman, *et al.*, “Land-use land-cover classification by machine learning classifiers for satellite observations—a review,” *Remote Sensing*, vol. 12, no. 7, p. 1135, 2020.
- [18] C. Yang, C. Zhang, Q. Li, H. Liu, W. Gao, T. Shi, X. Liu, and G. Wu, “Rapid urbanization and policy variation greatly drive ecological quality evolution in guangdong-hong kong-macau greater bay area of china: A remote sensing perspective,” *Eco-logical Indicators*, vol. 115, p. 106373, 2020.
- [19] S. Borra, R. Thanki, and N. Dey, “Satellite image enhancement and analysis,” in *Satellite Image Analysis: Clustering and Classification*, pp. 13–29, Springer, 2019.
- [20] P. Singh and R. Shree, “Analysis and effects of speckle noise in sar images,” in *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall)*, pp. 1–5, IEEE, 2016.
- [21] P. Xiao, Y. Guo, and P. Zhuang, “Removing stripe noise from infrared cloud images via deep convolutional networks,” *IEEE Photonics Journal*, vol. 10, no. 4, pp. 1–14, 2018.

- [22] A. Khamparia and K. M. Singh, “A systematic review on deep learning architectures and applications,” *Expert Systems*, vol. 36, no. 3, p. e12400, 2019.
- [23] W. Wang, D. Liang, Q. Chen, Y. Iwamoto, X.-H. Han, Q. Zhang, H. Hu, L. Lin, and Y.-W. Chen, “Medical image classification using deep learning,” in *Deep Learning in Healthcare*, pp. 33–51, Springer, 2020.
- [24] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, “A review of object detection based on deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 23729–23791, 2020.
- [25] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [26] R. M. Haralick and L. G. Shapiro, “Image segmentation techniques,” *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100–132, 1985.
- [27] S. Naz, H. Majeed, and H. Irshad, “Image segmentation using fuzzy clustering: A survey,” in *2010 6th international conference on emerging technologies (ICET)*, pp. 181–186, IEEE, 2010.
- [28] X.-Y. Wang, T. Wang, and J. Bu, “Color image segmentation using pixel wise support vector machine classification,” *Pattern Recognition*, vol. 44, no. 4, pp. 777–787, 2011.
- [29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

- [30] A. Garcia-Garcia, S. Orts-Escalano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” *arXiv preprint arXiv:1704.06857*, 2017.
- [31] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational intelligence and neuroscience*, vol. 2018, 2018.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [33] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, “Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation,” *Medical Image Analysis*, p. 101693, 2020.
- [34] Y. Liu, Q. Ren, J. Geng, M. Ding, and J. Li, “Efficient patch-wise semantic segmentation for large-scale remote sensing images,” *Sensors*, vol. 18, no. 10, p. 3232, 2018.
- [35] M. Wu, C. Zhang, J. Liu, L. Zhou, and X. Li, “Towards accurate high resolution satellite image semantic segmentation,” *IEEE Access*, vol. 7, pp. 55609–55619, 2019.
- [36] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- [37] L. Wu, Y. Xin, S. Li, T. Wang, P.-A. Heng, and D. Ni, “Cascaded fully convolutional networks for automatic prenatal ultrasound image segmentation,” in *2017 IEEE*

14th International Symposium on Biomedical Imaging (ISBI 2017), pp. 663–666, IEEE, 2017.

- [38] E. Goceri, “Challenges and recent solutions for image segmentation in the era of deep learning,” in *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, IEEE, 2019.
- [39] A. S. M. Belal, “Maritime boundary of bangladesh: Is our sea lost,” *Bangladesh Institute of Peace and Security Studies*, 2012.
- [40] L. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *CoRR*, vol. abs/1706.05587, 2017.
- [41] V. Khryashchev, R. Larionov, A. Ostrovskaya, and A. Semenov, “Modification of u-net neural network in the task of multichannel satellite images segmentation,” in *2019 IEEE East-West Design & Test Symposium (EWDTs)*, pp. 1–4, IEEE, 2019.
- [42] Q. Li, P. Yuan, X. Liu, and H. Zhou, “Street tree segmentation from mobile laser scanning data,” *International Journal of Remote Sensing*, vol. 41, no. 18, pp. 7145–7162, 2020.
- [43] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, “Tomato leaf segmentation algorithms for mobile phone applications using deep learning,” *Computers and Electronics in Agriculture*, vol. 178, p. 105788, 2020.
- [44] H. Yoon, S. Park, and J. Yoo, “Real-time hair segmentation using mobile-unet. *electronics* 2021, 10, 99,” 2021.

- [45] A. Nurhadiyatna and S. Lončarić, “Semantic image segmentation for pedestrian detection,” in *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, pp. 153–158, IEEE, 2017.
- [46] Z. Chu, T. Tian, R. Feng, and L. Wang, “Sea-land segmentation with res-unet and fully connected crf,” in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 3840–3843, IEEE, 2019.
- [47] D. Angluin and P. Laird, “Learning from noisy examples,” *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.
- [48] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, “Learning with noisy labels.,” in *NIPS*, vol. 26, pp. 1196–1204, 2013.
- [49] G. Algan and I. Ulusoy, “Image classification with deep learning in the presence of noisy labels: A survey,” *Knowledge-Based Systems*, vol. 215, p. 106771, 2021.
- [50] G. Algan and I. Ulusoy, “Label noise types and their effects on deep learning,” *arXiv preprint arXiv:2003.10471*, 2020.
- [51] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, “Deep learning is robust to massive label noise,” *arXiv preprint arXiv:1705.10694*, 2017.
- [52] S. Boonprong, C. Cao, W. Chen, X. Ni, M. Xu, and B. K. Acharya, “The classification of noise-afflicted remotely sensed data using three machine-learning techniques: effect of different levels and types of noise on accuracy,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 7, p. 274, 2018.

- [53] Marineregions.org, “Marine Gazetteer Placedetails.” <https://www.marineregions.org/gazetteer.php?p=details&id=25431>, 2020. Online; accessed 05 January 2020.
- [54] H. Ahmad, “Bangladesh coastal zone management status and future trends,” *Journal of Coastal Zone Management*, vol. 22, no. 1, pp. 1–7, 2019.
- [55] M. Habibullah-Al-Mamun, M. K. Ahmed, M. S. Islam, M. Tokumura, and S. Masunaga, “Occurrence, distribution and possible sources of polychlorinated biphenyls (pcbs) in the surface water from the bay of bengal coast of bangladesh,” *Ecotoxicology and environmental safety*, vol. 167, pp. 450–458, 2019.
- [56] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv preprint arXiv:1811.03378*, 2018.
- [57] N. Tustison and J. Gee, “Introducing dice, jaccard, and other label overlap measures to itk,” *Insight J*, vol. 2, 2009.
- [58] Z. Wang, E. Wang, and Y. Zhu, “Image segmentation evaluation: a survey of methods,” *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5637–5674, 2020.
- [59] Y. Ho and S. Wookey, “The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling,” *IEEE Access*, vol. 8, pp. 4806–4813, 2019.
- [60] A. Asokan and J. Anitha, “Adaptive cuckoo search based optimal bilateral filtering for denoising of satellite images,” *ISA transactions*, vol. 100, pp. 308–321, 2020.

- [61] Q. Zhou, L. Tian, J. Li, H. Wu, and Q. Zeng, “Radiometric cross-calibration of large-view-angle satellite sensors using global searching to reduce brdf influence,” *IEEE Transactions on Geoscience and Remote Sensing*, 2020.
- [62] M. Lopes, P.-L. Frison, M. Crowson, E. Warren-Thomas, B. Hariyadi, W. D. Kartika, F. Agus, K. C. Hamer, L. Stringer, J. K. Hill, *et al.*, “Improving the accuracy of land cover classification in cloud persistent areas using optical and radar satellite image time series,” *Methods in Ecology and Evolution*, vol. 11, no. 4, pp. 532–541, 2020.