Urban Pattern Recognition from Multi-spectral Satellite Images and Flood Prediction Using Machine Learning Models

by

Rifah Tasnia
20101452
Sorder Md Farhan Fuad
20301058

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Department of Computer Science and Engineering
Brac University
May 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student’ s Full Name & Signature:

Rifah Tasnia	Sorder Md Farhan Fuad
20101452	20301058

Approval

The thesis/project titled “Urban Pattern Recognition from Multi-spectral Satellite Images and Flood Prediction Using Machine Learning Models” submitted by

1. Rifah Tasnia(20101452)
2. Sorder Md Farhan Fuad(20301058)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 21, 2024.

Examining Committee:

Supervisor:
(Member)

Md. Saiful Islam
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Rafeed Rahman
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Bangladesh suffers from the cumulative effects of floods brought on by water flashing from nearby hills, the accumulation of the inflow of water from upstream catchments, and locally heavy rainfall made worse by drainage congestion because it is located in such a basin and is less than 5 meters above mean sea level. Additionally, the rapid landscape changes in river areas brought on by the strong water flow make them more vulnerable to flooding. The purpose of this study is to create a system of detection of flooding in a given area using satellite-collected Multi-Spectral satellite imagery and numerical data collected from Bangladesh Water Development Board. It can then forecast if a flood will occur in the area shortly based on the landscape presented and its current shape. Additionally, it can show the likelihood of a flood as well as whether there is a chance of one. Five classification algorithms, VGG19, GoogleLeNet, UNET, ResNet, and Inception which represent various machine learning concepts, have been chosen and implemented on a free and open-source basis on the image datasets and MLP was used on the numerical dataset and output of the models was feed forwarded to an FCNN model to detect the likelihood of a flood. The multi-spectral image datasets and numerical datasets used for this study' s foundation date from 2015 to 2023.

Keywords: Flood prediction, Flood inundation, multi-spectral satellite image, Sentinel-1, Machine Learning, VGG19, GoogleLeNet, UNET, ResNet, Inception, MLP, Deep Neural Network, FCNN.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Considering the high rate of climate change, floods are one of the most frequent and destructive natural disasters. These have the potential to result in fatalities as well as extensive property and infrastructure destruction. Especially in low-lying countries, the rate of flood is higher than the rest. Researchers have been trying to identify the patterns of these floods for decades. However, in the past few decades, there has been a growing interest in using satellite imagery to detect floods. Satellites can provide a wide range of information about the Earth's surface, including the breadth of floods, the depth of water, and the rate of the flow. This information can be used to rapidly and properly assess the impact of a flood which will also aid in disaster relief operations. There are a variety of different algorithms and methods that can be used to detect floods using satellite imagery. Among all these supervised and unsupervised methods, the people working with climate change and studying the behaviors of natural disasters have been trying to find the best methods to identify the waves of floods. One common method is to utilize change detection.[17] This involves comparing images of the same area taken before and after a flood to identify areas that have been inundated. They use satellite images of different timelines when the floods hit the area and also of when it was dry season. Comparing these images gives an overview of the highest dangerous point for a flood to hit as well. Another method is to use feature extraction .[14] This entails spotting characteristics of flooding in satellite photos, such as adjustments to the vegetation or the appearance of water bodies. In our research, we have included two different types of data, image and tabular, we processed those data into two different sets of models, ResNet, VGG19, Unet, GoogLEnet, and Inception for Image and MLP for tabular data, and then processed the outputs of those data into and FCNN feed forwarding model to compare the best outputs. These outputs have given us the best algorithm that can be used to identify patterns of urban areas and flood-prone areas. We have concluded our research with the best outcome possible with the help of satellite images for flood detection.

1.2 Importance

Bangladesh is a country with an economy that depends upon agriculture, based in the tropical monsoon zone hence vulnerable to natural disasters such as cyclones and floods, identifiable through satellite imagery and can thereby be mitigated to some extent. Satellite images go beyond borders; instead, they capture the Earth's surface, and its properties and they are accounts such as glaciers, urban areas, and others that include the risk flood. This information is necessary for people dwelling on such territories to be aware of the imminent danger and evacuate in safety. In addition, future floods will be predicted by observing the movement of flood waters from space photos. These particulars are necessary for disaster mitigation planning as well as ensuring that floods catch no one unawares. Satellite images can be used for determining the level at which flooding takes place, and also in identifying and tracking floods.[10] The details are equally useful for coordinating relief activities or evaluating the level of destruction caused. Satellite images can also be used to monitor floods in real time. In general, it is possible to follow the course of floods using satellite images, and places at risk of being flooded are also easy to identify. Flood identification and monitoring are some of the main uses of satellite photos. By knowing this people can save their lives and property through averting and mitigating the effects of floods.

1.3 Motivation

Around the world, urban areas are expanding quickly, increasing the need for effective and precise urban planning techniques. Therefore, employing effective urban planning techniques is essential for Bangladesh's as well as the world' s sustainable growth and disaster relief. Effective catastrophe management and responses got wasted. If that flood had been predicted, the damage could have been lessened in a large number. However, the existing methods that were used to predict floods often had limitations such as accuracy, timeliness, and scalability.
Accurately monitoring the Flood detection using multi-spectrum satellite images has become crucial to predict floods. This model will mention the problems of inadequate ineffective flood prediction models. The current methods used in flood detection, often rely on manual interpretation or simplistic or classic techniques.[11] Approaching this problem with a more automated model will give a better understanding and accuracy rate of monitoring Flood detections including buildings, water bodies, roads, and such from multi-spectrum satellite images. On the same note, the current flood prediction models usually use a limited set of input features and classic algorithms. This oftentimes cannot capture the complex interaction between Flood detections and flood dynamics. More advanced models such as this can help to effectively incorporate multi-spectrum satellite imagery data, Flood detection information, and hydrological variables to give a more accurate and specific prediction. This model will use more advanced methods to calculate the risks in different areas. Multiple models are used to extract the workable matrix to work with a MLP generated matrix that would be feed forwarded to a FCNN model for final results on detection.

1.4 Problem Statement

There is a rapid growth in the urban area which indicates the growth rate of flood intensity. There have been massive floods in recent years. Many suffered unimaginable losses in that flood. Many people died, lost their homes, and agro-foods got wasted. If that flood had been predicted, the damage could have been lessened in a large number. However, the existing methods that were used to predict floods often had limitations such as accuracy, timeliness, and scalability.
Accurately monitoring the Flood detection using multi-spectrum satellite images has become crucial to predict floods. This model will mention the problems of inadequate Flood detection recognition as well as ineffective flood prediction models. The current methods used in Flood detection recognition, often rely on manual interpretation or simplistic or classic techniques.[11] Approaching this problem with a more automated model will give a better understanding and accuracy rate of monitoring Flood detections including buildings, water bodies, roads, and such from multi-spectrum satellite images. On the same note, the current flood prediction models usually use a limited set of input features and classic algorithms. This oftentimes cannot capture the complex interaction between Flood detections and flood dynamics. More advanced models such as this can help to effectively incorporate multi-spectrum satellite imagery data, Flood detection information, and hydrological variables to give a more accurate and specific prediction. This model will use more advanced methods to calculate the risks in different areas.

1.5 Research Objectives

Many countries in this world have faced unbearable losses due to natural calamities such as floods. If only we could have predicted the area which was going to be flooded, we could have reduced the damage. Our country Bangladesh is agriculture-based. We depend on a large amount of income through our agro-foods. In our country climate change is a regular thing and during heavy rain and floods we lose a large amount of agro-food every year. For this reason, flood prediction has become a necessity. The main goal of this research is to accurately predict weather conditions and develop a multi-featured flood prediction model. These are the following objectives of our research:

- Monitoring the rainfall of an area, measuring the temperature, wind speed, and humidity will help us to predict the flood of that area.
- Gathering a huge and accurate amount of data sets in a small area. We can get a great outcome from a small area rather than a large area for flood prediction.
- Using some advanced CNN-based deep learning algorithms, e.g. ResNet, VGG19, Unet, GoogLEnet, and Inception etc[2] to find the possible model that works best with the FCNN model in detection.
- Utilizing MLP model with the tabular data, by mapping and filtering .[12], [21] Further generating a matrix to feed forward to the FCNN Model.

- Creating an FCNN model that takes matrices as inputs which is output from the deep learning model and MLP model and the processes those to detect flood.

- This research' s main focus is to predict the flood accurately and in the shortest time possible.

They depend on accurate flood forecasts, yet conventional flood prediction models are frequently constrained by inadequate or erroneous data.
Multi-spectral satellite imagery has been an effective technique for gathering comprehensive data about Bangladesh's urban environment. Machine learning models have recently made strides, making it possible to automatically evaluate this imagery and extract useful patterns and attributes.[14], [18] These methods help us better understand how urban features are distributed spatially in Bangladesh[21], which can help us develop more sensible urban planning methods. Since Bangladesh has a low-lying terrain and monsoon climate, urban planning as well as flood prediction has become a critical issue in Bangladesh. To address the preceding shortcoming, a model will be shown for the planning of Flood detections and flood detection using multi-spectrum satellite images.
With innovative satellite imagery and tabular data, we can predict floods better through a more all-rounded comprehension of the flood dynamics. Also, we will design original feature engineering techniques that will enable us to derive meaningful ideas and predictions from both satellite and tabular data thus improving predictive accuracy. Come up with a combined machine learning model that is ideal for incorporating multiple aspects of data, and that can depend on multi-modal learning or attention mechanisms. Suggest new assessment and confirmation measures that are unusual to flood forecast activities by taking into account spatial links and temporal changes. Work on a system for forecasting floods in real-time by keeping an eye on the image as well as tabular information streams simultaneously with the potential to make decisions on time and mitigations. Designing a unique machine learning model optimized for both data sources, which can potentially use multi-modal learning or attention mechanisms. Presenting new validation and performance evaluation metrics specifically designed for flood prediction tasks that take into account spatial dependencies and temporal dynamics. Development of the real-time flood forecasting framework that continuously watches the flows of image data and tables enabling timely decision-making processes and disaster response.
The goal of our research is to develop a model that will understand the Flood detections from multi-spectrum satellite images and predict floods more accurately than its predecessor in terms of predicting floods. To achieve that, we will collect data from a comparatively small region and use it to develop several hybrid machine-learning techniques. Sequentially, we will compare and evaluate these results and choose the best one. As a result of Bangladesh's subtropical monsoon climate, which is characterized by significant seasonal changes, research on predicting floods from urban images has not been done before which has motivated us to undertake this careful research topic.

Chapter 2

Literature Review

From the study Urban Matanuska Flood Prediction using Deep Learning with Sentinel-2 Images (Chellapa, R. S. R. et al. 2021)[15], the author suggested a way for foretelling urban floods in the Matanuska area utilizing deep learning methods and Sentinel-2 satellite images. The authors discuss the requirement for precise flood prediction models in flood-prone metropolitan regions. They use the Sentinel-2 satellite's high-resolution multispectral photos to identify pertinent information and develop deep-learning models for flood prediction. The approach, including data preprocessing methods, deep learning architecture, model training, and evaluation procedures, is presented in full in the paper. The Sentinel-2 photos are analyzed and flood predictions are made using convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Using the proper measures, they assess the performance of the suggested models and compare them to baseline models and currently used flood prediction techniques. This paper is relevant to our topic because it demonstrates how deep learning methods and satellite imagery can be used to enhance flood prediction skills, assisting with urban planning and disaster management initiatives. A unique strategy for overcoming the difficulties of predicting urban floods in similar geographic situations is provided by the utilization of Sentinel-2 imagery and the deployment of deep-learning models.

In the research "Urban Land Use and Land Cover Classification Using Novel Deep Learning Models Based on High Spatial Resolution Satellite Imagery" , the authors (Zhang P et al. 2018) [12] have mentioned different kinds of methods to identify patterns in worldview 2 and worldview 3 satellite sensors. They have used a few methods such as SVM prediction, CNN prediction, U-Net prediction, ASPP-Unet prediction, and ResASPP-Unet prediction. They have identified the differences between such methods and their significance. They have shown all the layers to implement such methods and get the best results for urban pattern recognition. The images they collected from the satellite were trained and tested from WV2 and WV3 over the city of Beijing. Their strategy showed the best results with ASPP-Unet prediction and ResASPP-Unet prediction for dry weather in that area. The variety of intermediate feature maps (IFMs) and layer depth have a substantial impact on classification accuracy, according to the study's investigation into the effects of model parameters. The 64 IFMs and 11-layer depth are found to be the ideal specifications. While increasing these factors can sometimes improve efficiency, it also makes computing more difficult. Accuracy also depends on the input picture bands chosen, with models developed using 8-band imagery outperforming those developed using 4-band or 3-band data. Furthermore, using NIR imagery instead of solely RGB imagery boosts accuracy. The suggested models provide a practical method for classifying urban land coverage. Nevertheless, they require a sizable number of exceptional ground truth samples for training, which in big cities tends to be labor-intensive. However, once trained, such models can be quickly used to map land cover in other urban regions that have high-resolution datasets that are comparable. This makes it easier to observe urban changes effectively and offers accurate data for development plans. Future projects involving the categorization of urban land cover could benefit from the development of weak supervision networks.

Research has further revealed the difference between supervised classification methods and unsupervised classification methods. Such as, in "Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning" (Tanim. A. H., McRae C. B., Tavakol-Davani H., Goharian. E. 2022)[20], the authors did a study that combines a combination of ground-based observations and satellite imaging to enhance rapid flood detection in metropolitan areas. To identify flooded regions in the City of San Diego, machine learning models, including supervised and unsupervised methods, are built and trained. The outcome of the evaluation demonstrates that the modification to the detection-based unsupervised flood detection method works better than competing models while using fewer data and computational resources. Their methodical approach could help other flood-prone cities such as ours and improve planning for urban transportation and infrastructure in order to reduce flood hazards.

The main purpose of this study on "Automatic Flood Detection from Satellite Images Using Deep Learning" (Çalışkan, Ö. B. 2022)[18] is to identify the area which has been affected by floods and other natural disasters. They have used deep learning algorithms and satellite images for this research. They have also used Synthetic Aperture Radar (SAR) for high-resolution earth images and moving target detection. Every step of the model has been described to the point in this paper. They have used sentinel-1 satellite images which operate day and night at both times. This satellite provides the image regardless of any weather condition. To analyze and edit the image they have used the SNAP (Sentinel Application Platform) tool. They obtained the satellite images from ESA Copernicus (European Space Agency). By using every step properly and running every algorithm, and training they got the perfect output for their model. However, they had some lacking such as making this model for finding flood disaster places but our goal is to make a model for flood prediction. By collecting the previous dataset, and measuring the soil condition we can predict the flood of a certain area and clear the area before it happens or prevent that by taking the necessary steps.

This study on "Performance Evaluation of Machine Learning Algorithms for Urban Pattern Recognition from Multi-spectral Satellite Images" (Wieland, M., and Pittore, M. 2013)[5] presents a framework for classifying urban patterns in multi-spectral satellite images. They used WorldView-2, Quickbird, and Ikonos for high-resolution images. And Landsat ETM, TM, and MSS for medium-resolution images. The framework uses machine learning algorithms to identify different types of urban land use, such as residential, commercial, and industrial areas. The objective of the study is to investigate the potential of machine learning algorithms for urban pattern detection and to evaluate how well they function in various scenarios. We chose four classification methods including Normal Bayes, K Nearest Neighbors, Random Trees, and Support Vector Machines to put them into practice. The study determined that satellite picture type and image scene influenced machine learning algorithm effectiveness. Classification accuracy was affected by data variations in quantity, categories, and sizes present in the feature vector, and picture segmentation impact. Finally, this survey provides important knowledge concerning how machine learning algorithms can be applied in urban pattern recognition among various issues. These findings from this survey indicate that the use of machine learning algorithms could serve as a useful instrument in urban planning and disaster management.

This study [13] aims to tackle the urgent issue of identifying floods via images taken from space, underscoring the mounting disaster incidences across the globe owing to floods as well as their effects on socio-economic stabilities. In the area where artificial intelligence can be applied, this article introduces the Sen1Floods11 dataset, which consists of 11 flood events worldwide that are represented as labeled pieces in the form of Sentinel-1 remote sensing images [13], which depend on machine learning using convolutional neural networks (CNNs) as a basis. This entails asking four research questions, intended to enhance flood detection efforts and checking if CNNs can be operationalized in creating maps globally. They carried out a study on several permanent water data types and flood events before processing images or training several CNN models that would seek to explain elements of flood detection. To rate how well models perform reviewers use metrics such as intersection over union (IOU) as well as error rates. In general, this research paper presents a holistic method for flood detection by employing sophisticated machine-learning techniques that in turn strengthen disaster response and management practices. In this research [13], they used the FCNN model. We have also used the FCNN model in our paper. We saw that they have trained FCNN on optical data vs. radar data, hand-labeled data vs. weakly supervised data, and trained on permanent water only vs. flood and permanent water. We trained our data and then sent it to the FCNN for the final result.

The research provided [16] described how microwave signals can be used to identify likely areas submerged by water in case of flooding as well as predicting how deep these places may submerge using machine learning technologies. This project was carried out in the Kosi River region of Bihar, India where floods are a common disaster using satellite sensor data like Synthetic Aperture Radar images from Sentinel-1 satellite together with Digital Elevation Models (DEMs) which contain information on relief profiles in any given location [16]. With such a model hybridization, this ensures the binary conversions method as well as some supervised machine learning classifiers e.g. the Random Forest together with the K Nearest Neighbor help in realizing the correct classification of regions submerged or not by floods and the same tool can also be used to predict how deep the waters could go when superimposing the extent of a flood upon DEM data. There is good agreement with previous flood mapping studies, as a benchmark. In essence, it offers a way of systematically tracking floods and mobilizing rescuers in real time making a point of how vital it is to have holistic models in the analysis process. In this paper, they used a Random Forest Classifier (RFC) and KNN (K Nearest Neighbour) model. We did not use these models because in our paper we are not trying to look for the best features but the important ones. And for KNN that model is an old model for data processing. We tried to use some newer and advanced models for our research.

Chapter 3

Proposed Methodology

3.1 Overview

We start our process by acquiring the Sentinel-1 satellite imagery dataset and water level, rainfall tabular dataset. To acquire and pre-process the Sentinel-1 data, we used google earth engine. In the earth engine code, we first defined ROI or Region of Interest which in our case was Barisal, Chittagong, Dhaka, Khulna, Rajshahi, Rangpur, Sylhet. After that, sentinal-1 images were loaded and filtered with Vertical-Horizontal (VH) polarization and images were acquired in Interferometric Wide (IW) mode. Images were selected for both ascending and descending mode of the satellite. The VH band that was selected before collection, created a mosaic by combining all images of the collection into one single image. This mosaic was clipped into our ROI boundary. Then a function named RefinedLee was implemented to do Speckle Filtering by reducing noises in our SAR imagery. For both before and after image there was a conversion done from decibels to natural units and then again the filtered image was converted back to decibels from natural unit. Then the calculation and detection of flood area was done by creating a flood mask where the before filter was greater than -20 and the after filer was less than -20. A water mask was created for the identification of permanent water body. Then before image, after image, water area and flood area were visualized. These pre-processed data were exported for each region and all the years from 2015-2023. They were exported as 4 different images of before image, after image, water area and flood area into TIF format. They were then stacked on top of each other where the before image and after image worked as the background and water area and flood area were mapped in consequently RGB blue and red color. Then they were exported as jpeg image for further work. On each of those composite images, five different machine learning model were applied; they are ResNet, VGG19, Inception, GoogleLeNet, UNet. They extracted the features from all of those images to create 7x7 matrix for each of them as output.

On the other hand, the tabular data of water level and rainfall acquired from the Hydrology department of Bangladesh Water Development Board was mapped and filtered. Since they had a lot of districts, all the districts were mapped into the regions/divisions. The year and months were filtered for the months that showed flood images. Then Multilayer Perceptron was applied to extract the features from them and generate a 7x7 matrix as output.

For each of the matrix of image feature extraction, a feed forwarding model, Fully

Convolutional Neural Network was applied with another input from MLP matrix. They gave us a result of the effectiveness of those models. The precision, recall, F-1 score and accuracy was calculated as result for each of them.

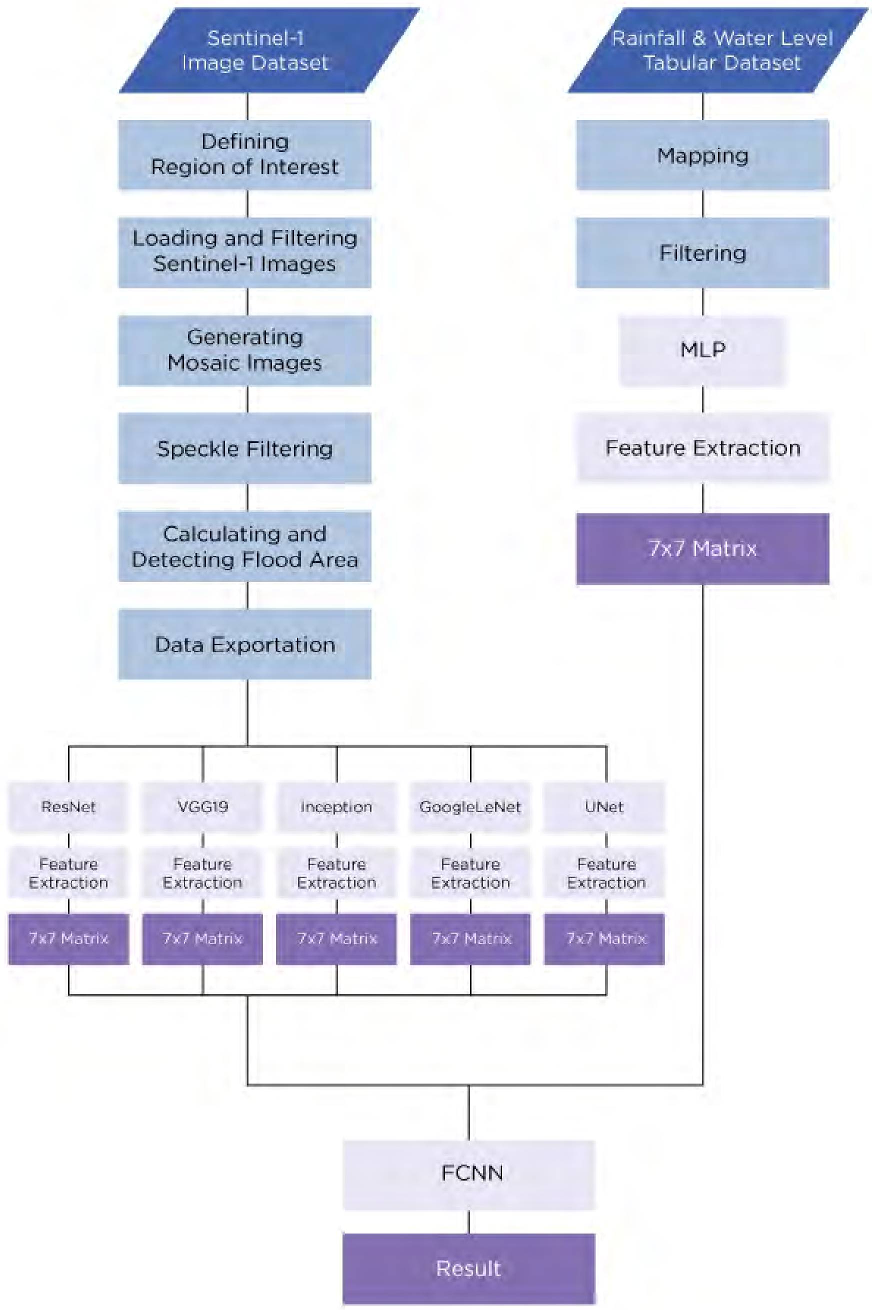

Figure 3.1: The flow chart of the proposed model

Chapter 4

Data Analysis

.

4.1 Dataset Collection

We have collected both image data and tabular data for floods in Bangladesh for 2015-2023.
We had also collected 65 years of climate data from Kaggle [10] for Bangladesh. The official website of the Bangladesh Meteorological Department (BMD) [28] has these figures. The year data range from 1949 to 2013. For each month of the year, the data set contains information on maximum and minimum temperature, precipitation, minimum temperature, wind speed, cloud cover, and solar radiation In addition, it displays information about station numbers. X and Y coordinates, latitude and longitude. As flood data has been collected from many sources including books, Google, newspapers, and internet sites, new standards for annual or monthly flooding have emerged. However, the image data was not available for Sentinel-1 for year 1949-2013.
There were datas from 1972 onwards of satellites such as Landsat 1-3, Landsat 4-5, Landsat 7 etc. but they capture optical images. Sentinal-1 captures radar data which is better for flood detecion. The reason is, optical data can not pierce through cloud coverage and with high rainfall during the flood months, it would not have been able to capture any land images. On the other hand, to capture datas from microwave region, Sentinel-1 uses Synthetic Aparture Radar or (SAR) to collect the data that can pierece through the clouds to capture land and water images.

We are using both the image data and the numeric data because we wanted to make this a multi-modal model. We took the image dataset from the Sentinel-1 satellite to see the affected areas. To understand how vast the flood was. From the tabular data, we could have predicted the flood but with the image data, we can predict the data accurately and also identify the places where the flood happened before and take precautions early.

4.1.1 Image Data Collection

We have collected our image data from Sentinel-1. In the Google Earth engine, there was data available from 2014(from October) to today date. That is why we took our image data from 2015 to 2023. We labeled and pre-processed our data on the Earth engine.

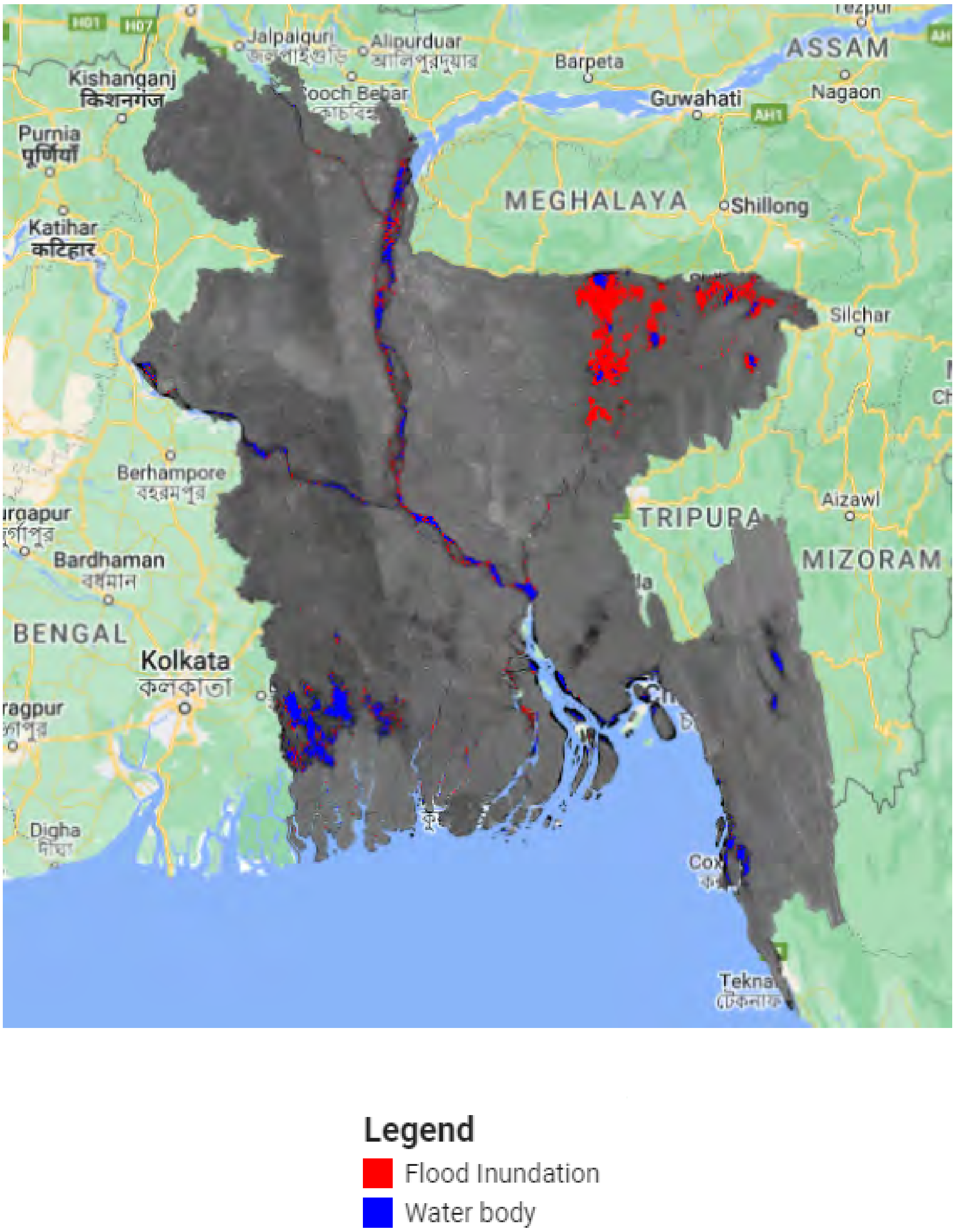

Figure 4.1: Flood Inundation Raw Image of 2022

4.1.2 Tabular Data Collection

We have also collected our rainfall dataset and water level dataset as tabular data from the Hydrology Department of the Bangladesh Water Development Board from 2014 to 2023. These are private data that is not easily accessible to the general public. The current water level metric system is SOB. However, a few years ago, the metric system was PWD. The difference between these are 0.85m. We collected them all for the SOB metric system.

Table 4.1: Monthly Water Level

District_Name	Year	Month	Monthly_Max WL(mMSL)	Monthly_Min WL(mMSL)	Monthly_Avg WL(mMSL)
Kurigram	2015	8	23.41	20.28	21.66
Bogura	2019	7	17.49	14.71	16.05
Sylhet	2022	6	8.35	7.37	7.84

Table 4.2: Monthly Rainfall

District	YEAR	MAY	JUN	JUL	AUG	SEP	OCT	Yearly RF Total(mm)
Rangpur	2015	536	1109.4	0	548	831.5	16	3431.4
Chattogram	2019	187.6	150.4	964.9	457.9	352.7	165.2	2510.3
Sylhet	2022	758	1381	605	208	400	187	3923

.

4.2 Data Pre-processing

We have taken our satellite images from Sentinel-1 (SAR) that needed some steps to pre-process. Also the tabular data needed a bit of pre-processing as well. Descriptions are given below:

4.2.1 Image Data Pre-processing

Defining Region of Interest

At first, we imported a feature collection representing administrative boundaries for Bangladesh (BGD_adm2). Then we filtered the feature collection to select the administrative areas such as Barisal, Chittagong, Dhaka, Khulna, Rajshahi, Rangpur, Sylhet and stored it in ROI. This defines the area of interest for flood analysis. In the satellite image, the administrative areas were the divisions of Bangladesh. However, the Mymensingh division was inside the Dhaka division and there was no separate image data for the Mymensingh division.

Loading and Filtering Sentinel-1 Images

Then we took the Copernicus Sentinel-1 image collection and filtered it according to our needs. We selected images with Vertical-Horizontal (VH) polarization and images acquired in interferometric Wide (IW) mode and also selected images from ascending and descending modes [16].

Two date ranges are defined here to represent the before and after potential flood events. Both before and after collections are further filtered to include only pixels within the Region of Interest or ROI.

Generating Mosaic Images

We selected the VH band from the before collection which creates a mosaic by combining all images in the collection into a single image. Then clip the mosaic to the ROI boundaries. Accordingly, after image collection was created for the images acquired after the potential flood event.
.

Speckle Filtering

Then we have speckle filtering. A function named RefinedLee is defined to implement the Refined Lee Filter which reduces noise (speckle) in SAR imagery.

First, we applied the RefinedLee filter on the before image. Then we convert the image from decibels (db) to natural units. And finally, we convert the filtered image back to db from natural units. As for the after image, we applied the same filtering process.
.

Calculating and Detecting Flood Area

Now comes an essential part which is flood detection and area calculation. Here a threshold was applied to identify potential flood areas. We created a flood mask where values before filtered are greater than -20 and values after filtered are less than -20. Then a binary flood mask was created where only pixels classified as flooded are retained. A water mask was created to identify the permanent water bodies based on the same criteria applied to both before and after filtered images. We added various layers to the map for visualization. After image, before image, before filtered, after filtered. These were added for comparing pre and post-filtered images. Flood and water masks were visualized on the map with distinct colors—red for flood inundation and blue for permanent water bodies.

We used the geometric function to calculate the total area of the ROI in hectares and printed it to the console. After that, we multiplied the flood mask with the image representing pixel area to count for the area of each pixel.
.

Pre-Processed Data Exportation

At the very end, we exported these images into our Google Drive folder as .tif format and then downloaded them for further use. For each region of each year, 4 separate images were exported as before_image, after_image, water_area, flood_area.

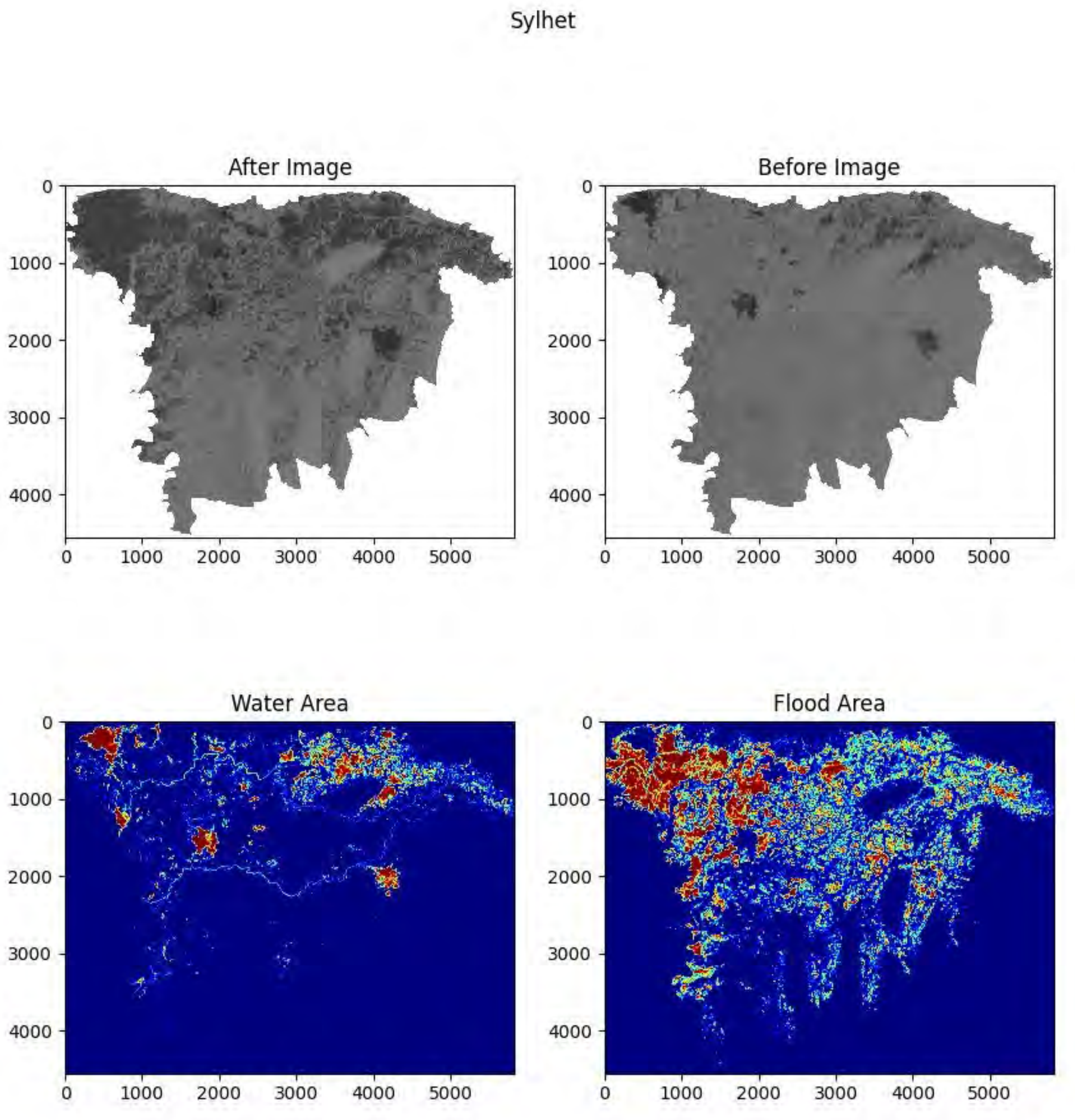

Figure 4.2: after_image, before_image, water_area, flood_area of Sylhet 2022

From these exported images, we scaled the image datas to the range [0,1] that was adjusted by the scaling factor. The scaling factor was set to 1.5 for better contrast of the images. Then we generated a composite image by setting the before_image and after_image as the background and stacking the water_area and flood_area on top of them. Here, the water_area was set to blue with the RGB value [0, 0, 1] and flood_area was set to red with the RGB value [1, 0, 0] for identifying the water and flood areas better. Then we used "matplotlib" to visualize our composite image and saved it as that. We generated the composite images for each region by looping through them and then generating them for each year from 2015-2023.

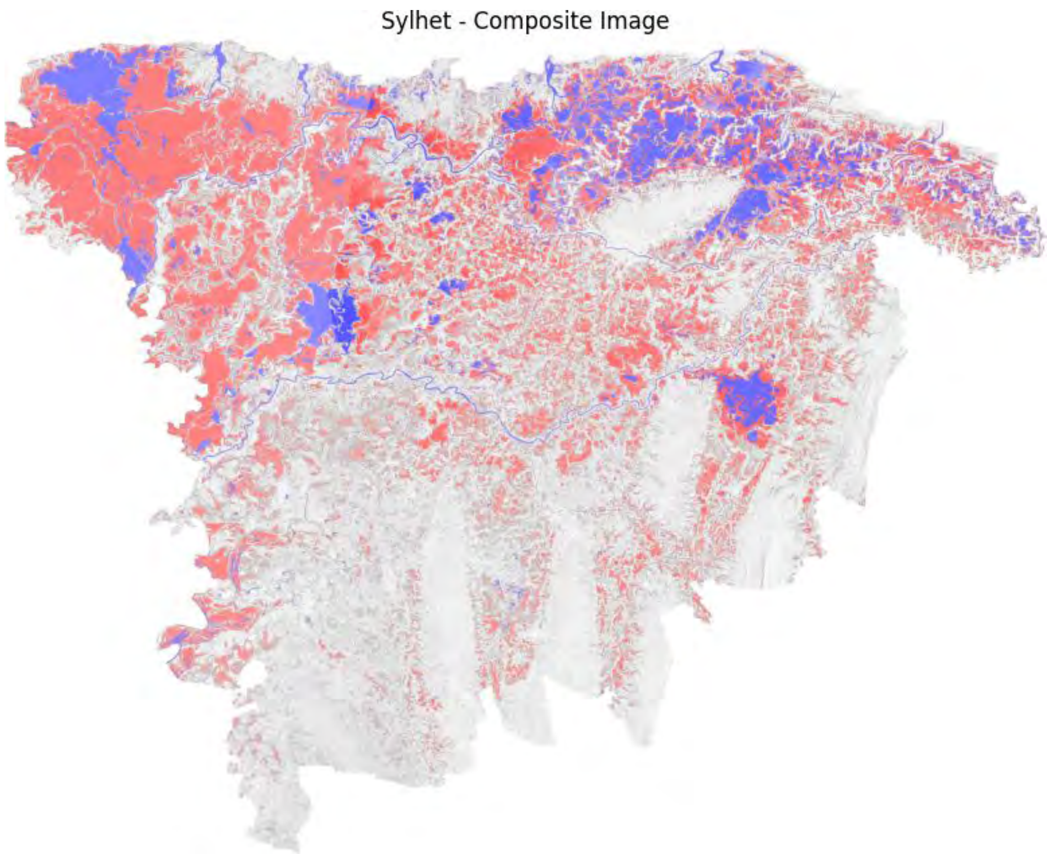

Figure 4.3: Composite image of Sylhet 2022

4.2.2 Tabular Data Pre-processing

We had 6924 data for our Water Level data and 740 data for our Rainfall data. These data were for different stations that takes measurements of Rainfall and Water Level. So for Water Level we had 61 districts and for Rainfall we had 71 districts. We had to map these places and filter them according to the region that we used for our image data.

.

Mapping

We mapped both the places of Water Level data and Rainfall data to their corresponding region data. The district name in water level was mapped to the division or regions and also the district in rainfall was mapped to the division or regions as well. The mapping was done like below:

For Water Level mapping,

"Bogura": "Rajshahi",
"Moulvi Bazar": "Sylhet"

and so on.

For Rainfall mapping,

"Bandarban": "Chittagong",
"Barguna": "Barishal",

and so on.

.

Filtering

We filtered the months and years of our tabular data. We had sentinel-1 images for 2015-2023. But we had tabular data of water level and rainfall for 2014-2023. So we filtered these relevant years. Also, the images gave flood inundation results for a few specific months. These months were the ones when floods occurred throughout these years. So those months were the most relevant ones. That is why we filtered the month's data as well to prepare it for our model.

4.2.3 Feed Forwarding Data

From the output 7x7 matrix of those five models that we used to extract feature from the image and the output 7x7 matrix of the MLP model that was used for tabular data's feature extraction, we took those outputs as input for out feed forwarding model, FCNN. On those data we did preprocessing pipelining with data augmentation, We also flattened the feature and applied PCA to reduce dimensionality [29]. Then we used batch normalization and dropout layers to enhance our FCNN model. After that we applied the FCNN model to get the flood classification analysis.

Chapter 5

Predictive Models and Algorithms

In this section, we applied different machine learning models. First, we applied ResNet18, VGG19, Inception, GoogLeNet, UNet to extract feature from our image dataset and create a 7x7 matrix as an output. We useed 60 epoches and the learning parameter of 0.001 for our models. We also applied Multilayer Perceptron (MLP) model to extract feature from both of our tabular data and create a 7x7 matrix as an output. Then we took both of the outputs of each image feature extraction model and MLP as input and applied a feed forwarding model, Fully Convolutional Neural Network to train and test. We also used adam optimizer for our models. We found which of the image extraction model worked better with the MLP to give better accuracy during testing.

5.1 Image Feature Extraction Models

We applied five machine learning models; ResNet, VGG19, Inception, GoogLeNet, UNet to etract features from our image dataset of Sentinel-1 which gave us 7x7 matrix for all of them.

5.1.1 ResNet

ResNet, a model commonly utilized, has been established upon CNN architecture. An alternate version of ResNet architecture known as Resnet18 contains 18 total convolution layers. It makes use of both forward propagation and backward propagation in computing prediction outputs. One of the problems resolved by this architecture is achieving VGG16' s performance through stacking more and more other layers on top of it. For example, a tightly packed model formed from stacking several layers is capable of capturing the peculiarities of complex data distributions has been illustrated in the model' s prediction results. As an input to the network is passed through its layers linearly, a low-value layer that can cause inaccuracies is created when the first layer is superimposed on the second layer. This low-value layer is created by connecting a ResNet Model to a Residual Network bypassing connection, thus allowing for efficient stacking of layers without losing accuracy [9]. This network input passed linear manner between convolution layers. The stacking of these layers hence enables for creation of the low-value layer which in turn hinders accuracy. By using this way we can bypass the creation of low-value layers inside our model. The ResNet Model avoids this trap by employing a connection called a

Residual Network skip connection; a connection that links the convolutional layers such that any low-value data layers are avoided, consequently, layers can be effectively stacked without losing any accuracy. The remaining structure is utilized to increase the general accuracy of the design.

If we take a subnetwork of certain stacked layers and if its performance function is H(x), then the residual function would be [9]:

$$F(x) = h(x) - x \tag{5.1}$$

The output of the subnetwork would be [9]:

$$y = F(x) + x \tag{5.2}$$

As the model also backpropagates, the outcome function for that is similar to the subnetwork forward propagation result function. The photo input size for the network is 256 x 256.

In our model, we used ResNet18 model from the repository of pytorch [26]. For the image, initially, it verifies if there are contents in the specified directory(location). Then it lists the contents of the directory if the directory is non-empty. It later defines a class named FeatureExtractor that can be plugged into the ResNet model at level 4 to obtain its output while it is run forward [22]. After that, it loads the ResNet-18 model that has been pre-trained on ImageNet and attaches the layer 'layer4' to acquire feature maps. Then it defines a preprocessing pipeline that resizes and crops images to get them in tensor form, after which it normalizes such pictures for feeding the ResNet-18 model. It retrieves the image paths and labels from subfolders within a specific directory. Then it stores those in arrays. The data loader then splits the dataset into 80-20 consequently training and testing data. From there it pulls out the stored weights which come as numpy arrays. Both the extracted features and labels are stored in numpy arrays for both testing and training sets. Finally, it goes ahead to define a function that shows the feature maps extracted and uses it to visualize features in images for both training and testing sets.

We used this model for feature extraction from satellite images. Its deep architecture with residual connections helps capture intricate patterns and spatial dependencies, contributing to the model's understanding of flood-related features.

Figure 5.1: Visualization for Dhaka 2018 using ResNet

5.1.2 VGG19

VGG is a deep neural network that is used to classify images, this deep network is also a Convolutional Neural Network (CNN). Furthermore, this network consumes images of fixed size represented as 224x224 RGB colored images which results in a matrix input of size 224x224x3. Additionally, there' s only one operation that is carried out before feeding image data into the network and this entails subtraction of the average whole training set pixel by pixel mean RGB value [4]. In VGG, convolution is done using 3x3 pixel kernels with a stride of 1 pixel to capture small details in an image, and at the same time, spatial padding is employed for maintaining spatial resolution [4]. Pooling is done in 2x2 pixel windows by max pooling with a 2-pixel stride to lower the size of spatial dimensions and keep important features. The ReLU (Rectified Linear Unit) activation function is used to impose non-linearity, improving its effectiveness in classifications over older models with tanh or sigmoid functions, thus enhancing computational efficiency. In this network, we have three fully connected layers: the first two layers contain 4096 units each while the last one has 1000 channels meant for 1000-way classification during the ILSVRC competition. In the end, the final layer employs the softmax function to generate probability distributions towards the 1000 categories. This systematic approach illustrates the main building blocks of the VGG network and explains its architecture as well as why it works very well in handling image-related assumptions.

In our model, we used VGG19 model from the repository of pytorch [27]. It checks if the specified directory exists and lists its contents. Custom class 'FeatureExtractor' hooks into a specified layer of a neural network model to capture its output during the forward pass [19]. Then loads the inception model pre-trained on ImageNet and hooks the 'Mixed_7c' layer to extract the feature maps. Then it defines a series of transformations to resize, crop, normalize, and convert the images into tensors for input into the model. Then it collects image paths and labels and splits the dataset into training and testing sets. After that, it extracts the feature maps from the hooked layer for both training and testing images, converting the features to numpy arrays. Then it stores the extracted features and labels in numpy arrays for both training and testing sets. Defines a function to visualize the feature maps and uses it to visualize features for images in both training and testing sets. This model was used for image tasks, such as categorizing different types of land cover or identifying water bodies from satellite imagery to extract the features.

Figure 5.2: Visualization for Sylhet 2020 using VGG19

5.1.3 Inception

Inception V3 consists of both symmetric and asymmetric building blocks. These include convolutions, average pooling, max pooling, concatenations, dropouts, as well as fully connected layers. Batch normalization is used generously in the model and has been applied to activation inputs [8]. Softmax is used for loss computation. The model uses factorized convolutions and aggressive regularization to find better ways of scaling up networks.

Contrary to what other inception models require, Inception V3 expects input tensors to have a size of N x 3 x299 x 299 hence you must resize your images accordingly [8]; thus if you want to use it well then resize images properly else it will not work with them at all since without proper sizes those images cannot be used in Inception V3 model which has been designed such that all the available computer resources are used to make sure that image recognition task are well executed and completed swiftly in it. Both architectural innovation and strict compliance with standards enabled it to reach high-performance levels for big datasets such as ImageNet. It is worth noting that Inception v3 is an image recognition model with an exceptional accuracy that surpasses 78.1% image net dataset accuracy.

For our model, we used the InceptionV3 model from the repository of pytorch [25].Our model checks if the specified directory exists and lists its contents. Custom class 'FeatureExtractor' hooks into a specified layer of a neural network model to capture its output during the forward pass. Then loads the inception model pre-trained on ImageNet and hooks the 'Mixed_7c' layer to extract the feature maps. Then it defines a series of transformations to resize, crop, normalize, and convert the images into tensors for input into the model. Then it collects image paths and labels and splits the dataset into training and testing sets. After that, it extracts the feature maps from the hooked layer for both training and testing images, converting the features to numpy arrays. Then it stores the extracted features and labels in numpy arrays for both training and testing sets. Defines a function to visualize the feature maps and uses it to visualize features for images in both training and testing sets.

We also utilized this model for feature extraction tasks. Their efficient convolutional operations at multiple scales can help capture contextual information from satellite images, aiding in flood feature detection and classification.

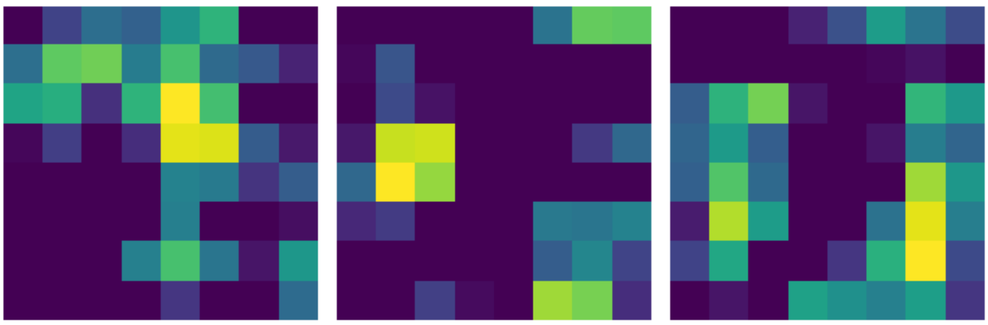

Figure 5.3: Visualization for Chittagong 2019 using Inception

5.1.4 GoogLeNet

GoogLeNet runs on a different structure from the likes of AlexNet and ZF-Net with increased depth made possible through various techniques like 1 by 1 convolutions being introduced and global average pooling. While designing inception architecture they decided to incorporate 1×1 convolutions instead to cut down on the overall amount of parameters (which are weights plus biases) accompanying each layer. If the parameters are reduced, the depth of the architecture is increased as well. At the end of the network in GoogLeNet architecture, there is a method known as global average pooling which is employed. It averages a feature map of 7×7 to 1×1 in this layer. Inception module stands out from other architectures like AlexNet and ZF-Net. In every layer, fixed convolution size prevails. 1×1, 3×3, and 5×5 convolutions alongside 3×3 max-pooling for input are run concurrently in the Inception module [8]. This results in an output that is then produced by stacking all of them up together before ending up at some final point. The middle part of Inception architecture has certain classifier branches that only come into play in the course of training.

The above paragraph describes a few concept blocks of a complete model. The 5-layer architecture consists of a single convolutional block having 128 filters, followed by two fully connected layers of 1024 and 1000 units, respectively, finally ending up with a category prediction layer that applies the softmax function to output class probabilities. A loss of those blocks contributes to the overall loss calculation using its value multiplied by 0.3.

For our model, we used the GoogLeNet model from the repository of pytorch [24].Firstly, in our model, it ensures that the directory specified as image_data_dir is present and lists what it contains in case it is found. For forward pass capture, implement a class FeatureExtractor that interfaces through a specific layer of GoogLeNetinception5bmodel. On ImageNet, deploy the GoogLeNet model and attach the 'inception5b' layer for extracting feature maps [8]. A preprocessing pipeline is a set of processes that are utilized to resize, crop, be converter as a tensor, and then normalize in readiness for inputting into the GoogLeNet model. Collects image paths and labels from subdirectories within a particular directory and keeps them within lists. Separate the dataset into training data and tests using an 80-20 split. Feature maps are then extracted from the hooked layer in training and testing images, and stored as numpy arrays for both testing and training sets..numpy arrays are used to store both labels and the extracted features for both testing sets and training sets. Finally, it goes ahead to define a function that shows the feature maps extracted and uses it to visualize features in images for both training and testing sets.

This model is similar to Inception models, GoogLeNet was used for feature extraction. Its multiple parallel convolutional pathways allow for capturing diverse features from satellite images, which can enhance the model's ability to identify flood-related patterns. Like all the model, below is an example of the 7x7 matrix output that was generated by this model. This is the visualization of the said matrix.

Figure 5.4: Visualization for Barisal 2016 using GoogLeNet

5.1.5 UNet

The UNet is a famous design meant for semantic labeling in computer vision, which attempts to predict pixel-level labels leveraging both local and global contexts. It comprises two major components; the constricted path and the expansive path. In the contracting path, two 3x3 convolutions are applied to each block respectively, without padding. Then comes the ReLU activation function which also serves to decrease spatial dimensions. Also, there is a 2 x 2 max pooling that divides the feature map size by two in particular instances where the channel numbers are increased by twice because more complex features can be captured by the training network [7]. We may begin with a broad approach, which involves upsampling the feature map so that the resolution is increased. A 2x2 "up-convolution" reduces the feature channels by half. Additionally, the feature map obtained from the up-convolution is merged with one that has been cropped from the path also known as contraction to retain all fine-grained high-resolution information within the resulting map. Upon concatenation, we follow with two operations that consist of convolutions having dimensions of 3×3 followed by ReLU activation functions to further improve upon it.
For our model, we used the Unet model from the repository of pytorch [23]. First, in our model, it checks to see if the folder you specified exists and returns a list of what it has. A custom PyTorch module that can tap into a specific layer in any model to extract feature maps during its operations. This function is responsible for attaching forward hooks on our selected layer that will help us store all its outputs at runtime before they get overwritten or destroyed by any further passes over these same layers. It then readies itself up so that we can call upon these things whenever required, after this we run it through our models forward pass method and collect all the features of interest. Load an already trained UNet network with ResNet34 serving as the encoder. Specify the layer where you want to extract features from. This code defines a chain of operations that are applied to images before they can be fed to a model. It uses resizing, center cropping, converting to tensor, and normalizing with ImageNet statistics as some of its steps [7]. Then, it collects paths of images and assigns them tags based on folder structure. After this, it splits data into training and test sets using splitting it using 'train_test_split' . It' s also designed to extract features from an intermediate layer, followed by their processing within a certain function called feature extractor. It will process each image in the training set and extract features from it as well as corresponding labels. It will then process each image in the testing set. For verification purposes, it prints out the

shapes of feature arrays from training data and those from test data. Additionally, using matplotlib library, extracted feature maps are visualized. After that, it loops over every image found in the training and testing set so that it can extract features and visualize them.
We used this model for image segmentation tasks related to flood mapping from satellite images. It excels in precisely delineating flooded areas or other relevant features from satellite imagery, providing valuable input for flood prediction models.

Figure 5.5: Visualization for Rangpur 2017 using UNet

5.2 Tabular Data Feature Extraction Models

Here, we took inputs from our tabular dataset (Water Level and Rainfall) and generated 7x7 matrix after applying Multilayer Perceptron Model.

5.2.1 Multilayer Perceptron

A Multilayer Perceptron like MLP is an advanced and feedforward network peculiar to artificial neural systems. It consists of several fully connected layers in such a way that each layer uses a nonlinear activation function. As part of classification algorithms, they are based on many different features with varying significance levels like colors or shapes. [1]

The initial layer of the MLP gets input attributes. In this tier, an attribute from the dataset is represented by each neuron. An input layer that leads to one or more output layers. To do this, the cells in these layers assign weights to their input cells which they add up, and then apply some non-linear activation function at the end to get an output. Various non-linear activation functions like ReLU, Sigmoid, and Tanh help the network in understanding sophisticated connections [1]. The final layer makes final predictions on various tasks. In classification, these output neurons normally adopt softmax functions which give class probabilities. Conversely in regression, linear functions can be applied instead. Every neuron found within a single layer is linked to all available neurons from the next such layer. Such dense connections permit complex data patterns captured with ease. Since MLPs utilize supervised learning methodology, they are easy to teach. Backpropagation is the most common method used for teaching Multilayer Perceptrons; in addition, optimization algorithms such as Stochastic Gradient Descent (SGD) and Adam are employed [1]. In an attempt to make its predictions as accurate as possible, the system alters connection strengths that link neurons in various layers as it tries to reduce the difference between what it believes will happen and what occurs.

We used this model to process the tabular data we collected from BWDB. It's effective in learning complex patterns and relationships within the tabular data, providing insights into various factors influencing flood occurrences such as rainfall, river levels, etc.
The code maps the districts to corresponding divisions to process water level and rainfall information. First, it sets up these mappings and then links Google Drive to load the data into pandas DataFrame. Correct columns are ensured by this code, districts are mapped to their respective divisions, and rows that lack mappings are removed. Data are filtered for specific months and years and grouped by division to calculate mean values. Standardize the features that prepare the data for building a Multi-Layer Perceptron (MLP) model after merging the data. The MLP model is trained by repeating the data at least 49 times, and its predictions are made. The reshaped predictions are viewed as a heatmap of 7×7 matrix which reveals the data patterns in a manner that is easy to comprehend.
Here's the 7x7 matrix for the heatmap that was generated as the output:

$$\begin{bmatrix} 1.41025238 & 4.63703321 & 7.59122109 & 11.54640986 & 21.34104947 & 4.6489153 & 1.41025238 \\ 4.63703321 & 7.59122109 & 11.54640986 & 21.34104947 & 4.6489153 & 1.41025238 & 4.63703321 \\ 7.59122109 & 11.54640986 & 21.34104947 & 4.6489153 & 1.41025238 & 4.63703321 & 7.59122109 \\ 11.54640986 & 21.34104947 & 4.6489153 & 1.41025238 & 4.63703321 & 7.59122109 & 11.54640986 \\ 21.34104947 & 4.6489153 & 1.41025238 & 4.63703321 & 7.59122109 & 11.54640986 & 21.34104947 \\ 4.6489153 & 1.41025238 & 4.63703321 & 7.59122109 & 11.54640986 & 21.34104947 & 4.6489153 \\ 1.41025238 & 4.63703321 & 7.59122109 & 11.54640986 & 21.34104947 & 4.6489153 & 1.41025238 \end{bmatrix}$$

5.3 Feed Forwarding Model

5.3.1 Fully convolutional Neural Network

The FCNN, also termed A Fully Connected Neural Network, is some type of feedforward artificial neural network in which every neuron in one layer is connected to each neuron in the next layer, this dense connection allows this network to build sophisticated interactions between features, An input layer, one or more hidden layers, and an output layer are typically found in this network. Every hidden layer of neurons calculates a weighted average of all input signals to them before going through an activation function that could either be Relu, Sigmoid, or Tanh to enable the network to learn nonlinear patterns [6]. The last output layer gives what the network thinks will happen next as determined by how many neuron units are used at that point and what type of real functions for instance, for regression or classification. Fully connected neural networks are taught by the backpropagation method, tuned with tools such as Stochastic Gradient Descent (SGD) or Adam and the selection of loss function (for example Cross-Entropy Loss used in classification problems, Mean Squared Error in regression tasks) depends on the task at hand however [6]. Despite their ability to perform diverse functions, fully connected neural networks may be so demanding in terms of computation while at the same time suffering from overfitting when regularization steps are not followed. Despite tackling the above

challenges, FCNNs still stand as a powerful instrument when it comes to complex data modeling across different areas such as image classification, spam detection, and predictive analytics.
We made this a multi-modal model. So for that, we used FCNN. We took the results from the image processing and data processing and processed them together in FCNN for the final result. It's capable of learning non-linear mappings between input features and flood predictions, capturing interactions between different factors affecting floods effectively.
In our model, a custom dataset is used in cod to establish a pipeline for image classification that involves preprocessing, feature extraction, and an improved fully connected neural network (FCNN) model starting with the ImageDataset class to load images with augmentation of data [13]. Training and testing sets are separated for the dataset at an early stage and features are pulled from some model layers. The process uses flattening features to turn our 3D tensor feature input into 1D array for our model. It also uses label encoding to turn string labels into integer to feed into our model. The process then uses PCA to squash or reduce dimensionality while preserving as much characteristic as possible. Encoding is done on tags and the data is now in PyTorch' s tensor form. A model known as 'EnhancedFCNN' which includes batch normalization as well as dropout layer. We also used early stopping to prevent it from over-fitting or over-training. Precision, F1 Score, recall or sensitivity, along with accuracy values provide the evaluation measures for the model' s effectiveness. A heat map is used to visualize the confusion matrix made up of true positives, true negatives false positives, and false negatives plus F1 Score, Recall, or Sensitivity and Precision.

Chapter 6

Result Analysis

In this study, multispectral satellite images were used. Several machine learning models including ResNet, VGG19, Inception, GoogleLeNet and UNet were used to extract features from the flood images. Multilayer Perceptron was used to extract feature from the tabular data of water level and rainfall. The MLP model gave a 7x7 matrix of heat map that is given below.

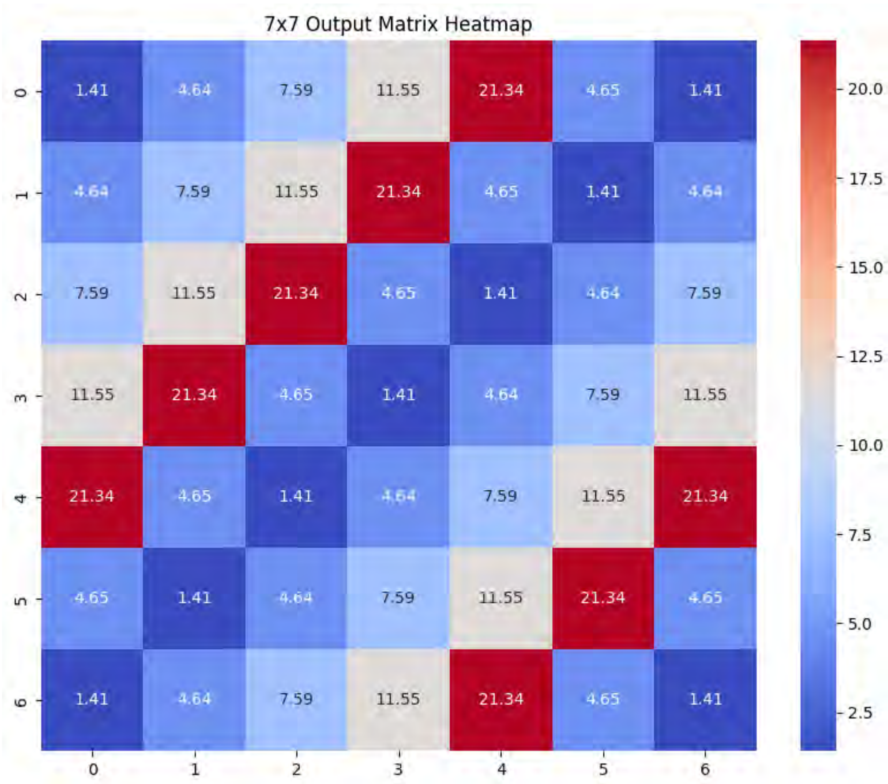

Figure 6.1: MLP 7x7 Matrix Heatmap

To detect flood from both of those data a forward feed model, FCNN was used. So from both the ResNet and MLP, the output matrix was taken as input for the FCNN model to train and test to find our the precision, accuracy, F1 score and accuracy. The same was done for VGG19 with MLP, Inception with MLP, GoogleLeNet with MLP and UNet with MLP. The explanation of the result are provided below.

6.1 Performance Evaluation

For prediction performance, precision and recall are very important measures. Especially when dealing with some imbalanced classes. These metrics can measure the fitness of a result and can be used to handle accuracy and loss functions.

6.1.1 Precision

Precision is an important indicator of positive predictive accuracy. It is calculated using the equation below [3]:

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

After using the formula, we got 94.2% for Resnet, 96.1% for VGG19, 92.3% for Inception, 80.7% for GoogleLeNet and 100% for UNet.

6.1.2 Recall

The magnitude of the truly valid consequence is determined by recall, also known as sensitivity. Calculate it with the equation [3]:

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

The results we got after using this are, 92.3% for Resnet, 76.9% for VGG19, 84.6% for Inception, 69.2% for GoogleLeNet, and 84.6% for UNet.

6.1.3 F1 Score

The F1 score provides a valid assessment of the accuracy of the test, taking into account both precision and recall. Its value ranges from 0 (worst) to 1 (best) and is the harmonic mean of these two measurements. F1 scores reflecting the trade-off between recall and precision are calculated as follows [3].

$$F1Score = 2 * \frac{Precesion * Recall}{Precesion + Recall} \tag{6.3}$$

The results for this formula are, 92% for Resnet, 83.3% for VGG19, 87.9% for Inception, 70.5% for GoogleLeNet, and 90.5% for UNet.

6.1.4 Accuracy

The accuracy derived from the confusion matrix calculates or represents the percentage of correct predictions.It is calculated as follows [3]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.4}$$

The results of this formula are, 92.3% for Resnet, 76.9% for VGG19, 84.6% for Inception, 69.2% for GoogleLeNet, and 84.6% for UNet.

6.2 Performance Analysis of Machine Learning Models

6.2.1 Performance Analysis

In our study, we thoroughly evaluated the performance of five different machine learning models; ResNet, VGG19, Inception, GoogleNet and UNet combined with Multilayer perceptron and evaluated them by forward feeding them through Fully Convolutional Neaural Network to get the precision, recall, f1 score, accuracy and also generated the Confusion matrix. The confusion matrix shows how many of the predictive class and actual class were correctly classified and how many were misclassified. Each of these models was developed with a specific flood prediction objective in mind. We were able to make meaningful comparisons between the different models and we learned important information about the specific benefits of each. The ResNet model when combined with MLP and applying FCNN showed robustness and high accuracy which shows it's ability to train deep networks without vanishing gradient. The VGG19 model combined with MLP and applying FCNN also gave a better accuracy but it was computationally slower. The Inception model combined with MLP and applying FCNN gave a similar accuracy ratethat needed careful implementation. The GoogleLeNet model combined with MLP and applying FCNN gave a little lower accuracy rate than the rest but it was computationally faster. Finally, the UNet model combined with MLP and applying FCNN showed a similar high accuracy which was possible due to it's segmentation and augmentation features.

Table 6.1: Performance Table

model	precision	f1-score	recall	accuracy
ResNet	0.942	0.920	0.923	0.923
VGG19	0.961	0.833	0.769	0.769
Inception	0.923	0.879	0.846	0.846
GoogleLeNet	0.807	0.705	0.692	0.692
UNet	1.000	0.905	0.846	0.846

6.2.2 Confusion Matrix

We generated confusion matrix that visualized our result for five different classes.
Here,
Class 0: No Flood
Class 1: Minor Flood
Class 2: Moderate Flood
Class 3: Major Flood
Class 4: Severe Flood

ResNet

With an accuracy of 92.31%, we generated a confusion matrix. The confusion matrix is given below:

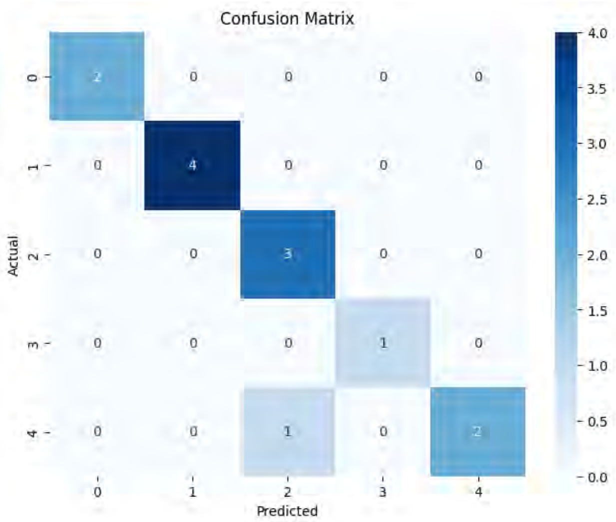

Figure 6.2: ResNet confusion matrix

VGG19

With an accuracy of 76.92%, we generated a confusion matrix. The confusion matrix is given below:

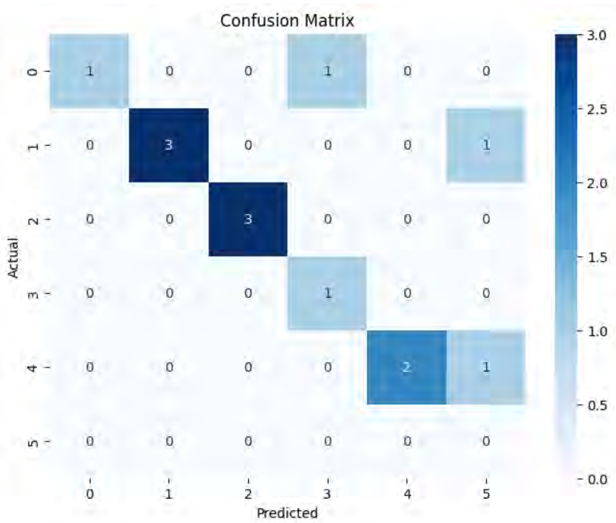

Figure 6.3: VGG19 confusion matrix

Inception

With an accuracy of 84.62%, we generated a confusion matrix. The confusion matrix is given below:

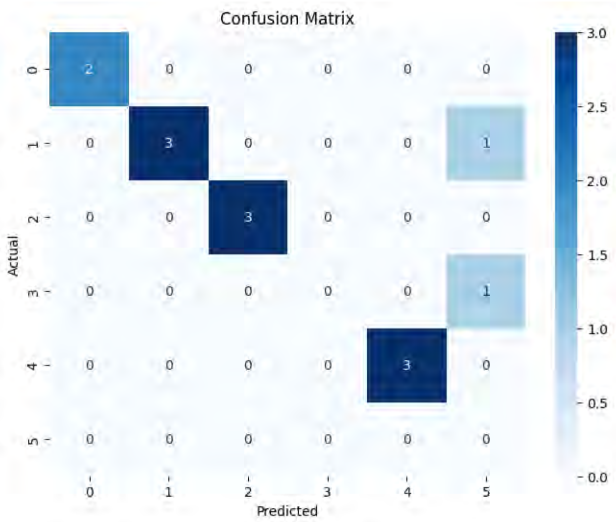

Figure 6.4: Inception confusion matrix

GoogleLeNet

With an accuracy of 69.23%, we generated a confusion matrix. The confusion matrix is given below:

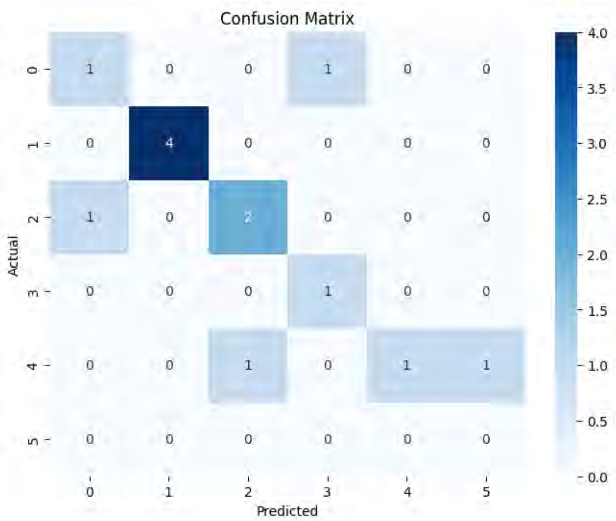

Figure 6.5: GoogleLeNet confusion matrix

UNet

With an accuracy of 84.62%, we generated a confusion matrix. The confusion matrix is given below:

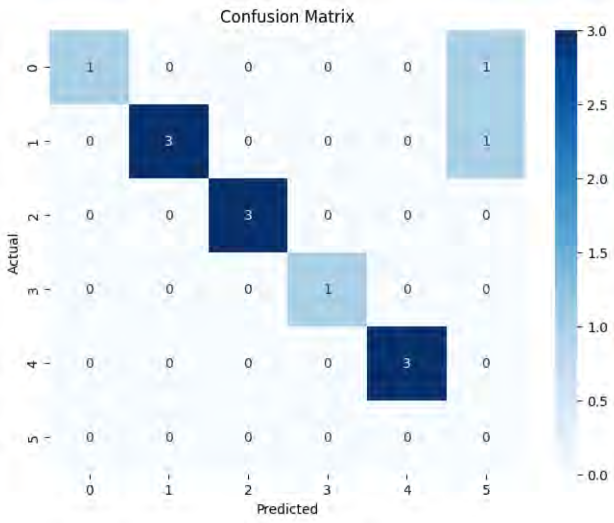

Figure 6.6: UNet confusion matrix

6.3 Comparison

The ResNet model showed high accuracy, obtaining 92.31% accuracy. It gives the classification accuracy of the FCNN model that we applied after taking the output of ResNet model and MLP model as input. When balanced for precision, F1 score and recall they were 0.942, 0.920 and 0.923 indicating adequate power.
The VGG19 model performed well for flood after testing where the accuracy came to 76.92%. With precision, F1 scores and recall as 0.961, 0.833, 0.769, they were highly optimal. So VGG19 obtained an impressive output.
The Inception model acquired better accuracy than VGG9 model but it was not as good as the ResNet model. With an accuracy rate of 84.62% and precision, F1 score and recall of 0.923, 0.879, 0.846, the model gave promising output.
Among all the models, GoogleLeNet gave the lowest accuracy with an accuracy rate of 69.23%. The precision, F1 score and recall for it was 0.807, 0.705, 0.692.
The UNet model gave the same accuracy rate as Inception model which was 84.62%. However, the precision was highest for this model which was 1 and the F1 score and recall were consequently 0.905 and 0.846 which were high as well.

When we compared the accuracy of machine learning models for flood detection, we found significant differences. With an impressive accuracy of 92.31%, ResNet proved adept at accurately classifying floods. With an accuracy of 84.62%, Inception model and UNet model came in second place, indicating its ability to accurately predict flooding classification. The VGG19 model in flood detection was highlighted by its exceptional performance with an accuracy score of 76.92%. The GoogleLeNet model also showed great promise with an accuracy rate of 69.23%. Overall, these results confirm the accuracy performance of all the models and provide useful insights into the performance of different models in urban pattern recognition and flood prediction tasks.

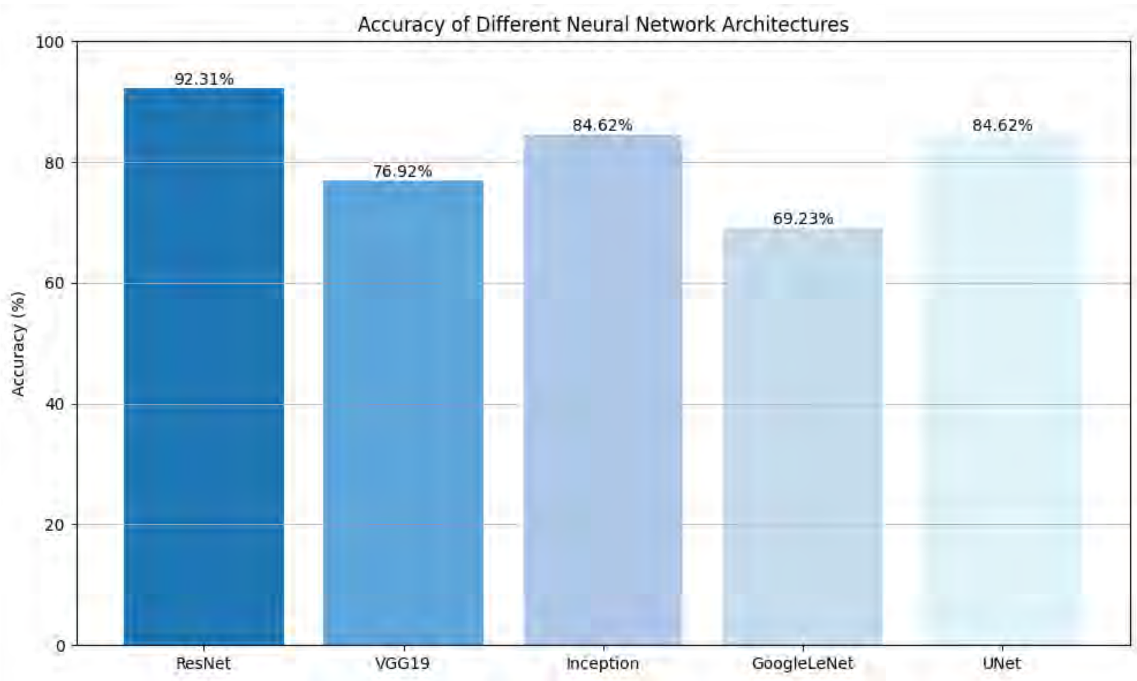

Figure 6.7: Accuracy Comparison

6.3.1 Comparison with Previous Models

The models we found from our literature review, used one or more types of satellite data. However, we took a multi-modal approach that works by taking both the satellite image data and tabular data. Here's a comparison table to compare the models, data types and methods of this paper with the previous techniques.

Source	Models	Data Type	Methods
Previous Models	CNN, RNN	Image Data Sentinel-1 satellite	Data preprocessing, deep learning architecture, model training and evaluation procedures.
	SVM, CNN, U-Net, ASPP-UNet, ResASPP-UNet	Image Data Worldview 2 and Worldview 3 satellite	Pattern recognition, training and testing of images, IFM' s and optimizing for layer depth utilization of NIR imagery.
	RF, SVM, MLC, CD	Image Data Sentinel-1 satellite	Supervised and unsupervised classification methods, flood detection; resource optimization for data and computations.
	UNet, LinkNet, SegNet	Image Data Sentinel-1 satellite	High-resolution earth imaging, SNAP tool for image analysis, moving target detection, and usage of ESA Copernicus data.
	Normal Bayes, KNN, RF, SVM	Image Data Worldview-2, Ikonos, Landsat ETM, TM, MSS satellite	Urban pattern classification, machine learning algorithm evaluation, image segmentation impact, data variation analysis.

Table 6.2 continued from previous page

<table>
<tr><th>Source</th><th colspan="3">Models</th><th>Data Type</th><th>Methods</th></tr>
<tr><td rowspan="2"></td><td colspan="3">FCNN</td><td>Image Data Sentinel-1 satellite</td><td>Flood detection, suitable watermarks, CNN model training and IOU as well as error rate parameters.</td></tr>
<tr><td colspan="3">RFC, KNN</td><td>Image Data Sentinel-1 satellite</td><td>Flood detection and forecast, binary to decimal conversions, flood mapping using supervised machine learning classifiers.</td></tr>
<tr><td rowspan="5">Our Models</td><td>ResNet</td><td rowspan="5">MLP</td><td rowspan="5">FCNN</td><td rowspan="5">Image Data Sentinel-1 satellite

Tabular Data Monthly Water Level and Monthly Rainfall data from hydrology department of BWDB</td><td rowspan="5">Flood mapping, Image feature extraction using ResNet, VGG19, Inception, GoogLeNet and UNet. Tabular data feature extraction using MLP. Extracted features feed forwarded to FCNN. Training and testing data. Flood detection classification.</td></tr>
<tr><td>VGG19</td></tr>
<tr><td>Inception</td></tr>
<tr><td>GoogLeNet</td></tr>
<tr><td>UNet</td></tr>
</table>

Table 6.2: Comparison Table of Previous Models and Our Models

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this paper, we presented the applications of Deep Neural Network Models. We collected image data from Sentinel-1. Sentinel-1 works best with flood or land images as it capturres radar data that can pierce through the cloud cover its vision. We used 5 different models for this multi-spectral satellite image to extract features; They are ResNet, Inception, GoogleLeNet, UNET, and VGG19.
We also collected tabular data from the Hydrology department of Bangladesh Water Development Board. There we have 2 sets of data; one is the monthly water level data and another is the monthly rainfall data. We applied Multilayer Perceptron for our tabulat data to extract features from it and generate a similar ourput.
To get better results we augmented the dataset before feed forwarding it into our Fully Convolutional Neural Network. From the acquired results after implementing the models, we see all the models give good accuracy, and all the models have great precision, recall, and F1 score, signifying this dataset is a good one and the models we chose are accurate in this research. ResNet works best with this type of data, with ResNet achieving the highest accuracy rate of 92.31%. We also have Inception and UNET achieving an accuracy rate of 84.62%. This is the second-highest accuracy rate. This result implies that the dataset acquired is of high quality and works for multiple architectures.
In summary, this thesis has advanced the use of multi-spectral satellite pictures and machine learning models for flood prediction and urban pattern recognition. Researchers, urban planners, and politicians can benefit greatly from the conclusions and approaches offered in this study when tackling the problems associated with urban development and flood control. Further study and innovation in this area hold enormous promise for building resilient and sustainable cities as technology develops.

7.2 Limitations and Future Work

For prediction, we require weather data, cloud pattern data, cloud thickness data, etc. which are currently unavailable to us. For this reason, we could not make a prediction model that would be accurate enough to predict the flood accurately. Our future priority would be making a prediction model by collecting those data, analyzing them, and coming up with a model that works best for the prediction.

Bibliography

[1] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[2] L. Peter, M. Matjaž, and O. & Krištof, "Detection of flooded areas using machine learning techniques: Case study of the ljubljana moor floods in 2010," Tech. Rep., 2013.

[3] G. K. Armah, G. Luo, and K. Qin, "A deep analysis of the precision formula for imbalanced class distribution," *International Journal of Machine Learning and Computing*, vol. 4, no. 5, pp. 417–422, 2014.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] M. Wieland and M. Pittore, "Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images," en, *Remote Sens. (Basel)*, vol. 6, no. 4, pp. 2912–2939, 2014.

[6] 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*, 2015, pp. 3431–3440.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[9] Y. Yang and Q. J. Wu, "Extreme learning machine with subnetwork hidden nodes for regression and classification," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2885–2898, 2015.

[10] *65 years of weather data bangladesh (1948 - 2013).csv (version v1)*, version V1, Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/emonreza/65-years-of-weather-data-bangladesh-preprocessed.

[11] A. Mosavi, P. Ozturk, and C. Kwok-wing, *Flood Prediction Using Machine Learning Models: Literature Review — mdpi.com*, https://www.mdpi.com/2073-4441/10/11/1536, [Accessed 18-09-2023], 2018.

[12] P. Zhang, Y. Ke, Z. Zhang, M. Wang, P. Li, and S. Zhang, "Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery," en, *Sensors (Basel)*, vol. 18, no. 11, p. 3717, 2018.

[13] D. Bonafilia, B. Tellman, T. Anderson, and E. Issenberg, "Sen1floods11: A georeferenced dataset to train and test deep learning flood algorithms for sentinel-1," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 210–211.

[14] P. Jain, B. Schoen-Phelan, and R. Ross, "Automatic flood detection in SentineI-2 images using deep convolutional neural networks," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, New York, NY, USA: ACM, 2020.

[15] R. S. Chellapa, S. K. Rajan, G. J. Eanoch, H. R. Yesudhas, L. Kumaragurubaran, H. V. Long, and R. & Kumar, *Urban matanuska flood prediction using deep*, https://www.researchgate.net/publication/354424895_Urban_Matanuska_Flood_Prediction_using_Deep_Learning_with_Sentinel-2_Images, Accessed: 2023-5-25, 2021.

[16] G. Prakash, P. K. Gupta, G. V. Rao, and D. Pratap, "Flood inundation mapping and depth modelling using machine learning algorithms and microwave data," *J Geomatics*, vol. 15, no. 2, pp. 221–229, 2021.

[17] G. Antzoulatos, I.-O. Kouloglou, M. Bakratsas, A. Moumtzidou, I. Gialampoukidis, A. Karakostas, F. Lombardo, R. Fiorin, D. Norbiato, M. Ferri, A. Symeonidis, S. Vrochidis, and I. Kompatsiaris, "Flood hazard and risk mapping by applying an explainable machine learning framework using satellite imagery and GIS data," en, *Sustainability*, vol. 14, no. 6, p. 3251, 2022.

[18] Ö. B. Çalışkan, *Automatic flood detection from satellite images using deep learning*, en, https://medium.com/@omercaliskan99/automatic-flood-detection-from-satellite-images-using-deep-learning-f14fafd369e0, Accessed: 2023-5-25, Aug. 2022.

[19] K. K. Mohbey, S. Sharma, S. Kumar, and M. Sharma, "Covid-19 identification and analysis using ct scan images: Deep transfer learning-based approach," in *Blockchain Applications for Healthcare Informatics*, Elsevier, 2022, pp. 447–470.

[20] A. H. Tanim, C. B. McRae, H. Tavakol-Davani, and E. Goharian, "Flood detection in urban areas using satellite imagery and machine learning," en, *Water (Basel)*, vol. 14, no. 7, p. 1140, 2022.

[21] M. Islam, N. J. Ria, and J. F. Ani, "Satellite imageries for detection of bangladesh' s rural and urban areas using yolov5 and cnn," *Mobile Information Systems*, vol. 2023, p. 1 814 906, 2023. DOI: 10.1155/2023/1814906.

[22] J. Jackson, S. B. Yussif, R. A. Patamia, K. Sarpong, and Z. Qin, "Flood or non-flooded: A comparative study of state-of-the-art models for flood image classification using the floodnet dataset with uncertainty offset analysis," *Water*, vol. 15, no. 5, p. 875, 2023.

[23] M. Buda and PyTorch, *Brain tumor segmentation with unet - pytorch hub*, https://pytorch.org/hub/mateuszbuda_brain-segmentation-pytorch_unet/, Accessed: 2024-05-21, 2024.

[24] PyTorch, *Googlenet model - pytorch hub*, https://pytorch.org/hub/pytorch_vision_googlenet/, Accessed: 2024-05-21, 2024.

[25] ——, *Inception v3 model - pytorch hub*, https://pytorch.org/hub/pytorch_vision_inception_v3/, Accessed: 2024-05-21, 2024.

[26] Torchvision Models, *Resnet18 model - torchvision*, https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html, Accessed: 2024-05-21, 2024.

[27] ——, *Vgg19 model - torchvision*, https://pytorch.org/vision/main/models/generated/torchvision.models.vgg19.html, Accessed: 2024-05-21, 2024.

[28] (). "Bangladesh meteorological department," Bangladesh Meteorological Department, [Online]. Available: https://live3.bmd.gov.bd/p/Temperature-Data.

[29] A. R. PRADANA, A. F. HADI, and I. INDARTO, "Application of pca-cnn (principal component analysis–convolutional neural networks) method on sentinel-2 image classification for land cover mapping,"