Masters Program in Geospatial Technologies

EXTRACTING SURFACE WATER BODIES FROM SENTINEL-2 IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

Janak Parajuli

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
Universitat Jaume I
Westfälische Wilhelms-Universität Münster

EXTRACTING SURFACE WATER BODIES FROM SENTINEL-2 IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

Dissertation supervised by:
Filiberto Pla Bañón, PhD
Professor, Institute of New Imaging Technologies (INIT),
Universitat Jaume I (UJI),
Castellon de la Plana,
Spain

Co-supervised by:
Rubén Fernández-Beltrán, PhD
Institute of New Imaging Technologies (INIT),
Universitat Jaume I (UJI),
Castellon de la Plana,
Spain

Co-supervised by:
Marco Octávio Trindade Painho, PhD
Professor, Institute for Statistics and Information Management (ISEGI),
University of Nova, Lisbon
Portugal

February 21, 2021

Acknowledgements

First and foremost, I would like to heartily acknowledge the **Erasmus Mundus Scholarship** scheme for providing me this glorious opportunity of pursuing Master of Science in Geospatial Technologies. I am hugely indebted to the consortium universities; **Universitat Jaume I, WWU Muenster and University of Nova, Lisbon**; which accommodated me with their graceful education. Similarly, I can not remain aloof from bidding huge loads of thanks to all my professors specially the coordinator **Prof. Dr. Joaquín Huerta Guijarro**, the tutors, technicians, librarians and all the staff who came by me directly or indirectly. I would also like to remember **Mr. Ruben Garcia Vidal** for his continuous help with administrative issues. On this auspicious moment how can I forget all those helping hands of **Señora Elena Martínez Santamaría** during the application process and throughout her tenure at UJI. I can just wish she could hear me saying this.

I would like to express my utmost gratitude to my supervisor **Prof. Filiberto Pla Bañón**, my co-supervisors **Rubén Fernández Beltrán** and **Prof. Marco Painho** for their untiring mentorship throughout the thesis. It was always an honor to have these men of letters as your supervisors. The knowledge I learnt under their guidance is simply inexplicable in words and I will always remain indebted for this.

The thesis was written in latex using the open source 'novathesis LaTeX template'. Hence, I am much obliged to **Assoc. Prof. João Lourenço**, the developer of this template.

I am very grateful to **Ms. Tina Baidar** and **Mr. Sanjeevan Shrestha** for always encouraging me to enter in the arena of deep learning which I believe was one of the sweetest decision to make. Similarly, I would like to raise my vote of thanks to **Mr. Mahesh Thapa** and **Ms. Laxmi Thapa** for providing me insights about deep learning and neural networks.

Besides, I would like to heartily thank my **parents, brothers, sisters, in-laws** and my beloved wife **Ms. Sushma Pokharel** who always kept on believing and encouraging me to move ahead. Finally I am grateful to all the relatives and friends who keep on inspiring me positively throughout my life ...

EXTRACTING SURFACE WATER BODIES FROM SENTINEL-2 IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

Abstract

Water is an integral part of eco-system with significant role in human life. It is immensely mobilized natural resource and hence it should be monitored continuously. Water features extracted from satellite images can be utilized for urban planning, disaster management, geospatial dataset update and similar other applications. In this research, surface water features from Sentinel-2 (S2) images were extracted using state-of-the-art approaches of deep learning.Performance of three proposed networks from different research were assessed along with baseline model. In addition, two existing but novel architects of Convolutional Neural Network (CNN) namely; Densely Convolutional Network (DenseNet) and Residual Attention Network (AttResNet) were also implemented to make comparative study of all the networks. Then dense blocks, transition blocks, attention block and residual block were integrated to propose a novel network for water bodies extraction. Talking about existing networks, our experiments suggested that DenseNet was the best network among them with highest test accuracy and recall values for water and non water across all the experimented patch sizes. DenseNet achieved the test accuracy of 89.73% with recall values 85 and 92 for water and non water respectively at the patch size of 16. Then our proposed network surpassed the performance of DenseNet by reaching the test accuracy of 90.29% and recall values 86 and 93 for water and non water respectively. Moreover, our experiments verified that neural network were better than index-based approaches since the index-based approaches did not perform well to extract riverbanks, small water bodies and dried rivers. Qualitative analysis seconded the findings of quantitative analysis. It was found that the proposed network was successful in creating attention aware features of water pixels and diminishing urban, barren and non water pixels.

All in all, it was concluded that the objectives of the research were met successfully with the successful proposition of a new network.

Keywords

Index-Based Approach

Deep Learning

Convolutional Neural Networks

Densely Convolutional Network

Residual Attention Network

State-of-the-art Approaches

INDEX OF THE TEXT

INDEX OF FIGURES

INDEX OF TABLES

Acronyms

ADAM	Adaptive Moment Estimation
ARL	Attention Residual Learning
A-SLIC	Adaptive Simple Linear Iterative Clustering
AI	Artificial Intelligence
AttResNet	Residual Attention Network
BN	Batch Normalization
CAMS	Copernicus Atmosphere Monitoring Service
CIFAR	Canadian Institute For Advanced Research
CLI	Command Line Interface
CNN	Convolutional Neural Network
CNNCWC	Convolutioanl Neural Network for Complex Wetland Classification
CNNWQC	Convolutional Neural Netowrks for Water Quality Control
DEM	Digital Elevation Model
DenseNet	Densely Convolutional Network
DL	Deep Learning
DNN	Deep Neural Networks
DS	Data Science
DSN	Deep Supervised Net
ECE	Edge Commission Error
EOA	Edge Overall Accuracy
EOE	Edge Omission Error
ESA	European Space Agency
ETM+	Enhanced Thematic Mapper Plus
EWI	Enhanced Water Index
FEA	Feature Expansion Algorithm
GEE	Google Earth Engine
GeoTiff	Geotagged Image File Format
GF	Gaofeng

GPS	Global Positioning System
GPU	Graphics Processing Unit
GWI	General Water Index
IoU	Intersection over Union
MAE	Mean Absolute Error
MIR	Mid-Infrared
ML	Machine Learning
MNDWI	Modified Normalized Difference Water Index
MODIS	Moderate Resolution Imaging Spectroradiometer
MRDED	Multi-Resolution Dense Encoder and Decoder
MSCNN	Multi-Scale Convolutional Neural Network
MSE	Mean Squared Error
NAL	Naive Attention Learning
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NIN	Network In Network
NIR	Near-Infrared
NWI	New Water Index
OLI	Operational Land Images
OTOP	Offline Training Online Prediction
PCA	Principal Component Analysis
PCs	Principal Components
QGIS	Quantum Geographic Information System
RADARSAT	Radar Satellite
REIS	Rapideye Earth Imaging System
ReLU	Rectified Linear Unit
RF	Random Forest
RGB	Red Green Blue
RNN	Recurrent Neural Network
S2	Sentinel-2
SAPCNN	Self Adaptive pooling Convolutional Neural Network
SGD	Stochastic Gradient Descent
SLIC	Simple Linear Iterative Clustering
SNAP	Sentinel Application Platform
SSAE	Stacked Sparse Autoencoder
SVHN	Street View House Numbers
SVM	Support Vector Machine

SWIR	Short Wave Infrared
TIFF	Tagged Image File Format
TIRS	Thematic Infrared Sensor
TL	Transfer Learning
UN-OCHA	United Nations Office for the Coordination of Outer Affairs
UTM	Universal Transverse Mercator
WGS	World Geodetic System
WRI	Water Ratio Index
WRN	Wide ResNet

Chapter 1

Introduction

1.1 Contextual Background

Water is significant part of nature with substantial role in human life [1]. It is one of the intensively exploited natural resources and hence its frequent monitoring is necessary for sustainable management [2]. Extraction of water bodies from satellite images is crucial for urban planning, disaster management, updating geospatial datasets, detection of droughts, monitoring of floods, navigation and other applications [1, 3, 4]. Knowledge on water-bodies can continuously monitor the conditions of available water resources and play significant role in environment conservation along with sustainable development [4]. [5] consider waterbodies as crucial factors for environmental testing, heat-island effects and ecosystem. They found changes in water distribution can have huge impact on human lives which can cause soil subsidence, inland inundation and health hazards.

Water bodies are also the integral part of different thematic and topographic maps used by human beings. They change from time to time unlike other features like buildings and roads which are considered as relatively stable[6]. Hence, timely update of the water dataset is necessary. Unfortunately, it is often found to be difficult because of the hectic and time consuming traditional approaches [7].

Satellite images in this research context are the images of earth captured by various satellites. Satellites are operated by countries and business organisations throughout the world. In remote sensing, there are several satellites

which provide earth observation imageries. Some of them are discussed here:

- **Landsat:** It is the first satellite of its kind with longest history of observing earth since the first launch in July 23 1972. Since then eight versions of the satellites have been launched out of which only Landsat 7 and 8 are currently operational [8, 9]. Landsat 7 contains one Enhanced Thematic Mapper Plus (ETM+) sensor and Landsat 8 contains two sensors called Operational Land Images (OLI) and Thematic Infrared Sensor (TIRS). Landsat 8 provides first 7 bands and 9^{th} band in 30m resolution, band 8 panchromatic in 15m. Band 10 and 11 are provided as thermal infrared in 100m resolution [10].

- **Sentinel:** Sentinel is a mission of European Union's earth observation program called Copernicus. As of now, there are six missions of Sentinel namely 1,2,3,4,5 &5P with different objectives. Sentinel 1 is supposed to provide aids in continuous radar mapping of the earth [11]. Sentinel-2 provides data for different applications like land ecosystem monitoring, land cover change, water quality monitoring, public security and disaster mapping [12, 13]. Sentinel 3 was launched with the purpose of environment and climate monitoring, temperature measurement of sea surface and developing ocean forecasting systems [14]. Sentinel 4 provides support to Copernicus Atmosphere Monitoring Service (CAMS) by assessing the primary gases and aerosols influencing the air quality [15]. In addition to Sentinel 4, the satellites Sentinel 5 and 5 precursor also support CAMS in the regards of air quality, composition-climate interaction and atmosphere monitoring [16, 17].

- **MODIS:** Terra and Aqua satellites have same instrument on board called Moderate Resolution Imaging Spectroradiometer (MODIS) which provide data in 36 spectral bands [18]. The data are related to the events occurred on lower atmosphere, oceans and land thereby visualizing the global dynamics. Bands 1-2 are provided in 250m resolution while 3-7 in 500m and 8-36 in 1000m resolution [19].

- **Rapideye:** Rapideye is a constellation of 5 commercial satellites owned by BlackBridge. It provides data with 5 multispectral bands: Red, Green, blue, red edge and near Infra-Red bands [20]. The imaging system called as Rapideye Earth Imaging System (REIS) is a pushbroom instrument

which has ground sampling distance of 6.5m at nadir and pixels size of 5m [21].

Similarly, there are numerous other open and commercial satellites like ZY-3, EnviSat, Corona, RADARSAT etc. dealing with earth observation [22]. In this research Sentinel-2 image products were used because of the free availability, spectral resolution of 13 bands and higher spatial resolution of up to 10m.

1.2 Motivation and Problem Statement

Traditional approaches of water extraction have limitations in accurately distinguishing water from snow, mountains, buildings and shadows [1, 4, 5, 23]. Auxiliary data like Digital Elevation Model (DEM) and complex band equations are required to address these issues [24]. Another challenge with them is to choose the most suitable threshold value to extract out smaller water bodies accurately [25]. Also, the spatial dependency of the threshold value only exacerbate the issue. Moreover, traditional approaches are not suitable for global scales because they do not integrate the shape and textual information of water pixels which vary drastically on global water bodies [26, 27].

Due to larger depths, neural networks suffer vanishing gradient problem because not all the layers play same role in contributing the learning process while training [28]. Also larger depths and number of feature maps increase the number of parameters degrading the computational efficiency of hardware [23, 29]. Densely Convolutional Network (DenseNet), proposed in 2016 and fully developed in 2018, can address these problems in addition to strengthening feature propagation and reuse [30]. In addition, Attention network can allow the decoder to dynamically 'pay attention' to only the relevant layers of current decoding step thereby increasing the quality of a network [31, 32]. Residual layers help by providing better representation of features inside deeper layers [23]. The limitation with these architects is that their individual efficacies are well demonstrated within CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets [33] but not integrated to extensively use in water feature extraction problems to the knowledge of the researcher.

1.3 Aim and Objectives

The main aim of the research was to extract surface water bodies from Sentinel-2 imagery using convolutional neural networks. To achieve the main aim, work

breakdown structure was followed by dividing the research works in following objectives:

- To explore the state-of-art approaches used for extraction of water bodies from satellite images.
- To implement the state-of-art approaches in the study area data and compare the performance achieved.
- To innovate and design a CNN architecture with highest qualitative and quantitative performance using the available architectures.

1.4 Research Workflow

The overall workflow was divided into three phases: Preprocessing, Processing and Post Processing. Preprocessing phase was begun with the downloading of Sentinel-2 images from ESA Copernicus website. DEM and Ground Truth Data were acquired from United Nations Office for the Coordination of Outer Affairs (UN-OCHA). These data were pre-processed as described in the section 4.2 and made uniform in terms of spatial resolution, dimension and geo-location to make image-label pairs. Then patches of images and labels were extracted using the steps described in section 6.1.1. Thus prepared patches were fed as input for the models in the processing phase. A total of six different architects were used to train, test and assess their performance. Among them, one architect with two orthodox convolutional layers was prepared as baseline model. Three different architects proposed by [5, 34, 35] were selected to study their performance in the study area. Similarly, Attention network in conjunction with Residual layers (AttResNet) and DenseNet were also implemented as state-of-the-art approaches for water bodies extraction. Later, these two novel architects were fused to propose a new architect for water bodies extraction. The details on configuration and implementation of models for experiments is described in the chapter 5. In the post-processing phase, qualitative and quantitative assessment of the performance of all the networks and proposed network were done. Besides, the output from our network was also compared with four different index-based approaches. In addition, we also implemented a traditional but novel approach called Enhanced Water Index (EWI) proposed by [25]. The overall workflow from data downloading up to the evaluation of the performance can be found in the figure 1.1.

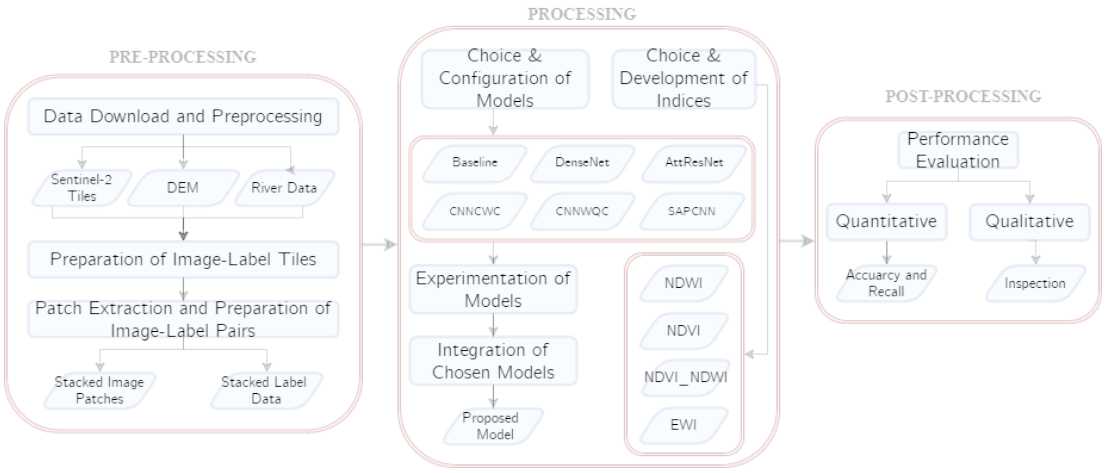


Figure 1.1: Overall Flow of Research Steps from data download to performance evaluation

1.5 Thesis Contribution

With consideration of the problems stated in the section 1.2, the thesis is expected to provide the following contributions to the study area and the scientific community:

- It is the novel application of neural networks in the study area to extract any features [36].
- It is the novel implementation of DenseNet, Residual and Attention networks to extract surface water bodies from Sentinel-2 images.
- A new network is proposed by integrating attention blocks from AttResNet with DenseNet.

1.6 Thesis Structure

The thesis contains eight chapters. **Chapter 1** talks about the contextual background of the thesis. Then it introduces the motivation, problem statement, aims and objectives of the research. It continues with describing the methodology followed and contribution of the research to the study area and scientific community. **Chapter** 2 reviews the literature on the traditional and modern approaches for water bodies extraction from satellite imagery. It explains about the index-based approaches like NDWI, MNDWI, NDVI, NWI, GWI, and EWI. Then it introduces and describes the modern approaches like SAPCNN, CNNWQC, CNNCWC, DenseNet and AttResNet used in deep learning. **Chapter**

3 introduces the theoretical framework of CNN where it briefly introduces AI, ML, DL and DS. Then it details about CNN, its architecture and different terminologies associated with it. **Chapter 4** familiarizes about the study area and the process of dataset preparation. **Chapter 5** describes the methodology of water bodies extraction using index-based methods and neural networks. It talks about how the baseline model was developed along with the modification of SAPCNN, CNNWQC, CNNCWC, denseNet and attResNet to make them uniformly comparable. **Chapter 6** presents all the outputs obtained from index-based approaches and neural network methods. It explains the experiments conducted and their numerical and visual results. **Chapter 7** talks about the limitations of the research and recommendations on how the similar works should be conducted in future. Finally, **Chapter 8** concludes with the overall insights and achievements of this thesis.

Chapter 2

Literature Review

This chapter explains the literature on water feature extraction using traditional and deep learning approaches. Section 2.1 talks about traditional approaches like threshold and indexing methods used commonly. Specially, it is focused on how water-index methods can extract water features using band equations and their limitations. Section 2.2 describes deep learning approaches for water feature extraction. It is focused on explaining the literature of five architects that were used in this research for experimentation.

2.1 Traditional Approaches

Water features found in satellite images include the likes of rivers, streams, falls, ponds, lakes etc. There are various approaches of water features extraction using different remote sensing methods. Some common methods employed in most of the researches can be categorized as [4]:

- Single-band threshold method
- Multi-band threshold method
- Water-index method

Single band threshold method distinguishes the single band spectral properties of water in contrast to other objects while the multiband sprectral threshold method do the same for multiple number of bands. Water index method employs the conjugative ratio of green and red bands to segregate water [37].

This approach is found to be mistaking building noise as water but it can be improved using mid-infra red band in place of red band [38, 39].

Researchers have used different water-index methods like NDWI, MNDWI, NDVI, NWI, GWI, WRI etc. Normalized Difference Water Index (NDWI) is the most commonly used one, calculated using green and NIR band as in equation 2.1.

$$NDWI = \frac{\rho_{green} - \rho_{NIR}}{\rho_{green} + \rho_{NIR}} \tag{2.1}$$

where ρ_{green} and ρ_{NIR} are reflectance in Green and NIR band respectively. The index value ranges between -1 and 1, positive values indicating water bodies [25, 40].But due to higher reflectance in Green and lower in NIR bands, builtup areas may also have positive values for NDWI. So NDWI fails to distinguish water bodies from builtup features properly [40]. [41] proposed remedy to this limitation by replacing NIR by Mid-Infrared (MIR) in index thereby introducing Modified Normalized Difference Water Index (MNDWI) as:

$$MNDWI = \frac{\rho_{green} - \rho_{MIR}}{\rho_{green} + \rho_{MIR}} \tag{2.2}$$

where symbols have their usual meanings. Even the built-ups become negative and hence the index can uniquely extract the water features. Despite this fact, MNDWI is suitable only for urban water bodies and mixes mountain shadows and snow cover [42].

Similarly, another index termed as New Water Index (NWI) was proposed by improvising equation 2.1. The green band was replaced by blue and the NIR band was added up with both MIR bands of Landsat TM image as given in the equation 2.3 [43].

$$NWI = \frac{\rho_{blue} - (\rho_{NIR} + \rho_{MIR_1} + \rho_{MIR_2})}{\rho_{blue} + (\rho_{NIR} + \rho_{MIR_1} + \rho_{MIR_2})} \tag{2.3}$$

In contrast to these ratio indices, [44] proposed a non-ratio index, General Water Index (GWI); using difference of visible and infra-red bands as:

$$GWI = (\rho_{Green} + \rho_{Red}) - (\rho_{NIR} + \rho_{MIR}) \tag{2.4}$$

where symbols have their usual meanings. Unlike other indices, this index is not normalized and hence the threshold needs to be set manually to identify the boundary between water and non-water features. This adds complexity and makes the process time consuming.

There are other indices like Water Ratio Index (WRI) and NDVI also used for extraction of water bodies. WRI employs the conjugative ratio of green

and red bands to segregate water [37]. NDVI employs the difference between NIR and Red bands to primarily extract vegetation but also performed well as negative index for water extraction in case of [45].

To overcome the limitations of these index-based approach and to enhance the efficient computation with reduced data size, use of Principal Component Analysis (PCA)) was proposed by [25] and coined Enhanced Water Index (EWI). They found that introducing new non-collinear Principal Components will remove the effect of collinearity between bands of image, and make outputs more accurate .A general practice in PCA is to take in account first few (usually three) Principal Components that can integrate more than 95 percent of the information from original imagery [46]. With EWI, they achieved reduced processing cost with limited data volume to be handled and best extraction result.

2.2 Deep Learning Approaches

There are numerous approaches proposed by different studies for water feature extraction with higher accuracies using deep learning. [1] extracted urban water from Landsat imageries by combining a 'Multi-Scale Convolutional Neural Network (MSCNN)' with Google Earth Engine (GEE). The parameters were computed offline by training MSCNN water extraction was done online on GEE using an approach called 'Offline Training Online Prediction (OTOP)'. The OTOP method was concluded to be accurate and satisfactory automation level and can be used to extract water on different temporal and spatial location. Stacked Sparse Autoencoder (SSAE) method was used by [4] by creating a unique feature matrix of water, vegetation and building indices for each pixel. Then the feature matrices were expanded considering the effect of neighboring pixels and fed to SSAE as input to extract water. This Feature Expansion Algorithm (FEA) method was found to be better than other models like Support Vector Machine (SVM) and older neural networks. [23] proposed a framework called 'Multi-Resolution Dense Encoder and Decoder (MRDED)' which is intended towards the extraction of water and shadows but is silent about the form of water and impact of terrain on the result of extraction. A model called DeepWaterMap, employed by [26] distinguishes surface water from land, ice, clouds, snow and shadows.

In the sections 2.2.1 and 2.2.2, the theory behind the three chosen approaches from other researchers and two novel approaches for water feature

extraction are explained.

2.2.1 Related Works

- **Self Adaptive pooling Convolutional Neural Network (SAPCNN) for Urban Water Bodies Extraction:** It was proposed by [5] for urban water extraction by using high resolution multispectral images of ZY-3 and Gaofeng-2 satellites. This method used the concept of improvised Simple Linear Iterative Clustering (SLIC) approach to do the segmentation of images.

  SLIC is an algorithm to create small cluster of pixels termed as superpixels that possess similar features [47]. It is superior than pixel-based algorithm and widely used in the process of acquiring local information, preserving boundary information and extraction of features [48]. SLIC requires previously determined number of clusters and has small search space. To address this limitation [5] improvised the SLIC by implementing affinity propagation clustering and expanding the search space. The improvised approach was termed as Adaptive Simple Linear Iterative Clustering (A-SLIC). Then the superpixels were classified as water and non water pixels with newly designed CNN to extract high-level water features from urban background. In the last step, thus extracted water superpixels were converted into a high resolution image.

  SAPCNN was implemented using four images of the three downtown districts of China; Beijing, Tianjin and Chhengdu which featured ponds, lakes, small rivers, water parks etc. The researchers conducted four different experiments to assess the abilities of their proposed methodology. They examined the impact of super-pixel segmentation on the performance of water mapping and found an effective improvement in water extraction accuracy. Similarly, the self adaptive pooling ability of this model was also compared with max pooling and average pooling models. The model outperformed the rest two in terms of Edge Overall Accuracy (EOA), Edge Commission Error (ECE) and Edge Omission Error (EOE). Additionally, ability of shadow distinction was compared with SVM and NDWI to find the better results generated by SAPCNN. Finally, a comparison of water extracting efficiency was made among SVM, NDWI and two other methods proposed by [26, 49]. It was found that SAPCNN extracted water with the highest overall accuracy and producer accuracy

with lowest EOE and ECE.

All in all, SAPCNN was found to be efficient in improving the accuracy of urban water detection from high resolution satellite images. The way how it was implemented in this research is given in section 5.1.2.

- **Convolutional Neural Netowrks for Water Quality Control (CNNWQC):** It was proposed by [34] to classify water quality of inland lakes from Landsat8 images. A 4-layered CNN with hierarchical structure was developed to estimate the non-optically active parameters responsible to determine water quality levels. The relationship between in-situ water quality levels and the images were detected and the surface quality of total water was also classified. To address the lack of data from in-situ measurement, a Transfer Learning (TL) approach was implemented. For this, the model was trained with the data of Erhai Lake, Yunnan Province and the knowledge was transferred to Chaohu lake of Anhui Province, China.

  A total of 81 images (41 of Erhai lake and 40 of Chaohu lake) from Jan 2014 through October 2018 were utilized. In parallel, water quality data of the same date range was also collected and integrated with the architect. Then water quality classification performance of the architect was assessed by comparing with the traditional machine learning methods SVM and RF. The CNN model was found to be the best one in learning all the shallow, discriminating and complex features from the image. Also, the robustness of the model was tested by conducting transfer learning with the model trained on Erhai lake and implemented to the in situ water quality measurement data of Chaohu lake. The CNN with TL outperformed the one without TL by 9.52%.

  All in all, this approach implemented CNN as a cost effective mode for water quality classification and demonstrated the power of the same in extracting the relationship between the satellite images and water quality levels. The way how it was implemented in this research is given in section 5.1.3.

- **Convolutioanl Neural Network for Complex Wetland Classification (CNNCWC):** It was proposed by [35] for classification of complex wetland using satellite imagery. The research applied high-level spatial features in classification schemes for land cover mapping by fine tuning the

pre-existing CNN; AlexNet. It was implemented in Newfoundland and Labrador, Canada featuring eco-regions with varying geo-morphology, hydrology and ecology. The study considered the wetlands like bog, swamp, fen, marsh and surface water. For image data, two level-3A RapidEye images from June 18 and October 22, 2015 were acquired. A total of 191 sample sites were visited in Summer of 2015 and fall of 2016. Spatial distribution and land cover types of each sample sites were recorded along with their GPS location.

Upon observation of extracted features from some random patches, it was found that initial layers tend to extract low-level features like edges. High level features like pattern and textures were extracted by deep layers of CNN. The classification results of CNN was compared with a machine learning approach called RF. Even the results from only three features as input to CNN outperformed that from eight features as input to RF. Producer and user accuracy from CNN outputs were higher in each of the classification outputs for bog, fen, swamp, marsh, upland, urban, shallow water and deep water than from RF respectively. RF was found to be performing better for non-wetland (deep water) classification than wetlands.

All in all, the research demonstrated its efficiency to serve as a baseline model for wetland mapping from remote sensing images. It also opened the quest of fine tuning existing CNNs like AlexNet, DenseNet, ResNet and so on for the purpose of complex wetland classification in future. The way how it was implemented in this research is given in section 5.1.4.

2.2.2 Novel Architects

- **Densely Convolutional Network (DenseNet):**

  The neural networks with deeper layers tend to suffer vanishing gradient problem since all the layers do not have equal role in feature extraction [28]. In addition the increment in the number of layers and feature maps effectively increases the number of parameters which is computationally costly for a hardware [23, 29]. DenseNet addresses these problems by strengthening feature propagation and reuse [30]. The feature maps of preceding layers become the input for each layer. If there are N number of layers in the network, traditional networks would have N number

of direct connections but denseNet do have $\frac{N(N+1)}{2}$ direct connections making the feature maps of current layers as input of succeeding layers.

DenseNet was evaluated on four standard datasasets: CIFAR-10, CIFAR-100, SVHN and Imagenet. The network was compared with the likes of FractalNets, Network In Network (NIN), Deep Supervised Net (DSN), Highway Network and variants of ResNet. On CIFAR and SVHN data, it was found that DenseNet had the least error rates than the rest networks. Increase in number of layers and growth rate produced more efficient results without suffering overfitting or optimization problems. Even with lesser parameters also, DenseNet produced lower error rates that its counterparts with higher parameters. Similar result was achieved with ImageNet dataset as well.

All in all, DenseNet models are proved to be compact and robust models which can enforce deep layers to learn high level features. It also reduces redundancy by making feature reuse and differentiating between the gained and preserved information between layers. The way how it was implemented in this research is given in section 5.1.5.

- **Residual Attention Network (AttResNet):**

  Attention networks possess the capability to determine the focus areas of features. It increases the quality of networks by paying attention to the concerned layers only in order to extract relevant features [31, 32]. Residual layers provide better feature representation within deep layers [23]. To take the benefit of both networks, [50] proposed 'Attention Residual Learning (ARL)' by stacking multiple attention modules to develop AttResNet. The peculiarity of this network is bottom-up top-down structure is embedded in each attention modules to integrate feed forward and attention feedback process into a single feed-forward process. The feed forward operation collects the total information of the image quickly and attention feedback integrates the information with the original feature maps.

  Attention and normalization modules are adaptable in accordance with the main features. Channel attention normalizes within all channels to remove spatial information from each position. Spatial attention normalizes within feature maps from each channel and performs sigmoid activation to retain spatial information only. The third attention called as mixed attention simply performs sigmoid activation for each position

and channel so that it keeps both information. It was found that mixed attention achieved the least Top-1 error percentage among the three. The network was further evaluated on three benchmark datasets called CIFAR-10, CIFAR-100 and ImageNet.

At first model effectiveness was assessed using ARL in comparison to Naive Attention Learning (NAL) and ResNet-164. ARL was found to have the lowest Top-1 error rate with significant reduction in noise without loss of much information. Relative Mean Response was also found to be suitable in contrast to NAL which vanished in the next stage. In addition, the encoder and decoder structure of the network was compared with local convolutions to again find the lesser Top-1 error rate than the later. Similarly, noisy label robustness was assessed by increasing the noise level percentage. Compared to ResNet-164, test error rate of attResNet declined more gradually. The network was than compared with the variants of two state-of-the-art approaches; ResNet and Wide ResNet (WRN). AttResNet obtained the least error rate in both CIFAR-10 and CIFAR-100 datasets compared to the rest networks. Further, experiments done on ImageNet data on mask influence and different basic units depicted the least Top-1 and Top-5 error rated for attResNet. It also outperformed the likes of ResNet, ResNext and Inception modules as well.

All in all, the AttResNet proved its worthiness in capturing the different types of attention aware features and its extensible ability to compose a larger network. The way how it was implemented in this research is given in section 5.1.6.

Chapter 3

Theoretical Framework of CNN

This chapter talks about the theoretical framework of deep learning and technical aspects of CNN in particular. Section 3.1 briefs about the relationship among Artificial Intelligence, Machine Learning, Deep Learning and Data Science. Section 3.2 focuses on the depth of CNN architecture and the common layers present in a network.

3.1 Overview of the Context

Artificial Intelligence (AI) refers to 'intelligent-machines' that have the ability to copy human traits from a large set of data observations [51]. It can be sub-categorized into Machine Learning (ML) where the computers learn from previous experience to solve real world problems [52]. ML implements a lot of techniques like SVMs, neural networks, regression, clustering, bayesian learning, decision trees etc. Deep Learning (DL) is a subset of the neural networks which has a series of interconnected networks to improvise the computational efficiency of computer[53]. It has the capacity to do multi-layered data processing with data abstraction to enhance the capabilities of machine learning[54]. For image classification, deep learning techniques such as Deep Neural Networks (DNN), CNN, Recurrent Neural Network (RNN) etc are currently in practice [53]. DNNs are the preliminary neural networks which has more than two layers for computation and can modulate complex functions. As they are fully connected between all the layers, they need large number of parameters and consume more memory [55] .

CNNs are the neural networks which can learn efficiently from huge set of data because they possess 'adaptive filters' to compute the weights from raw input for automatic feature extraction [3, 56]. RNNs are the networks which can process sequential data by repetitively using same functions and parameters [57]

On the other hand, Data Science (DS) is an extension of statistics and traditional data management evolved for handling huge amount of data [58]. It integrates the discipline of computer science with statistics and deals with massive amount of data. Generally it is an intersection of three aspects; knowledge on math and statistics, concrete expertise and hacking skills [[59] adapted from [60]].

The relationship among these state-of-the-art approaches can be summed up as in fig 3.1

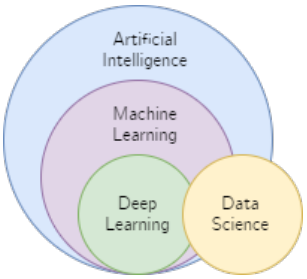


Figure 3.1: Relationship among AI, ML, DL and DS

3.2 CNN and its architecture

CNN is the state-of-the-art-technology for image processing tasks [61]. A typical CNN consists of a convolutional layer, activation function, pooling or subsampling layer, fully connected layer and output layer. It may also contain Dropout and Batch Normalization (BN) layers as optional components. Figure 3.2 shows the structure of a CNN architecture.

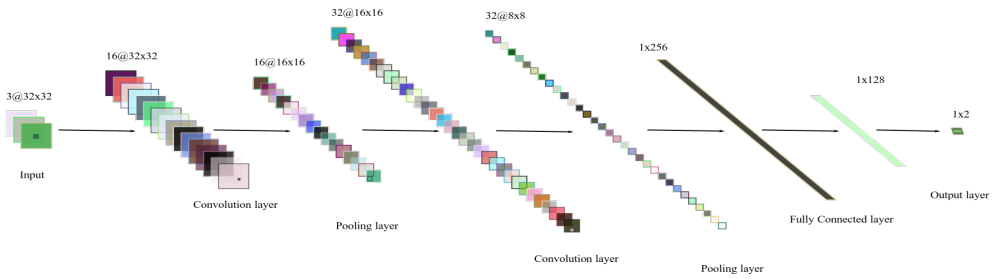


Figure 3.2: Architecture of a typical CNN

3.2.1 Convolutional layer

Convolutional layer is the basic component of a CNN architecture. It contains a set of learning filters called kernels [62]which use convolution operation to extract features from image and map them into feature maps. Convolution is a mathematical operation of two functions f and g derived by integration of the two according to the equation 3.1 [63]

$$(f * g)(n) = \sum_{m} f(m)g(n-m) \tag{3.1}$$

As shown in table 3.1, kernels slide over all the pixels of an input image and perform dot product with the local pixels to which a bias term and activation function are applied to form a feature map.

Table 3.1: Convolution Operation by a kernel of 3*3

0	1	0	0	1	1	1	0
0	0	1	1	1	0	1	0
0	0	1	0	1	1	1	1
1	0	0	0	0	1	0	1
0	1	1	1	0	1	1	0
0	1	0	1	0	0	1	0
0	1	1	0	1	1	0	1
1	1	0	1	1	0	0	0

Input Image

$\otimes$

0	1	0
1	0	1
0	1	0

Kernel

=

2	2	2	3	4	2
1	1	3	2	3	3
2	2	1	2	2	4
2	2	2	2	2	2
2	4	1	2	3	1
3	1	4	2	1	3

Feature Map

All the channels of input image share the same kernel to increase the feature detection ability irrespective of the location, reduce the number of parameters and enhance computational efficiency [64]. The activation function introduces the non-linearity in the model and it is briefly explained in the section 3.2.2

3.2.2 Activation Function

Activation function takes the output from each convolutional layer, increases the non-linearity of the network and convert them into activation maps [65] It transforms the activation level of a neuron into output signal in a defined range; typically -1 to 1 or 0 to 1 [63]. Some common activation functions are:

- **Tanh Activation function:** The hyperbolic tangent activation function outputs the signal in the range [-1,1].

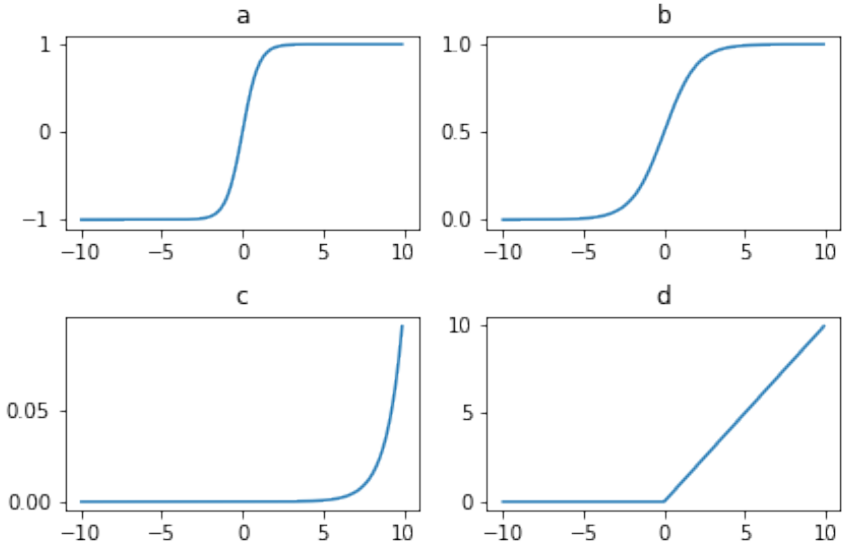


Figure 3.3: Common activation functions used in a CNN: a)tanh, b)sigmoid, c)softmax and d)ReLU

- **Sigmoid Activation function:** Sigmoid activation function is S-shaped curve used for binary classification. It similar to tanh but outputs the signal in the range [0,1]. Mathematically, it is represented as in equation 3.2

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

- **Softmax Activation function:** Softmax is a generalized sigmoid activation function used normally in multi-class problems but can be used in binary classification as well. Mathematically, it is represented as in equation 3.3

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{n} e^{x_k}} \tag{3.3}$$

- **Rectified Linear Unit (ReLU) Activation function:** ReLU is commonly used activation function which prompts the same value if the input is positive but converts into zero if it is negative. Mathematically, it is represented as in equation 3.4

$$f(x) = max(0, x) \tag{3.4}$$

Among these activation functions, ReLU is the preferable one in most cases because it has better performance with lowest vanishing gradient problem than the rest [66, 67]. The graphs of all the above activation functions is given in fig 3.3

3.2.3 Pooling or Sub-sampling Layers

Pooling layer is the sequential layer after convolutional and activation function layers. The prime objective of pooling layer is to reduce the spatial size of image and thereby reduce the complexity and number of parameters for further processing. It applies non-linear down-sampling to input image. Commonly there are three types of pooling layers: max-pooling, average-pooling and sum-pooling. A Max-pooling kernel returns the maximum value, average-pooling returns the average value and sum-pooling returns the sum of all the values in the input region.

Table 3.2: An instance of Max-Pooling

0	1	2	1	3	5	2	6
1	1	1	2	4	4	3	4
1	3	5	7	1	2	3	5
2	4	4	1	1	1	4	4
4	3	0	8	2	0	1	2
2	3	1	1	6	1	4	3
1	4	1	3	5	1	2	3
9	1	0	4	4	0	1	4

Input Feature Map

Max-Pooling
kernel 2*2,
stride 2

=

1	2	5	6
4	7	2	5
4	8	6	4
9	4	5	4

Pooled Map

As shown in table 3.2 pooling kernel slides over all the pixels of feature maps and return the respective values from the input region.

3.2.4 Fully connected layer

The stack of convolutional, activation function and pooling layers comprise the feature detection and extraction phase of a CNN network. The next phase is feature classification which consists of fully connected and output layers. The fully connected layer takes the output of previous layers and flattens them into a single vector. Each nodes of fully connected layers are connected with previous neurons and their values contribute in predicting the probability of a class [63]. Figure 3.4 shows two fully connected layers in conjunction with an output layer in a CNN.

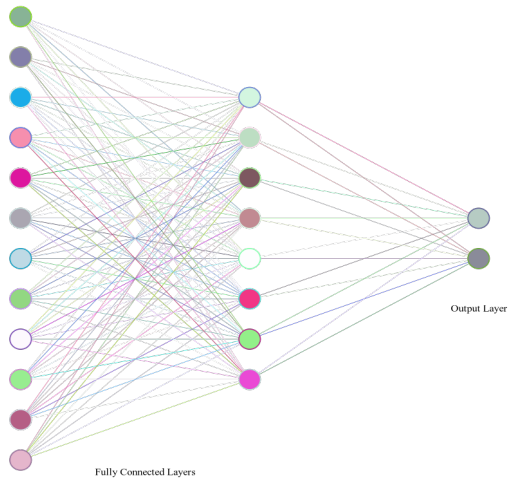


Figure 3.4: A schematic representation of fully connected layers in connection with output layer in a CNN

3.2.5 Output layer

Output layer is the last layer of a CNN which shows the probabilities of each predicted class called as class scores. In multi-class classification problems, softmax is used to compute the probabilities of each class. In binary classification problems typically sigmoid is used to determine whether the feature matches the class or not.

In addition to above layers, a typical CNN can contain optional layers like Dropout and Batch Normalization layers as well. Dropout layer is used to temporarily disable certain proportion of nodes of a hidden layer so that the network learns limited amount of information. This is done to prevent the overfitting and improve the generalization of the network [63]. Batch Normalization layers are introduced to reduce overfitting and model divergence since it also plays some role in the speeding up the model convergence with faster learning rates [68, 69]. Batch Normalization is done by normalizing each inputs of the layers such that their mean will be zero and variance be one [70].

3.3 Related Terms in CNN

Some other terms associated with a CNN are introduced as below:

- **Overfitting and Underfitting:**

  If the model is performing better on seen data (training data) but worse on unseen data (validation data), then the model is said to be overfitting [71]. It is because the training data is too simple such that it is just memorizing

the data instead of learning [72]. If the model is performing poorly on the training data without being able to learn the relationship between input and target values, then it is said to be underfitting. It is because the input data is too simple to describe the target data. The overfitting and underfitting model can be shown graphically as in figure 3.5.

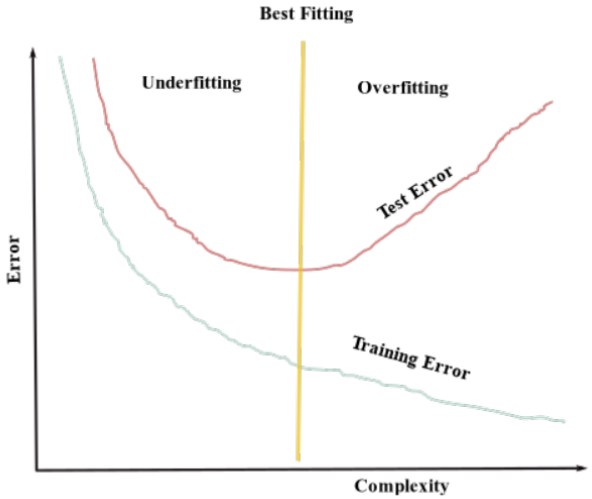


Figure 3.5: A schematic representation of a model performance in terms of error rate and complexity of the CNN layers

Overfitting is characterized by lower bias and higher variance while underfitting is by higher bias and lower variance [73]. The reasons for overfitting and underfitting of a model are due to the presence of noise, size of training data and types of classifiers [74]. The overfitting and underfitting problems can be addressed by three strategies:

- **Early Stopping:** An important question in model training is the duration i.e number of epochs to train the model. If it is very short then the model will suffer underfitting and will suffer overfitting if the duration is very long. So a compromise has to be reached by training the model up to the point (best fitting line in yellow color in fig 3.5) beyond which the model starts to overfit. This is monitored by an early stopping callback function with loss or accuracy and patience as parameters [73, 74].
- **Reduction of noise and outliers:** The noise and outliers are those variables which have little predictive power and only negate the quality along with the accuracy of the model [75]. As the models can only be as good as the quality of data used for training, the data

thus, should be provided such that it can be easily optimized by the model for learning and generalization. This optimization process is termed as feature transformation which tend to reduce the noise and outliers in the data [72].

– **Regularization:** Regularization is the process of adding information to find an optimal solution to a problem. It is used to prevent the overfitting of models and reduce their generalization errors [76, 77]. Particularly there are various regularization techniques like l1, l2, dropout and early stopping. L1 regularization tries to address the problem of overfitting by making the model sparse. L2 regularization is commonly used technique aims to minimize the sum of the square of differences between the label and predicted values [78]. It is also called weight decay and is given by the equation 3.5 [79].

$$S = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(y_{i,j} - f\left(x_{i,j}\right) \right)^2 \tag{3.5}$$

- **Loss Function:**

  Error of a model is the difference between the predicted values and their corresponding label values [80]. The function to compute such error is called loss function. In neural networks mean squared loss and cross-entropy loss are two commonly used loss functions.

  – **Mean squared loss:** It is the mean of the sum of the square of all the differences between target and predicted values given by the equation 3.6 [63, 81]

  $$MSE = \frac{1}{p} \sum_{i=1}^{p} (target(i) - predicted(i))^2 \tag{3.6}$$

  where i represents the i^{th} neuron.

  – **Cross-entropy loss:** Cross entropy loss is a loss function used in networks whose outputs are probability distribution. It is commonly used with classifiers like softmax and sigmoid [82]. The cross-entropy loss is given by equation 3.7 [63]

  $$F(x) = -\sum_{i} x_i^{'} log(x_i) \tag{3.7}$$

  where $x_i^{'}$ is the target value and x_i is the predicted value by the classifier.

Cross-entropy loss is of two types: binary and categorical. Binary cross-entropy is used for classification problems with two classes [83] and categorical cross-entropy is used for that with multi-classes.

- **Callbacks:** Sometimes the ongoing training process have to be influenced with certain tasks in different stages like start or end of an epoch, before or after of a mini-batch. In those case callback functions are used in the network. Early Stopping to find the right time to end training, model checkpoints to save the best models, reduction of learning rates on plateaus are some examples of callbacks in a CNN. Other tasks like viewing log files after every batch, observing the internal states, computing the statistics of models are also done with callback functions [84].

- **Sample, Batch and Epochs:** A sample is a single row of input dataset. Batch refers to the total number of samples fed to the network to update the network once. Batch size can be equal to a single sample or all the samples or a suitably chosen number of samples within the dataset. Number of Epochs is the number of times the learning mechanism works throughout the whole training dataset.

  For example, consider a dataset with 100 samples with the batch size of 20. Then the network is updated 5 times (5 batches) in a complete epoch. So if the epoch is 50, then it will pass through 50 times or 250 batches throughout the dataset during whole training process [85].

- **Optimizer:** Optimizer is an algorithm which changes the weights and learning rates of a network to minimize the loss after each batch of training [86]. Adaptive Moment Estimation (ADAM), Stochastic Gradient Descent (SGD) and RMSprop are among commonly used optimizers in CNN. ADAM is based on first and second order moments [87], SGD on gradient descent with momentum [88] and RMSprop on plain momentum [89]. Optimizers are used in conjunction with loss functions to compile a model.

- **Learning Rate:** Learning rate is the value used by optimizers to update the weights of attributes in a network. Learning rate is updated by a learning rate scheduler callback which takes the index of each epochs and current learning rate as inputs to return an updated learning rate as output [90]. Learning rate is also monitored by another callback called

reduce learning rate on plateaus. This callback updates the learning rate if the model does not improve after certain patience of epochs [91].

- **Metrics:** Metric is a function which numerically assesses the performance of a network [92]. Loss functions can also act as a metric of a model. Other metrics are Mean Squared Error, Mean Absolute Error, mean IoU error, classification metrics based on True/False positives negatives etc. Mean Squared Error is used to compute the mean of squared error [93], Mean Absolute Error to compute the mean of absolute error [94] and mean IoU to compute the mean of Intersection over Union [95] between the label and predicted data. Classification metrics based on true/false positives and negatives evaluate the performance in terms of precision, recall and accuracy [96]. [97] have defined these terms as in equation 3.8, 3.9 and 3.10

  - **Precision:** Precision refers to the ratio of correctly classified pixel to all the pixels classified into that category. Mathematically,

    $$Precision = \frac{TP}{TP + FP} \tag{3.8}$$

    where TP represents the true positive and FP represents the false positive.

  - **Recall:** Recall refers to the ratio of correctly classified pixels out of the total pixels of that category. Mathematically,

    $$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

    where TP represents the true positive and FN represents the false negative.

  - **Accuracy:** Accuracy refers to the ratio of total correctly classified pixels to the total number of pixels in the sample. Mathematically,

    $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.10}$$

    where TP represents true positive, TN represents true negative, FP represents false positive and FN represents false negative.

Chapter 4

Study Area and Dataset Preparation

This chapter talks about the study area of the research and the approach followed for dataset preparation. Section 4.1 talks about the geographical location and spatial extent of the study area. It also talks about geography of the terrain and biodiversity present in that part of the world. Section 4.2 gives a thorough overview of Image, DEM and Ground truth data preparation to make image-label pairs.

4.1 Study Area

The study area consists of 18 Terai districts of Southern plains of Nepal. It occupies about 28402.98 sq. km of territory in Everest Adjustment 1937 projection system of D Everest Bandladesh datum. The geographical extent is within 26.42^0 to 29.07^0 North latitude and 80.47^0 to 87.01^0 East longitude in WGS 1984 coordinate system of D WGS 1984 datum. The Terai is considered as the greenbelt covered with grasslands, tropical monsoon forests, savannah, clay and loam soil. With the 55.7% of total agricultural land within the range of 60m to 300m altitude, the region is the 'rice bowl' or 'agricultural production-house' of the country [98, 99]. Nearly 47% of total population inhabit in Terai region alone at an average population density of around 350 per sq. km [100]. It contains many seasonal and annual rivers mostly originated from the Siwalik hills on the northern side of the region. In terms of bio-diversity, the region

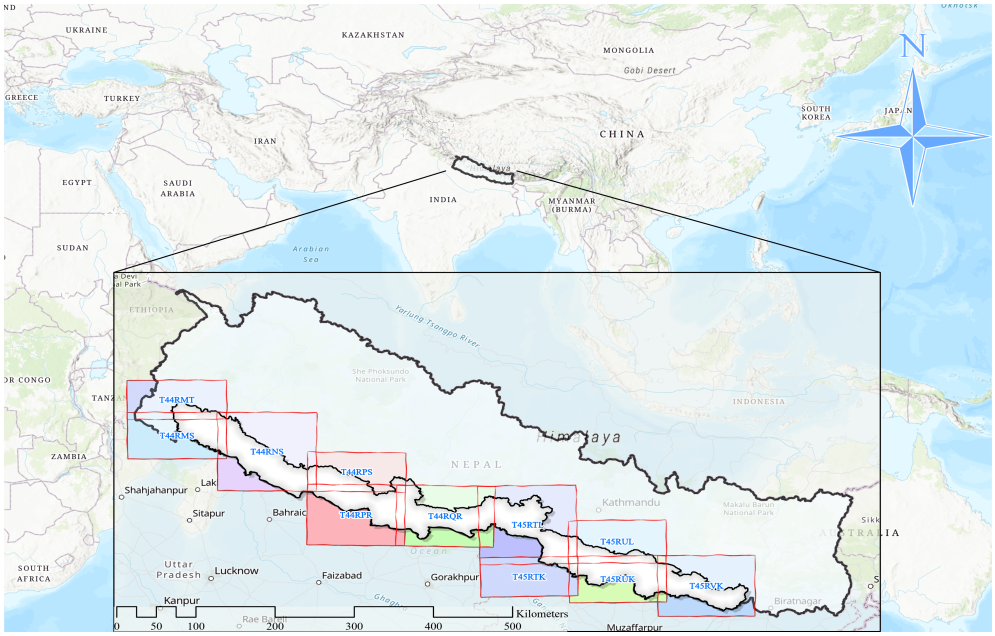


Figure 4.1: Study Area: Southern part of Nepal

is home to 35 species of mammals, 111 of birds, 46 of herpetos and 106 of fishes [99]. The region also feature 163 wetlands, 4 Ramsar sites and 2 World heritage sites [101]. Figure 4.1 shows the study area of the research and the tiles covering the area.

4.2 Dataset Preparation

Table 4.1: Bands information of a Sentinel-2 image

Bands	Spectrum	Resolution(m)
1	Coastal Aerosol	60
2	Blue	10
3	Green	10
4	Red	10
5	Vegetation Red Edge	20
6	Vegetation Red Edge	20
7	Vegetation Red Edge	20
8	NIR	10
8A	Vegetation Red Edge	20
9	Water Vapour	60
10	SWIR - Cirrus	60
11	SWIR	20
12	SWIR	20

Sentinel-2 images have 13 spectral bands in three spatial resolutions 10m, 20m and 60m. Blue, Green, Red and NIR bands are provided in 10m while four bands of Vegetation Red Edge and two bands of Short Wave Infrared (SWIR) are

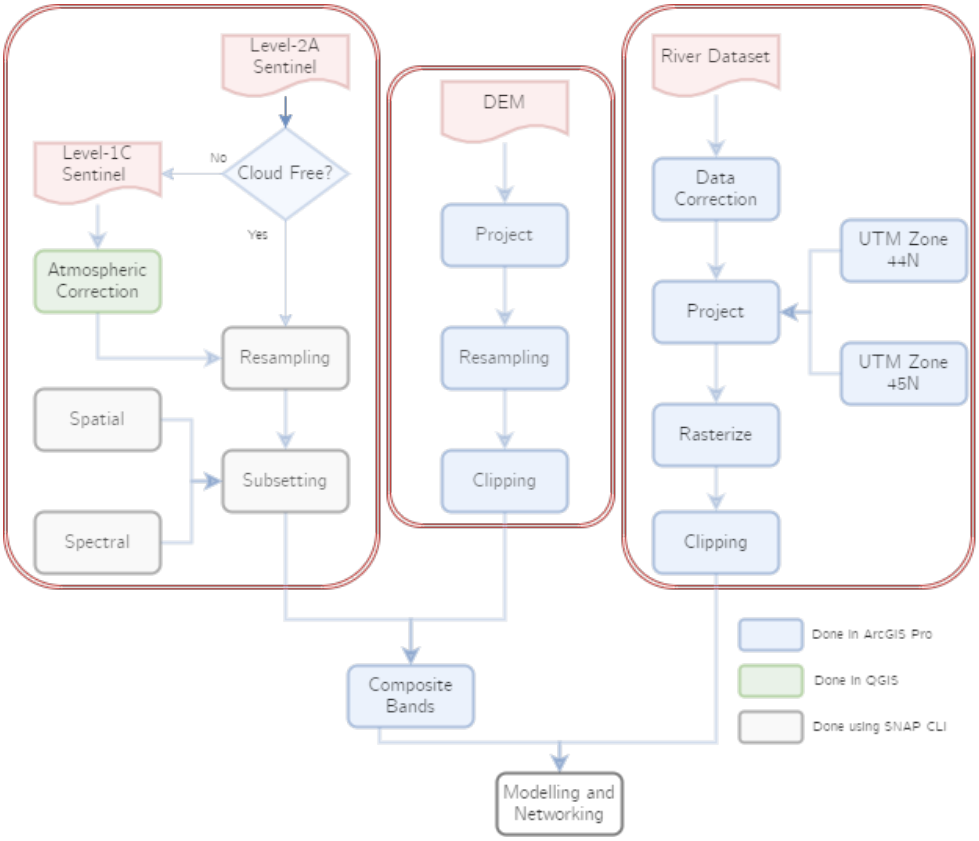


Figure 4.2: Preprocessing using SNAP Command Line Interface and ArcGIS Pro

provided in 20m resolution. Remaining bands Coastal Aerosol, Water Vapor and SWIR (Cirrus) are provided in 60m resolution [102].

4.2.1 Satellite Image Preprocessing and Preparation

Level 2 A Sentinel-2 image dataset was downloaded from ESA Copernicus website. A total of 11 cloud-free tiles (all from 2020) were downloaded which fully covered the study area. Preliminary inspection regarding the amount of cloud, cirrus, number of bands, spatial coverage etc were done for all the tiles. Few tiles (T45RUK, T45RUL) were missing some portion of image. Alternative tiles of other dates were downloaded and merged in order to compensate the missing areas. Some other tiles (T45RTL, T45RVK) were found to be having some incomplete bands (mainly band 6, 11 and 12). Such tiles were replaced with alternative scenes of other dates. It was difficult to find cloud- and cirrus-free scenes for some tiles at Level 2A products. Level 1c products of such tiles were downloaded from USGS earthexplorer website and converted into Level 2A product by applying DOS1 atmospheric correction at Semi-automatic Classification plugin in QGIS Desktop 3.14.15. After all the tiles were ensured to have

been converted from radiance values to surface reflectance values, following steps as shown in the figure 4.2 were continued using SNAP Command Line Interface. This could have been done in SNAP using Batch Processing Tools as well.

Graph Builder tool of SNAP was used to construct a sequential chain upto band subsetting for batch processing of all the tiles. The tiles were resampled to 10m and then subsetted reducing the spatial size of tiles and number of bands. Band 1 (Coastal/Aerosol band) and 10 (Cirrus band) were not required for our case so they were excluded for further analysis. After subsetting the tiles, band composition was done using ArcGIS Pro 2.6 to concatenate all the bands and DEM (from section 4.2.2) under same raster and exported into Tagged Image File Format (TIFF).

4.2.2 DEM Data Preparation

The 90m resolution DEM dataset based on Shuttle Radar Topographic Mission imagery (last updated on Nov 10, 2019) was acquired from United Nations Office of the Centre for Humanitarian Affairs Services. As satellite images had the UTM Projection system with Zones 44N and 45N, the DEM dataset was also projected accordingly and then resampled to 10m resolution. The DEM raster corresponding to the image tiles were extracted using Extract by Mask tools and pixel depth was used as 16 bit unsigned. Finally, the preprocessed DEM data was integrated as 12^{th} band with the image tiles prepared in the section 4.2.1.

4.2.3 Ground Truth Data Preparation

For ground truth data, River dataset (last updated on Nov 24, 2015) was acquired from UN-OCHA. As satellite images had the UTM Projection system with Zones 44N and 45N, the river dataset was also projected accordingly and converted into raster. River raster corresponding to the image tiles were extracted using Extract by Mask tools. Finally, thus preprocessed data were arranged in two folders for training and testing.

Chapter 5

Methodological Framework

This chapter deals with the flow of research works using traditional and neural network approaches. Section 5.1 first details about the implementation of neural networks. It starts with describing the determination of baseline model. Further, it delineates how the chosen five networks: SAPCNN, CNNWQC, CNNCWC, denseNet and attResNet were configured and implemented. Theoretically, the working environment and methodology of these models were not the same. But for experiment purpose, the core aspect of these models were extracted and uniformly used for comparison. Section 5.2 finally, explains the approaches how water features were extracted using four chosen traditional methods: NDWI, NDVI, NDVI_NDWI and EWI. It thoroughly details on the derivation process of the equation for EWI.

5.1 Implementation of Neural Networks

The implementation of neural networks for this research began with the determination of baseline model. Then remaining five models were thoroughly implemented with the hyperparameters from table 5.1. Same hyperparameters were used for training in order to make uniform comparison for all the networks.

Table 5.1: Hyperparameters used in the experiments

Random State	Validation Size	Test Size	Patch Size	Step	Batch Size	Epochs	Learning Rate	Patience	Momentum
1	0.2	0.4	8,12,16,20	8	128	100	0.001	15	0.9

5.1.1 Determination of Baseline Model

With the hyper-parameters as presented in table 5.1, baseline model was decided on hit and trial basis by adding and removing the convolutional layers. Considering the accuracy achieved in lesser time, it was decided that two layers of convolution would be the efficient one. The architect of the baseline model can be found in figure 5.1. It consisted of two sequences of convolution and max-pooling layers with one fully connected layer and a final output layer. The convolution layer was meant for feature extraction and max pooling for reducing the size of feature maps to increase the computational efficiency.

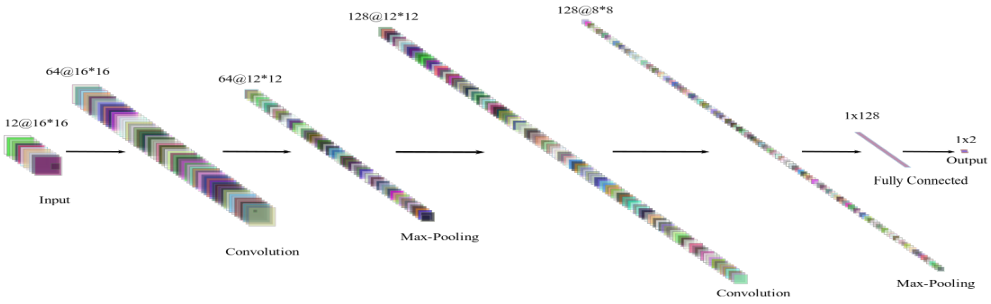


Figure 5.1: Architecture of baseline model

5.1.2 Implementation of SAPCNN

SAPCNN consisted of two sequences of convolutional layer with 5*5 kernel size and 2-D max pooling with 1*1 stride. Flattening and Dense layers formed the final layers with a 10% dropout introduced between the two dense layers. Instead of converting the pixels into superpixels and using self adaptive pooling as proposed by [5], simply patch based extraction method and max pooling were used. The implemented architecture of this network can be seen in the fig 5.2.

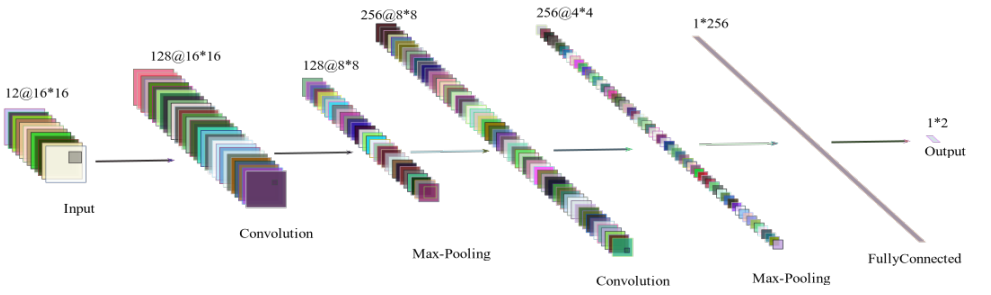


Figure 5.2: Network Architecture of SAPCNN

5.1.3 Implementation of CNNWQC

CNNWQC consisted of four stacks of convolutional layers followed by two fully connected layers and a final output layer. Uniform kernel size of 3*3 was used in all the layers with 50% dropout on the fully connected layers. As proposed by [34], poolings layers were not used to preserve information. Stride of 1*1 was used with ReLU activation in all the layers except the final one where softmax was used to compute the probability of each class. The architecture of CNNWQC can be viewed in figure 5.3.

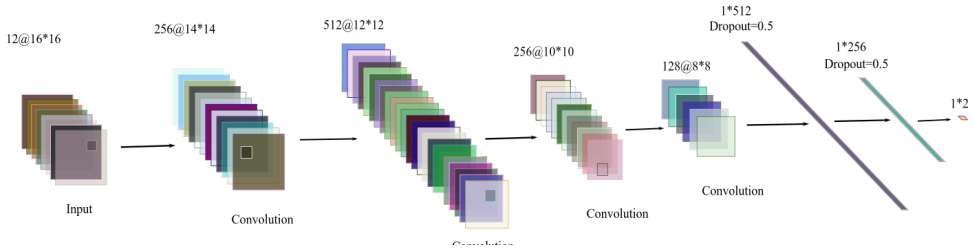


Figure 5.3: Network Architecture of CNNWQC

5.1.4 Implementation of CNNCWC

CNNCWC consisted of two sequences of convolutional and max-pooling layers followed by three convolutional layer, a max-pooling layer, two fully connected layer and a final output layer. The first convolutional layer had 11*11 kernel size, the second one had 5*5 while the rest had the same size of 3*3. The max-pooling layers had uniform kernel of 3*3 in all layers. All the convolutional layers had ReLU activation while the fully connected ones had tanh and the output layer had sigmoid activation. The architecture of CNNCWC can be viewed in figure 5.4

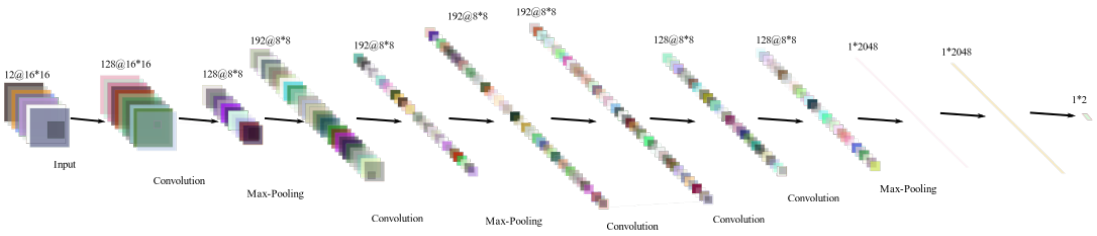


Figure 5.4: Network Architecture of CNNCWC

5.1.5 Implementation of DenseNet

Figure 5.5 shows the DenseNet architect which consisted of an initial layer of convolution and Batch Normalization followed by a sequence of three dense

blocks with two transition layers between the dense blocks. The dense blocks were concatenated with dense connectivity of multiple inputs into a single tensor. These inputs were fed into sequential operation of Batch Normalization, Rectified Linear Unit and convolutional layers of $3 * 3$ kernel. A computationally efficient convolutional layer of $1 * 1$ kernel was introduced as bottleneck before the larger convolutional layer$(3 * 3)$ to reduce the number of feature maps. The transition block continued with an average pooling layer with size $2 * 2$ to down sample the size of the feature maps. It then followed with a Batch Normalization layer. The dense blocks and transition blocks were followed by ReLU activation layer. A global average pooling was done to the results before passing them to final output layer activated by softmax.

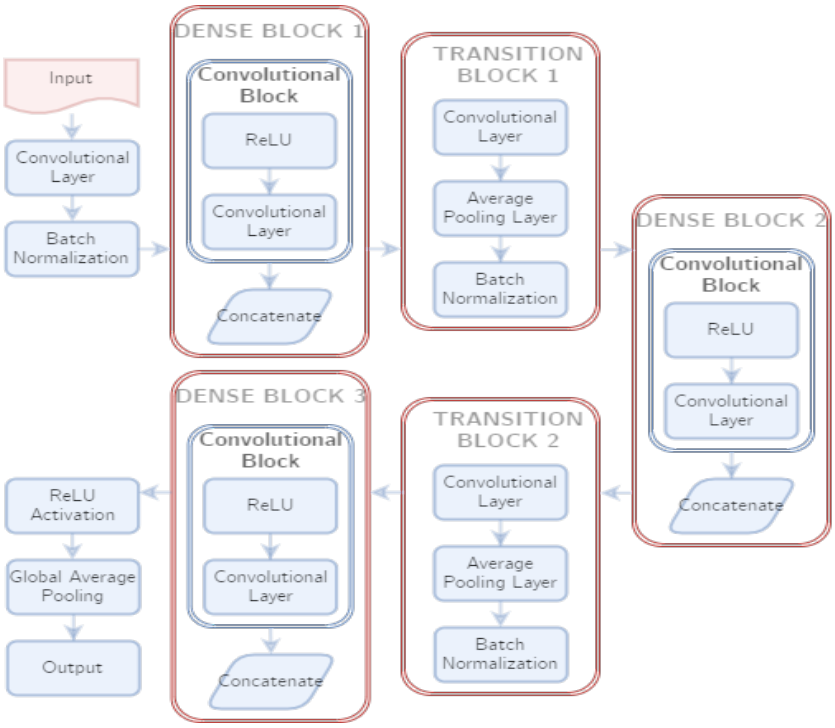


Figure 5.5: A schematic representation of DenseNet as implemented

5.1.6 Implementation of AttResNet

Figure 5.6 depicts AttResNet architect which consisted of convolutional, batch normalization and max-pooling layers as initial layers followed by three sequences of residual and attention blocks. Further, it consisted of three residual blocks followed by average pooling layer. The results from these blocks were flattened before putting them through the final layer for classification of water patches.

The residual blocks were composed of three blocks of batch normalization, ReLU activation and convolutional layers with 3*3 kernel size. The attention

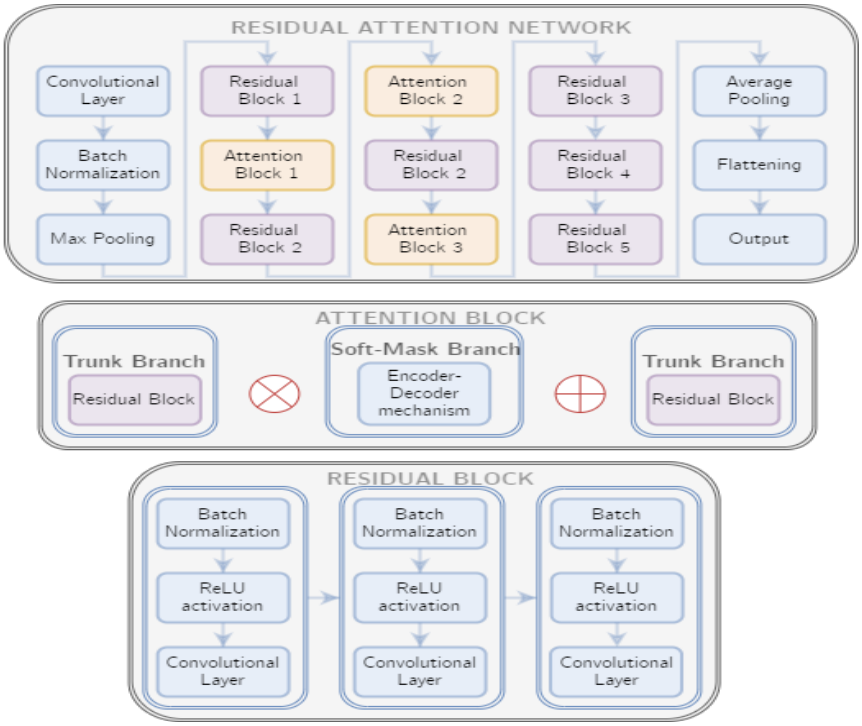


Figure 5.6: A schematic representation of AttResNet as implemented

network was integrated with residual blocks but segregated as trunk branch and soft mask branch with encoder-decoder mechanism. Down sampling in encoder branch was done by 2D max-pooling which was up-sampled by 2D up-sampling layer in decoder branch. The final layers contained two 1*1 convolutional layers with sigmoid activation.

5.1.7 A novel CNN approach for water bodies identification

Taking into account the problem and challenges described in section 1.2 and this work, a novel CNN architecture is also proposed. In order to exploit the benefits of both DenseNet and AttResNet, a new approach was developed by integrating both. The novelty of the proposed network lied in its composition. The proposed architecture consists of dense blocks and transition blocks from DenseNet along with attention block and residual block from AttResNet. All these blocks possessed own peculiarities to strengthen the proposed network. The dense block was supposed to address the problems of vanishing gradients and information loss from deep layers. The transition layer was expected to enhance the computational efficiency before the input is fed to computationally costlier dense blocks. Similarly the attention layer was aimed at paying attention to water features by suppressing non water bodies and residual layer for representing water bodies in a better way inside the deep layers. With these objectives in mind, a design for a novel network was proposed by integrating

attention block before feeding the inputs to dense blocks in DenseNet. The input data was fed through first layer of convolution and Batch Normalization before passing it to the attention block. The attention block consisted of trunk branch and soft-mask branch. The trunk branch possessed a residual block for feature representation and the soft-mask branch contained encoder-decoder algorithm to extract features. In summary, the attention block determined the areas of focus and passed the information to the regular DenseNet. The schematic diagram of the proposed network can be viewed in figure 5.7

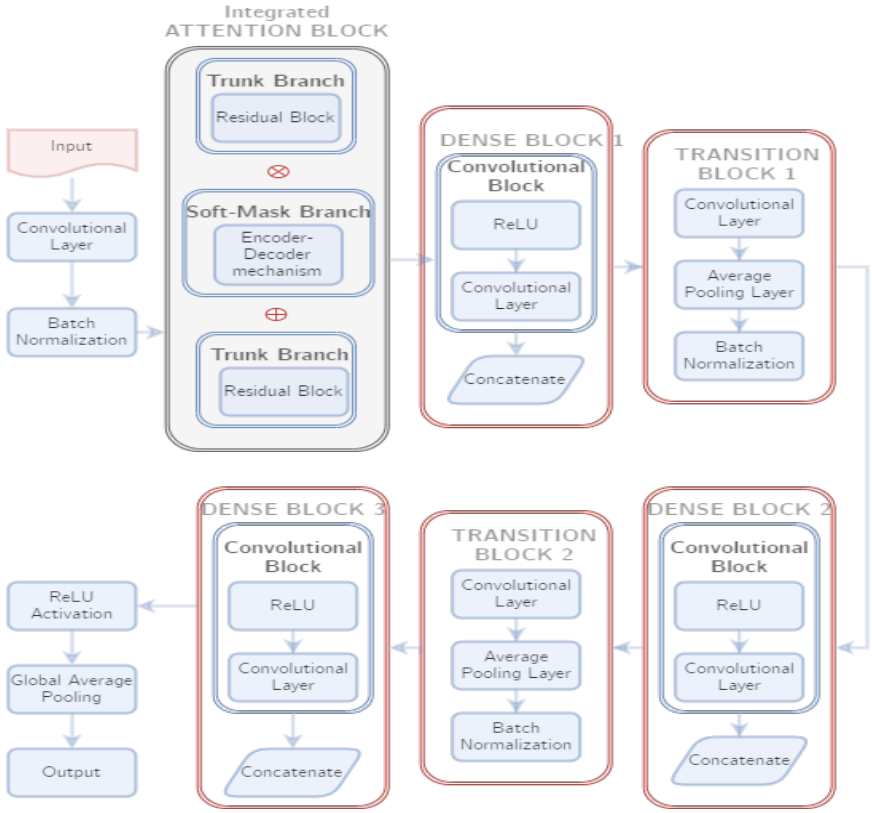


Figure 5.7: A schematic representation of the proposed network by integrating an attention block (grey background and black border) in DenseNet

5.2 Implementation of Index-based Methods

Four different water-index methods; NDWI, NDVI, NDVI_NDWI and EWI were implemented for this research. Water extraction was based on the equations of respective indices as discussed in section 2.1. To make fair comparison, similar approach of patch extraction and classification as in section 5.1 was used. Prediction was also made on the same tile T44RQR for both approaches. Performance of the indices for patch sizes 8, 12, 16 and 20 were computed. NDWI, NDVI and NDVI_NDWI already had predetermined band equations

and hence were pretty intuitive. To develop the EWI, the approach as proposed by [25] was followed thoroughly as discussed in section 5.2.1.

5.2.1 Development of EWI

The EWI was developed according to the procedure recommended by [25] taking the conjugative ratio of NDWI with Principal Components. At first, NDWI for tile T44RQR was computed from the equation 2.3. Then it was dimensionally reduced to 11 principal components. A total of 400 sample points 100 for each of water, forest, barren and urban were extracted to find the average spectral reflectance values for each features. The average spectral values of these sample features were plotted accordingly. The spectral graphs were assessed to derive and validate the EWI equation proposed by the [25]. Finally with this equation water features from T44RQR tiles were extracted. The overall EWI derivation process can be seen in the figure 5.8

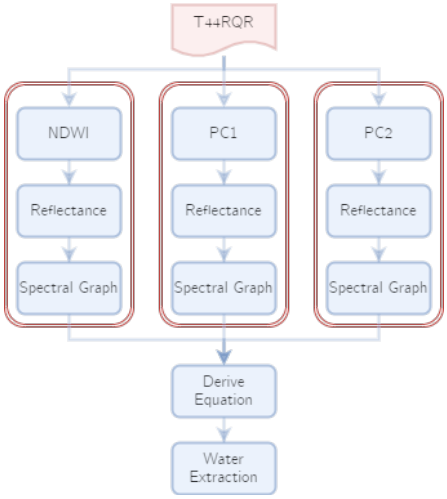
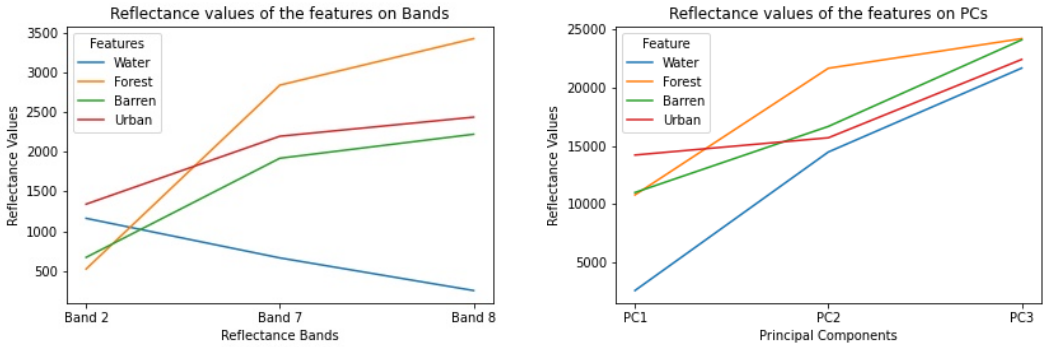


Figure 5.8: Steps for derivation of Enhance Water Index

The Spectral Graph shown in figure 5.9a shows the gradual reduction of reflectance values of the four features on bands 2, 7 and 8. Among all the features, the minimum reflectance was of water at band 8 and the maximum one was of Forest at band 8. Urban and Barren features nearly shared similar reflectance values at bands 7 and 8. However, Barren had lesser reflectance than urban at band 2. Unlike other features water was found to be following the decreasing trend of reflectance values with increasing serial number of bands. Water features reflected the most on band 2 and the leas on band 8. Since NDWI was calculated as conjugative ratio of the difference between band 2 and band 7, only water features received positive values for NDWI. Other features received negative values with forest having the least value.



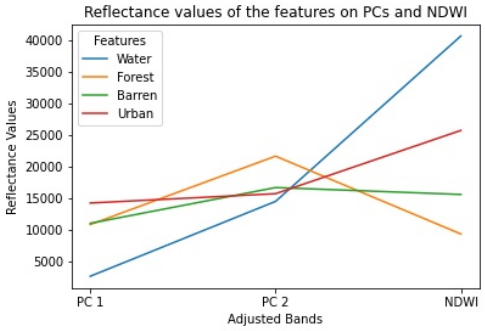
a on bands 2,7 & 8 b on PC1, PC2 & PC3

Figure 5.9: Spectral properties of water, forest, barren and urban features

The figure 5.9b, presents the spectral properties of the features on the first three principal components. It depicted that the reflectance values of all features had growing trend from PC1 through PC3. The minimum reflectance was of water on PC1 and the highest was of Forest and Barren on PC3. Water and urban shared similar reflectance properties throughout PC2 and PC3. Forest had the highest reflectance in PC2 and urban in PC1.

PC	EigenValue	Percent	Cumulative
1	1.30E+06	60.9533	60.9533
2	6.53E+05	30.6462	91.5994
3	9.07E+04	4.2551	95.8546
4	3.70E+04	1.7379	97.5925
5	2.08E+04	0.9774	98.5699
...	...	...	...
...	...	...	...
11	1.26E+03	0.0592	100

(a) Percentage and Cumulative Eigen Values



(b) Spectral properties of water, forest, barren and urban features on PC1, PC2 and NDWI

Figure 5.10a presents the percentages and cumulative values of original information accumulated by each pcs. It was found that PC1 and PC2 accumulated more than 90% of the information. PC3 shared only about 4.25% of original information, so it was safely replaced with NDWI as third band as presented in figure 5.10b. It shows that NDWI for water was the highest in comparison to rest of the features. Hence the sum of PC1 and PC2 differentiated from NDWI gave positive values for water features only.

Hence as proposed by [25], the same equation 5.1 was found to be working for our research.

$$EWI = \frac{\rho_{NDWI} - (\rho_{PC1} + \rho_{PC2})}{\rho_{NDWI} + (\rho_{PC1} + \rho_{PC2})} \tag{5.1}$$

Using the equation 5.1, water features from T44RQR tile were extracted and visually compared with its RGB tile.

Chapter 6

Results and Discussion

This chapter describes a thorough analysis of the extraction of water bodies using neural networks and index-based approaches. Section 6.1 details about the experimental setup regarding the hardware and software. It also delineates about the patch extraction process, balancing labels and determination of hyperparameters for training neural networks. Section 6.2 explains all the experiments conducted using neural networks. It presents the four variants of experiments which were conducted using varying channels of Sentinel-2 images. It also details on how the chosen networks were integrated to propose a new architect. Section 6.3 describes the experiments using index-based approaches conducted to compare with neural networks. It presents quantitative assessments of three indices (except EWI). It also discusses about the comparative effectiveness of neural networks over the index-based ones. Section 6.4 discusses the visual quality of outputs delivered by both neural networks and index-based approaches. It also compares the visual performance on two selected regions of the tile T44RQR. Later it analyses the performance by the proposed network on those regions. It also explore the areas where proposed network performed better than others.

6.1 Experimental Setup

The experiments for this research were performed on a server having Intel(R) Core (TM) i7-6850K processor with 110GB Random Access Memory (RAM) and two Graphics Processing Unit (GPU)s; GeForce RTX 1080 Ti 11GB and GeForce

2080 Ti 11GB. Keras-gpu 2.3.1 and tensorflow-gpu 2.1.0 were installed on Anaconda Framework with Python 3.6. Rasterio 1.1.7 was used to read and write image tiles and labels data for patch extraction and to save the predicted output into Geotagged Image File Format (GeoTiff).

The figure 6.1 shows the general steps followed to extract patches and labels to be fed into the networks.

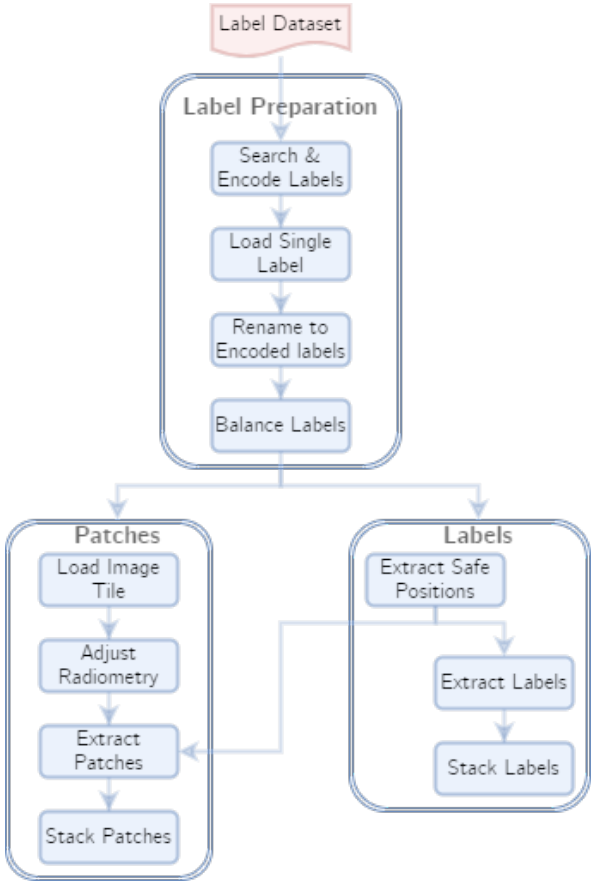


Figure 6.1: Steps of Patches and Labels extraction process

6.1.1 Patch Extraction

The input data for the network consisted of n number of p*p sized patches extracted from image tiles. Different patch sizes ranging from 8 to 48 were tested to determine the ideal sizes from conducting experiments. According to [103], larger patch size increases the accuracy of the network. It was also verified from the figures 6.2 and A.1a, but larger size was found to have higher computation cost of the hardware. Moreover, they were found to be vulnerable to overfitting due to greater volume of information. Small water-bodies were also not represented very well with higher patch sizes. So it was determined that patch sizes of less than 24 particularly 20, 16, 12 and 8 to be used in the

experiment of comparing the networks. A total of 11 image-label pairs were prepared out of which two tiles were taken out for testing and prediction. Out of remaining nine pairs, total extracted patches were distributed such that 60% was assigned for training and 40% for testing. Out of 60% of training patches 20% were assigned for validation. Validation set was used for improving the hyperparameters and test set for computing the confusion matrix. The performance of the network was monitored with validation loss and accuracy in comparison with training loss and accuracy.

6.1.2 Balancing labels

The ratio of non water to water pixels of total study area was around 30:1 due to which non water pixels would dominate the prediction of water pixels. To address this issue, labels balancing was done before feeding the patches to the network. One thing to note from figure A.1b was, the accuracy became higher when non water to water ratio was high but the recall value of water became lower. So an optimum values for accuracy and water recall was negotiated by keeping the non water to water ratio as 2:1. This was done by separately extracting and indexing the positions of non water and water from label data. Then the indices of non water was randomly shuffled and twice the number of water pixels were stacked in the final label. Patches of defined sizes corresponding to the positions of extracted labels were stacked.

6.1.3 Determination of hyperparameters

Table 6.1: Hyperparameters on which series of initial experiments were conducted

Hyperparameters	Patch Size	Step	Learning Rate	Patience	Batch Size	Epochs
Experiments	8,12,16,20, 24,32,40,48	4,8,12,16	0.01,0.001, 0.0001,0.00001	7,10,15, 20,25	32,64,128, 256,384,512	40,75,100, 150,200

Table 6.1 shows the instances of hyperparamters on which series of initial experiments were conducted. Patch size was determined according to the reasons explained in section 6.1.1. The accuracy was found to be increasing with the reduction in step size due to increment in the number of samples extracted (see figure A.1c). But this also effectively increased the computational cost. So an optimum value of 8 was chosen. Learning rate of 0.01 was found to be the fastest one to converge and 0.00001 to be the slowest one such that accuracy was also compromised. Also AttResNet was found to be performing better with slower learning rates in contrast to the rest. While 0.001 was found to be

the best among the tested ones (see figure A.2) in Appendix A, so the learning rate was safely fixed to 0.001 with the provision of reduction on error plateaus. Patience of size 7 was found too quicker to terminate the model as it was found that models improved even beyond that. So it was fixed to 15 considering that there was provision to alter the learning rate if error plateaus occur. Higher batch sizes smoothened the learning curves by reducing the local noise but it compromised the accuracy achieved. Hence 128 was chosen and finally number of epochs was chosen based on the maximum possible accuracy it could reach. It was decided that 100 would be the suitable one considering the time it may consume. Other parameters like momentum, growth rate, weight decay specific to a model were used as prescribed by the respective models.

6.2 Experiments conducted on neural networks

Four experiments were conducted to assess the performance of the neural networks. The channels were fed as inputs to the networks in four ways; RGB, RGB with DEM, selected S2 channels and S2 channels with DEM. The networks were implemented using patch sizes of 8, 12, 16 and 20. The performance was monitored with test accuracy and recall values on test set for quantitative assessment during the experiments. Other metrics like precision and f1-score also are appended in appendix B for the reference of readers. For qualitative assessment, comparative visual interpretation of the predicted maps were done.

6.2.1 Use of RGB Channels

The performance of the networks while using RGB channels only can be found in the table 6.2.

Table 6.2: Performance of neural networks on RGB channels

Channels	Models	Patch Size											
		8			12			16			20		
		Test	Recall		Test	Recall		Test	Recall		Test	Recall	
		Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water
RGB	baseline	80.83	64	89	83.45	68	91	83.98	71	91	84.80	73	90
	CNNWQC	81.98	68	90	84.58	74	90	84.90	74	90	84.93	74	90
	CNNCWC	No convergence			81.90	66	90	82.91	68	90	83.20	71	89
	SAPCNN	81.86	68	89	84.08	74	89	84.92	75	90	85.63	77	90
	denseNet	82.72	71	87	84.77	75	89	85.41	75	89	85.95	78	91
	attResNet	79.84	66	87	NA			82.29	71	88	NA		

The performance of the neural networks while using only RGB channels as input was found to be the least among all experiments on neural networks. The baseline model produced test accuracy of 80.83% with 64 and 89 as recall

values for water and non water respectively while DenseNet achieved the best results for all models in all patch sizes. The general trend was increment in performance with the increase in patch sizes. So the best performance was achieved by DenseNet on patch size 20. The test accuracy was 85.95% with 78 and 91 as recall values for water and non water respectively. AttResNet did not fit with patch size 12 and 20 while CNNCWC did not converge at patch size 8 due to lesser information on the smallest patch size. CNNWQC was found to be performing better than SAPCNN on smaller patch sizes 8 and 12 while the later outperformed the former on pathc sizes 16 and 20.

6.2.2 Use of selected S2 Channels

The performance of the networks while using selected S2 channels (channels 2,3,4,5,6,7,8,8A,9,11 & 12) as input can be found in the table 6.3.

Table 6.3: Performance of neural networks on selected S2 channels

Channels	Models	Patch Size											
		8			12			16			20		
		Test	Recall		Test	Recall		Test	Recall		Test	Recall	
		Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water
RGB	baseline	87.14	78	92	87.84	79	92	88.47	81	92	89.52	84	92
	CNNWQC	87.81	81	92	88.70	82	92	89.30	84	92	89.35	84	92
	CNNCWC	85.73	72	93	86.88	78	91	87.71	79	92	87.72	81	91
	SAPCNN	87.48	80	91	88.19	81	92	89.44	84	92	89.09	83	92
	denseNet	88.16	81	91	89.03	83	92	89.53	84	92	89.60	84	92
	attResNet	86.23	77	91		NA		87.14	81	90		NA	

With the use of 11 channels of Sentinel-2 imagery, the performance of the networks increased by around 5% respectively in all the patches as can be observed from tables 6.2 and 6.3. The baseline model achieved the test accuracy up to 87.14% with recall of water and non water as 78 and 92 respectively at patch size 8. It reached to 89.52%, 84 and 92 respectively for patch size 20. Similar to the finding from table 6.2, DenseNet again was the best performer in all patch sizes among the networks in this experiment as well. The general trend was the increment in accuracy and recall values with the increase in patch size but this time SAPCNN and DenseNet obtained best outputs in patch size 16. After patch size 16, the best performance was at patch size 20. Again CNNCWC was found to have the least accuracy followed by AttResNet and SAPCNN respectively. DenseNet outperformed the rest networks this time as well followed by CNNWQC.

6.2.3 Impact of DEM Integration

To find out whether the introduction of DEM improves the water feature extraction or not, an experiment was conducted with baseline model by feeding DEM integrated with RGB channels. The result is tabulated in the table 6.4.

Table 6.4: Performance of baseline network on RGB channels integrated with DEM

Channels	Models	Patch Size											
		8			12			16			20		
		Test	Recall		Test	Recall		Test	Recall		Test	Recall	
		Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water
RGB	baseline	83.09	65	92	84.45	71	91	85.44	75	91	85.87	76	91

Upon the head to head comparison of tables 6.2 and 6.4, the performance of baseline model was found to increase by 2.26% at patch size 8. This gradually reduced in magnitude when it reached to patch size 20 at which the increment was only 1.07%. It depicted that though not drastically, the DEM still was contributing to some extent for the improvement of water bodies extraction. Hence, it was decided to consider DEM as 12^{th} channel for further experiments. Besides, the baseline model had the best performance in patch size 20 reaching the test accuracy of 85.87% with recall values for water and non water to 76 and 91 respectively.

6.2.4 Use of selected S2 channels integrated with DEM

The performance of the networks while using selected S2 channels integrated with DEM can be found in the table 6.5.

Table 6.5: Performance of neural networks on selected S2 channels integrated with DEM

Channels	Models	Patch Size											
		8			12			16			20		
		Test	Recall		Test	Recall		Test	Recall		Test	Recall	
		Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water
RGB	baseline	87.61	79	92	88.40	80	92	88.65	82	92	89.37	83	92
	CNNWQC	88.42	82	92	89.26	83	92	89.34	84	93	90.23	85	93
	CNNCWC	86.52	77	91	87.75	80	92	88.01	81	92	88.51	82	92
	SAPCNN	88.23	81	92	89.05	83	92	89.19	83	93	89.83	85	92
	denseNet	88.88	82	92	89.58	84	92	89.73	85	92	90.41	86	92
	attResNet	86.90	80	91		NA		87.67	81	92		NA	

With the integration of DEM as 12^{th} band in the input image, the accuracy was found to be improved slightly by around 1%. In case of baseline model at patch size 20 and SAPCNN at 16, the integration of DEM did not prove to be beneficial. This showed that DEM had lesser contribution to the extraction process than the spectral properties obtained by addition of channels. Overall, DenseNet was again the best performer with metrics ranging from 88.88%

through 90.41% of test accuracy and 82 through 86 values of recall for water when the patch size was increased from 8 through 20. CNNCWC, as always performed the least with the AttResNet following it. The performance of DenseNet was followed by CNNWQC and SAPCNN respectively. Unlike previous experiments, this time the performance was the best for patch size 20 for all the models. Due to the compatibility issues, unfortunately the performance of AttResNet could not be assessed at 8 and 20 sized patches.

6.2.5 Selection of Networks

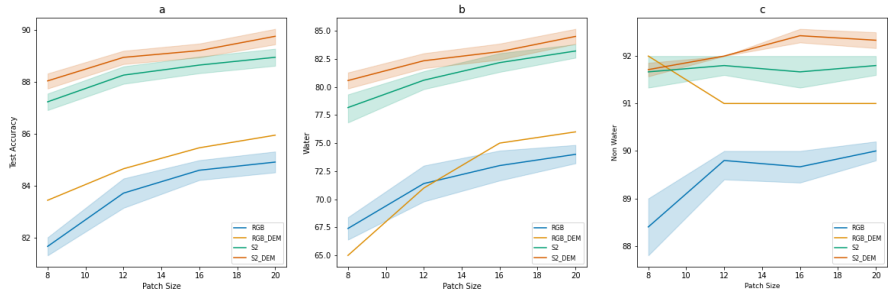


Figure 6.2: Relationship of patch size with performance metrics; a. on test accuracy, b. on recall of water & c. recall of non water using four different channels

Figure 6.2 depicts the effect of patch size on the performance metrics; test accuracy and recall values for water and non water. The values depicted by line graph was obtained by averaging the individual test accuracy of the six networks. It can be observed that the patch size 8 for RGB possessed the least value for test accuracy. Upon the increment in channels for the same patch size, the accuracy also increased, thereby RGB_DEM reaching the highest accuracy. Similarly, on increasing the patch size up to 20, the test accuracy was found to be directly proportional to it. Hence, the RGB_DEM channel reached the highest accuracy at patch size 20. Talking about the recall values, RGB_DEM at patch size 8 was found to be having the least value of recall for water but nearly the highest (92) recall for non water. For the recall of water of rest channels, it followed the trend of increasing in values on the increment of the number of input channels. Interestingly the recall values of non water for S2 channels was found to be similar for all patch sizes.

Figure 6.3 shows the performance of different models on increasing the size of patch. The line plots were obtained by averaging the values of four channels on the respective patch sizes. It was obtained that baseline had the least average test accuracy among all the networks followed by AttResNet. Unlike others, the

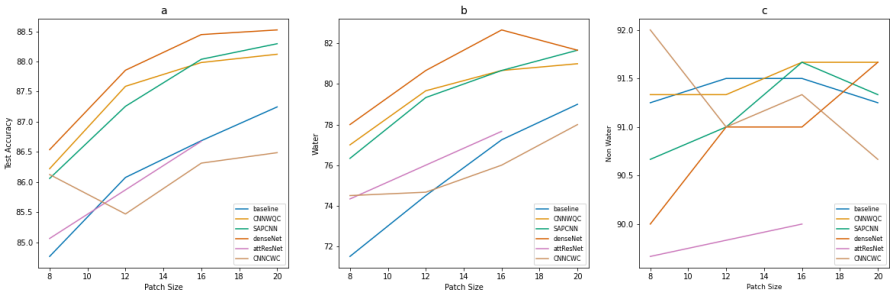


Figure 6.3: Ranking of of models on different patch sizes in terms of; a. test accuracy, b. recall of water & c. recall of non water

performance of CNNWQC was zig-zag shaped while for the rest the trend was that test accuracy increased with increase in patch size. DenseNet was found to be having the highest test accuracy across all patch sizes while CNNWQC and SAPCNN following it. Observing the recall for water, the same trend was found with DenseNet with highest value and baseline being the lowest. Interestingly, the recall values for DenseNet and SAPCNN at patch size 20 was the same. But the figure 6.3c shows that the recall for non water of DenseNet was greater that that of SAPCNN due to which in overall test accuracy, DenseNet won the contest. In case of attResNet, it was pity that it could not run with patch size 12 and 20 but with what was obtained it can be inferred that the test accuracy and both recall values increased with increase in patch size.

Overall it was observed that DenseNet performed the best in comparison to the rest networks in all the experimental conditions. Though AttResNet was not performing satisfactorily, the reason was inferred due to the common configuration used to run all the models. Necessity was felt that it needed different configuration to make it converge better than the baseline model. But AttResNet was experienced to be the slowest network consuming a lot of computational power. Considering the time constraints and limited hardware efficiency, it was decided to simply integrate an attention component from AttResNet with DenseNet to develop a new model for the study area. Hence, an attention block was integrated just after the Batch Normalization layer of DenseNet to create 'attention aware features'. It was also decided to choose patch size 16 for the reason that the test accuracy was found to be increasing with increasing patch size but 20 was not compatible for executing the AttResNet. Later the network was adjusted to make it work for other patch sizes as well.

6.2.6 Development and Performance of Proposed Network

A series of experiments were conducted to choose the number of dense blocks, depth of attention blocks and hyperparameters for the proposed network etc. The experiment was done with patch size 16, step 8 with non water to water ratio as 2:1. Other hyperparameters except the ones stated in the table 6.6 were kept constant.

Table 6.6 depicts the different instances of model execution by integrating the attention block with the DenseNet to propose a new network with highest quantitative performance. It was found that the network with 5 dense blocks provided the best test accuracy of 89.63% with recall of water 84. But considering the recall value of water being 85 and computational cost necessary to achieve a slight advantage of only 0.07%, 3 dense blocks architecture was chosen. Hence, the 3 dense blocks integrated with attention depth 1 executed at batch size of 64 and learning rate 0.01 was expected to provide the best performance. The performance of the proposed network reached to 90.29% with 86 and 93 as recall values of water and no water respectively. This is 0.56% more than the test accuracy obtained by DenseNet with patch size 16 in table 6.5. In comparison to the AttResNet on the same patch size, the proposed network yielded 2.62% more test accuracy than the former.

Thus, the numerical performance of the proposed network was found to be the largest among all the experimented models. The novelty of this network vis-a-vis its composition could be highlighted as:

- **Dense Blocks:** It checked the information loss and vanishing gradient problems within the deep layers.
- **Transition Blocks:** It made the training process computationally efficient by down sampling and reducing the size of feature maps before going for computationally huge dense block layers.
- **Attention Block:** It aided the network to 'pay attention' only to the concerned layer to create 'attention aware' water features.
- **Residual Block:** The residual blocks within the attention block supported the network by representing the features in deeper layers efficiently.

This proved that with the integration of strengths of two novel architects; DenseNet and AttResNet, a new architect can be developed with better numerical performance. To the knowledge of researchers, this is the first implementation of two novel architects DenseNet and AttResNet integrated together and

Table 6.6: Experiments to determine the hyperparameters for the proposed network

Parameters		test accuracy	Recall		Remarks
			Water	No Water	
Dense Blocks	2	88.73	83	91	Learning Rate 0.001
	3	89.56	85	92	
	4	89.54	84	92	
	5	89.63	84	92	
Attention Depth	1	88.89	82	92	Learning Rate 0.001
	2	88.49	82	92	
	3	88.01	81	91	
Batch Size	32	89.85	85	92	Learning Rate 0.01
	64	90.29	86	93	
	128	90.05	85	93	

that too for water bodies extraction.

6.3 Comparison with Index-based approaches

Four different indices; NDWI, NDVI, NDVI_NDWI & EWI were used to extract the water features from the imagery. The performance was expressed quantitatively in case of NDWI, NDVI and NDVI_NDWI. Due to the memory issues with the hardware to compute the principal components necessary for EWI, it was assessed qualitatively only.

Table 6.7: Performance of index-based approach to extract water

Indices	Patch Size											
	8			12			16			20		
	Test	Recall		Test	Recall		Test	Recall		Test	Recall	
	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water	Accuracy	Water	No Water
NDWI	74	51	97	74	51	97	75	52	97	74	51	97
NDVI	74	52	97	75	52	97	75	52	97	75	52	97
NDVI_NDWI	73	49	98	73	49	98	74	50	98	73	49	98

Table 6.7 presents the accuracy assessment of water extraction using NDWI, NDVI and NDVI_NDWI. The maximum accuracy reached for test was 75 with recall values of water at 52 and no water at 98. In comparison to the least accuracy obtained from the baseline model in table 6.2, it was still 5.83% less than the output of neural networks. Figure 6.4 depicts the graphical vision of their performance. It was found that the performance of all the indices were more or less similar to each other across all patch sizes. Specifically, NDVI slightly edged the rest indices in terms of all the metrics. NDWI was found to be have exactly the same performance as of NDVI at patch size 16. The recall values of NDVI_NDWI was the highest i.e 98 and other two had 97 throughout all the patch sizes.

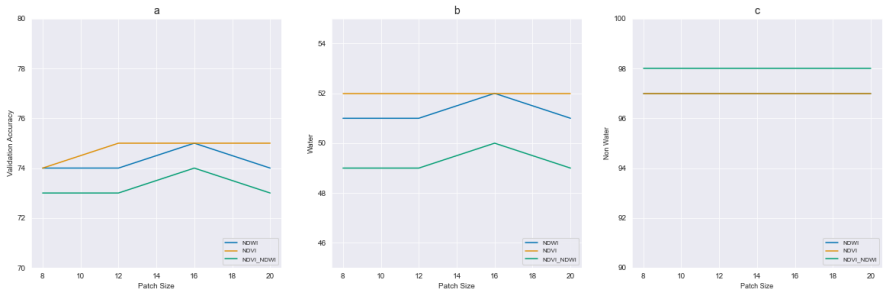


Figure 6.4: Performance of the three indices on different patch sizes in terms of; a. test accuracy, b. recall of water c. recall of non water

6.4 Qualitative Assessment of the performance

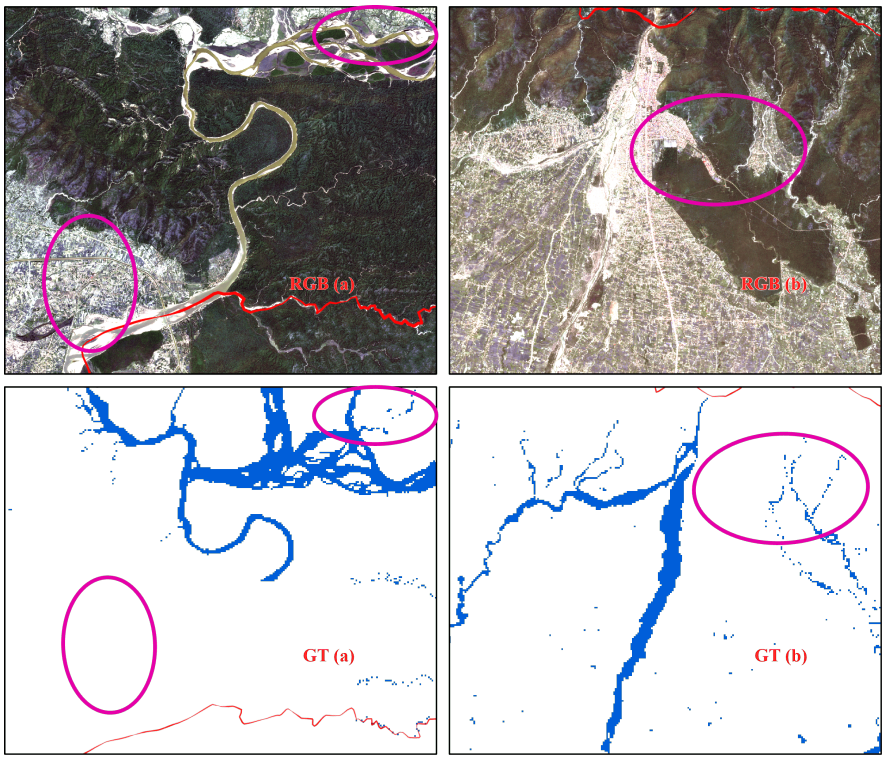


Figure 6.5: RGB and Ground Truth images at selected regions (by ellipse in magenta)

From the quantitative analysis of the performance of neural networks, it was found that DenseNet produced the best results. Thereby taking an attention block component from AttResNet and integrating with DenseNet, a new network was also proposed. The quantitative analysis of the three indices; NDWI, NDVI and NDVI_NDWI depicted their similar performance. In the section below, the qualitative assessment of all the experiments performed in the research is delineated. For uniformity in comparison all the outputs that were generated

on T44RQR only are depicted here. The outputs were visually compared with the respective RGB and ground truth maps of T44RQR tile in figure 6.5. Two different sites of the tile were chosen, one for the comparison of quality of water bodies extraction and another for the comparison of suppression of urban pixels during the extraction. One general observation found in the predicted maps was the broken pixels of water bodies. This was due to a considerably low spatial resolution (10m) and the reduction factor 8 that was necessarily applied while predicting. They effectively degraded the quality of visualization.

6.4.1 Performance of Neural Network Approach

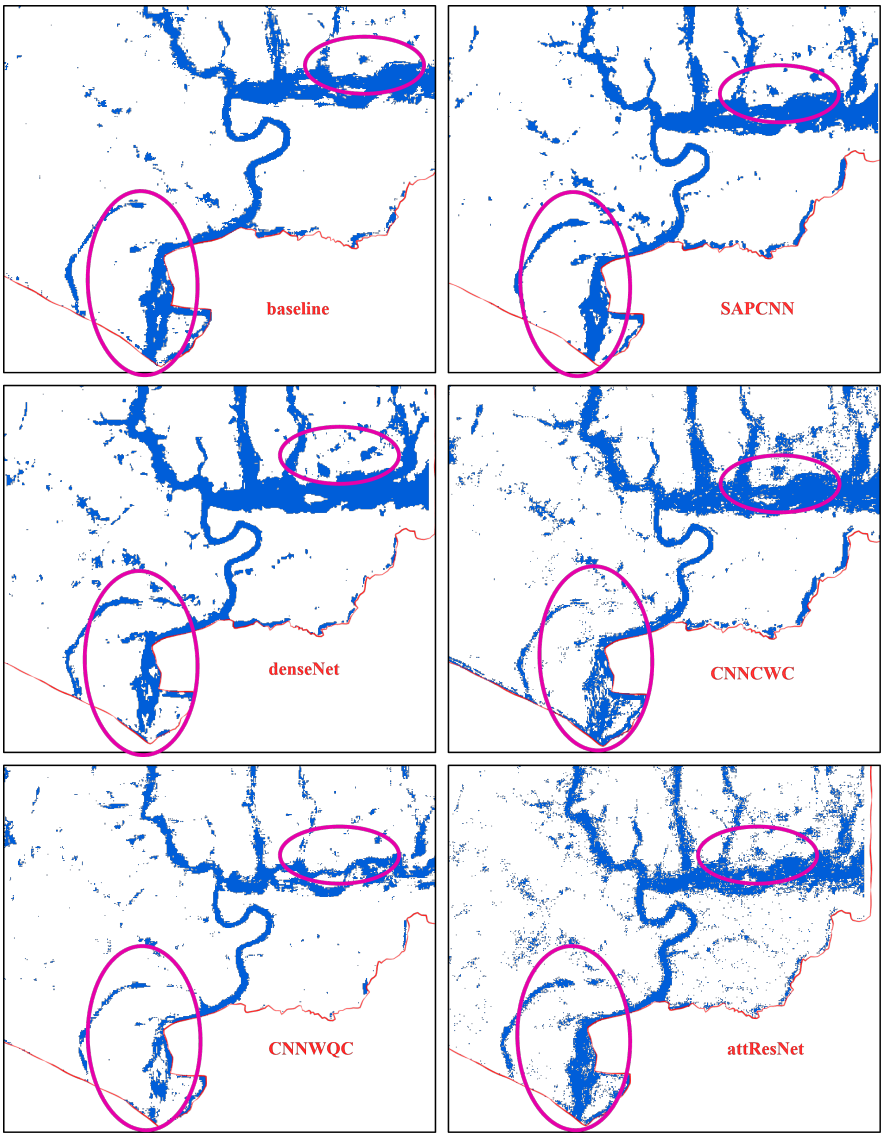


Figure 6.6: Performance of neural networks on extraction of water in selected regions (by ellipse in magenta)

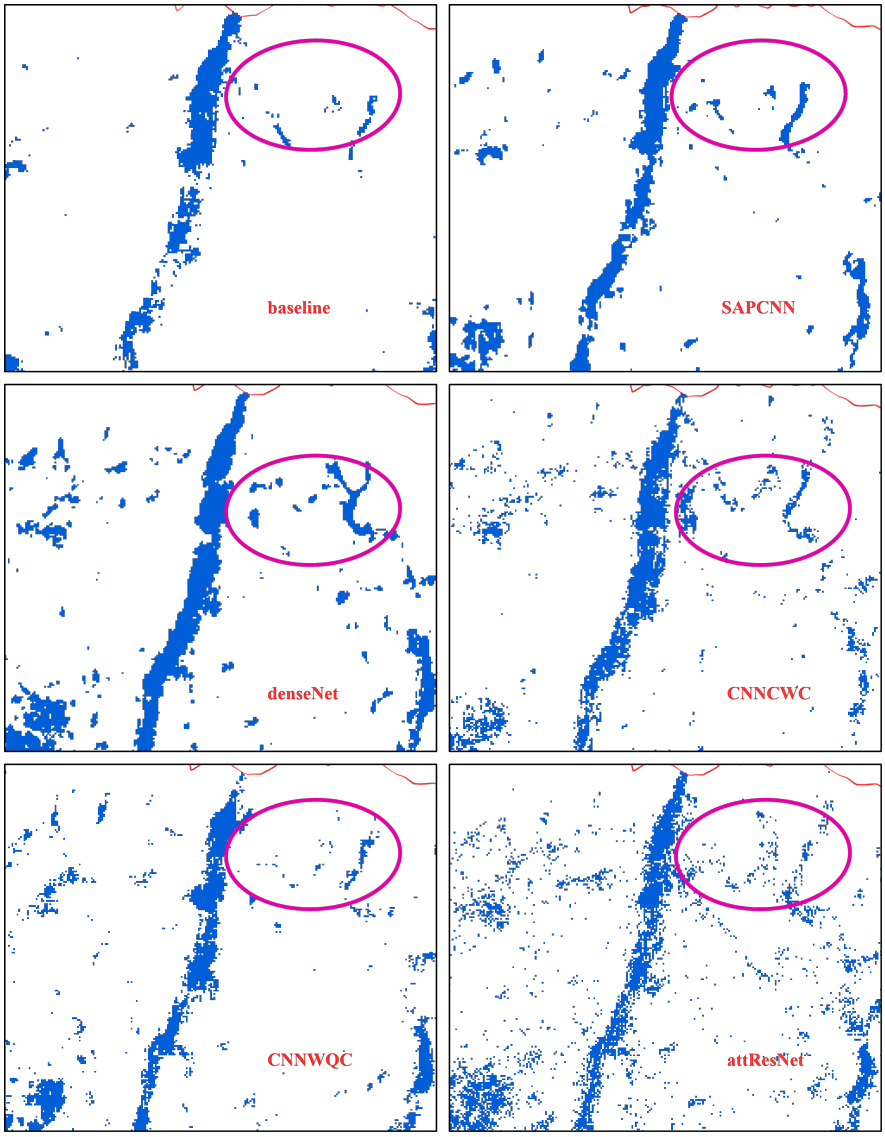


Figure 6.7: Quality of extracted water by the six networks in a selected region (by ellipse in magenta)

Figure 6.6 presents the outputs generated by the neural networks on the sample tile T44RQR. As case study two different regions; one inside the larger ellipse on the left-bottom and another inside the smaller ellipse on the top-right were considered. The ground truth data shown in figure 6.5a did not contain the entire region inside the larger ellipse (lower left). It shows that the models including the baseline, were efficient enough to predict water in places other than given by ground truth data also. On comparison to 6.8, the amount of water pixels extracted was more in figure 6.6. It extracted not only the pure water from water bodies, but also the river-banks, dried streams and small water bodies as well. It was found that DenseNet and CNNWQC better extracted true water pixels than the likes of CNNCWC and AttResNet. As seen on the smaller ellipse, SAPCNN also performed well in the sense that it

contained lesser false positives than AttResNet and CNNCWC. We can see a lot of non water pixels extracted as water in case of AttResNet and CNNCWC.

Figure 6.7 shows another case study of output tile T44RQR about the quality of extracted pixels by the six networks. The portion inside the left part of the ellipse were the pixels of urban features while that on the right were that of water bodies. It was found that unlike outputs from index-based approaches on fig 6.9, the networks better oppressed the urban pixels. Baseline model extracted a small linear streak of water pixels while SAPCNN even extracted better than the former. Just similar to the numerical performance of AttResNet at 87.67% being lower than that of CNNCWC at 88.01% in table 6.5, the visual performance also seconded the numerical performance. It was found that more non water pixels was extracted by the AttResNet than CNNCWC. In addition, CNNWQC was also found to be under-performing than SAPCNN here. The streak of water pixels on the right part of ellipse extracted by the later was more vivid than the former. This result was in contrast to the metrics in table 6.5 where the overall accuracy of CNNWQC being 89.34% was greater than that of SAPCNN being 89.19%. Indisputably, DenseNet was found to be performing the best among the rest in terms of oppressing the urban pixels. Moreover, it extracted the two tributaries of the river better than other networks inside the right part of the ellipse.

6.4.2 Performance of Traditional Approach

Figure 6.8 shows the visual extraction of water features by four traditional indices; NDWI, NDVI, NDVI_NDWI and EWI. It was found that all the four indices extracted water features from the pixels not in the ground truth map also. EWI was found to be extracting pure water pixels only. Though the numerical performance of NDWI, NDVI and NDVI_NDWI were similar, the visual performance depicted that NDVI_NDWI slightly edged the other indices in extracting small water bodies. Better extraction of water bodies can be seen in the smaller ellipse of NDVI_NDWI. EWI under-performed in that region because it was found to be extracting only pure water pixels. It left out the riverbanks and dried rivers. Its only positive was that it prominently suppressed the extraction of urban pixels. NDVI, NDWI and NDVI_NDWI all three indices extracted barren pixels also as seen in the larger ellipse. EWI extracted lesser water pixels than the rest in this regard. It can be inferred that to extract pure water pixels only, EWI can be used but at the cost of riverbanks and dried rivers.

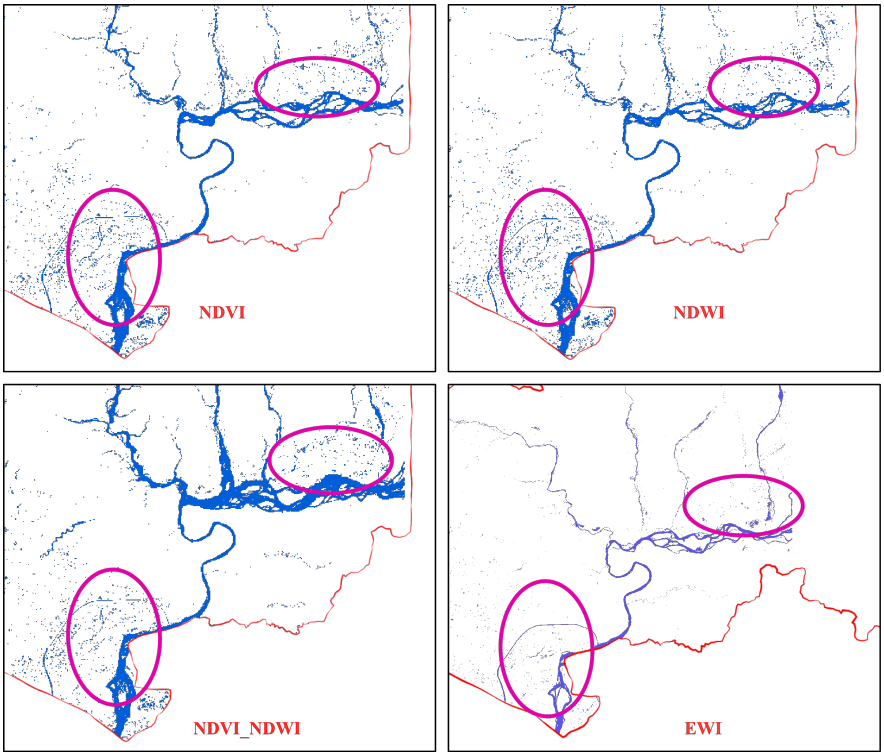


Figure 6.8: Performance of Water indices in a selected region (by ellipse in magenta)

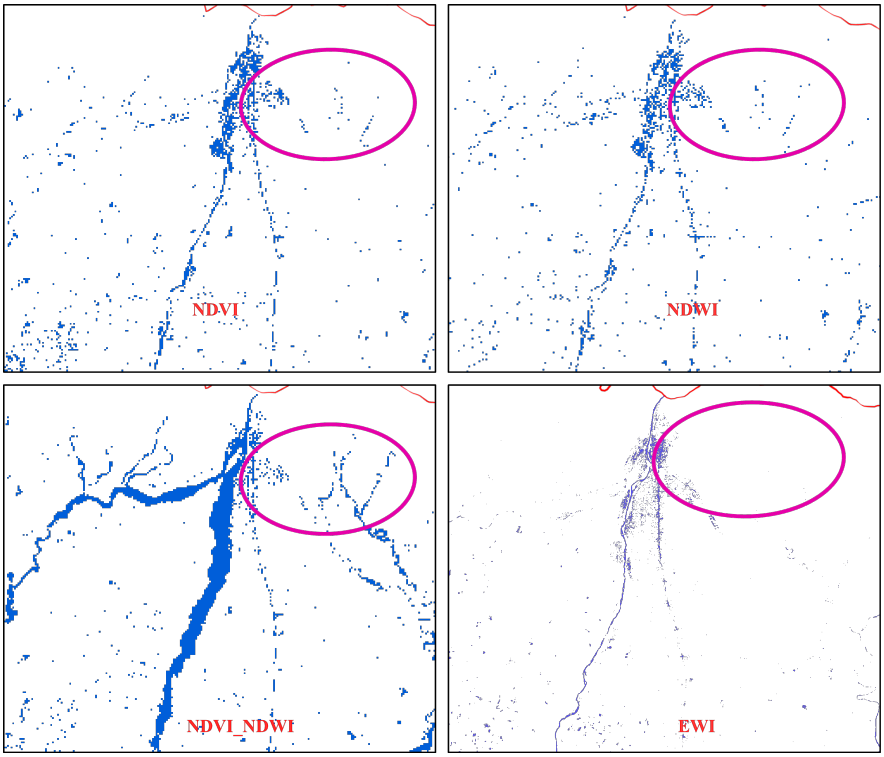


Figure 6.9: Quality of extracted water by four indices in a selected region (by ellipse in magenta)

Figure 6.9 presents the quality of water bodies extracted by the four indices. The indices were supposed to suppress the urban pixels on the left part of the

ellipse and extract the two tributaries of a small river on the right. It was found that all the indices were confused with the urban pixels and extracted a linear of road pixels as water features. Compared to NDVI and NDWI, the contrast between the water and urban was higher in NDVI_NDWI. The urban pixels could be eliminated by updating the threshold values but this would affect the extraction of the two tributaries of water on the right part of the ellipse. The tributaries were extracted best by NDVI_NDWI in comparison to the other indices. EWI did not perform well in the sense that there was less contrast between the water and urban pixels. It also eliminated the two tributaries on the right which made it worse than the rest indices. Hence in overall the performance of the indices could be ranked as NDVI_NDWI, NDVI, NDWI and EWI on the descending order.

All in all, it was found that the neural networks performed better than the traditional index-based approaches. The test accuracy of extraction using NDVI, for instance at 16 patch size increased from 75% to 89.73% on using DenseNet. Visually also, index-based approaches extracted only pure water pixels while excluding the riverbanks, small water bodies and dried streams. This issue was addressed by the neural networks in the figures between 6.8 and 6.6. It was also found from the figures 6.9 and 6.7 that the neural networks oppressed the urban pixels better than the index-based approaches.

6.4.3 Performance of the proposed network

The proposed network reached the test accuracy of 90.29% with recall values of water and non water as 86 and 93 respectively (see table 6.6). The visual performance of the proposed network was found to be improvised than DenseNet. Theoretically, the attention component of the network was supposed to focuse more on water features and suppressed non water features. Compared to the performance of DenseNet in figure 6.6, the figure 6.10a produced more vivid water pixels (see the left ellipse). It was also able to extract the smaller streams (see right ellipse) and oppress the non water features elsewhere in the study area. The non water pixels that were prevalent in the output from AttResNet was completely excluded by the proposed network. For further validation of these facts, the performance of the network was assessed in another tile T44RNS also. Our network was found to be correcting the mistakes in ground truth map also. The flow path within the region of the ellipse in ground truth map in figure 6.10b might be the path of river previously. But the actual flow of the river could be observed in the RGB image of corresponding figure. The

proposed network seconded the path of river in RGB image irrespective of the ground truth map.

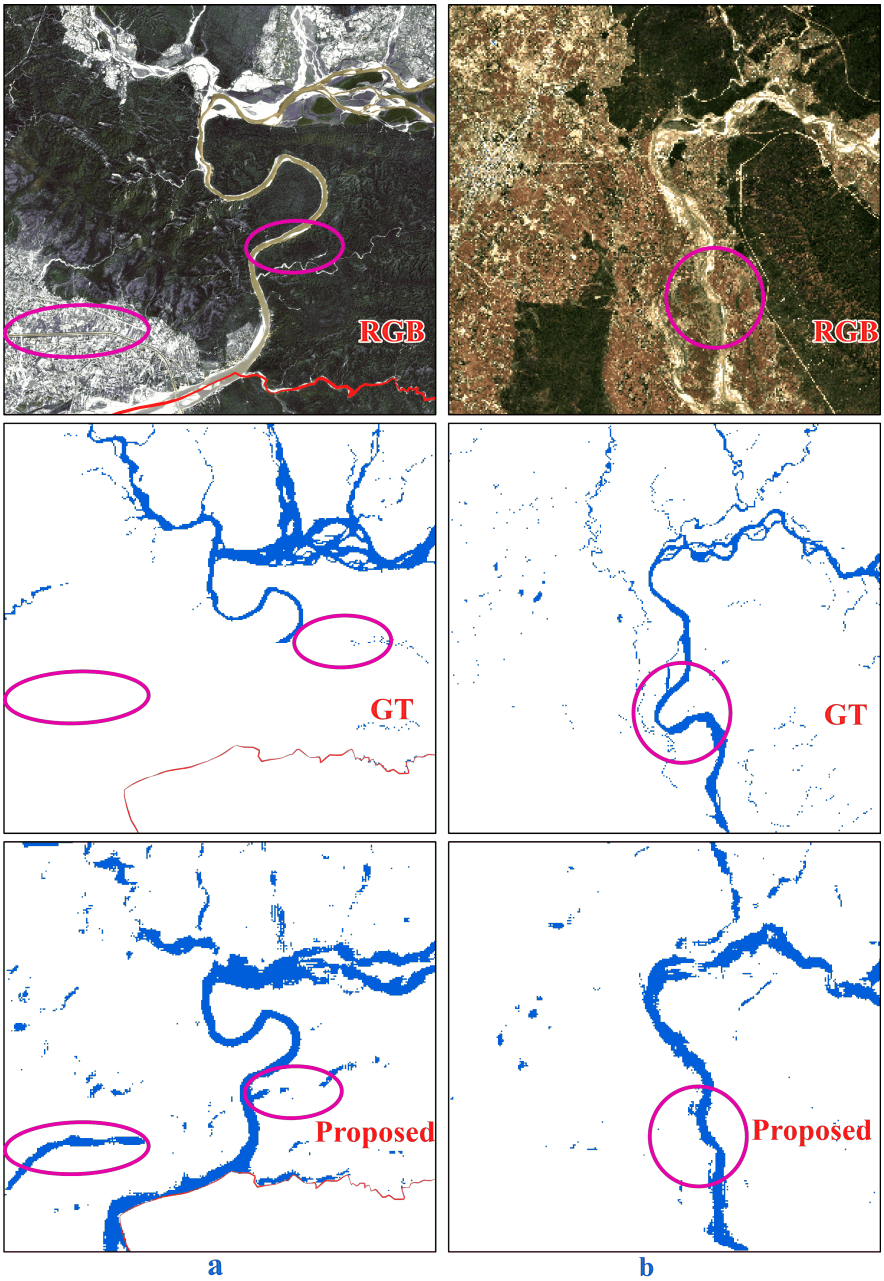


Figure 6.10: Performance of the proposed network in a. T44RQR and b. T44RNS tiles as highlighted in selected regions (by ellipses in magenta).

Hence the proposed network was found to be performing better than all the six experimented networks since its numerical performance was the best among all. In addition, its visual performance was assessed both in T44RQR and T44RNS tiles. There also it was found to be performing better than the rest and even the ground truth data as well.

Chapter 7

Limitations and Recommendations

This chapter explains about the limitations and recommendations of the research. Section 7.1 talks about the limiting factors that created obstacle in producing better outputs for water features extraction. Section 7.2 talks about suggestions on how the research works would have been improved to get better results. It also talks about the direction on how the research can be extended for future works.

7.1 Limitations of the research

The major limiting factor for the research was the hardware components and resources. The hardware comprised of 111GB of RAM and 11GB of GPU which proved to be insufficient for a detailed execution of models. Preliminary experiments were done on step 16 and 12 which ran smoothly. But as soon as step was reduced in quest for higher accuracy, the computational cost increased so much that it was not possible to go lesser than 8 to execute all the models. The ideal case would have been 1 but that was proved to be humongous ask considering the capacity of the server.

Similarly during EWI computation, the hardware was not able to compute the covariance of matrices to extract principal components from the large number of stacked patches due to memory issues. This made us shift to the regular approach of computing the components of the whole tile using ArcGIS Pro 2.7. This limited the assessment of the performance from EWI to qualitative approach only.

The spatial resolution of the tiles used was 10m. So with patch size of 8, it effectively covered 80m on the ground which could not properly represent the water bodies smaller than 80m. This prompted for the broken pixels of water in the predicted maps.

Finally the last limitation was the reduction of the image size before applying it to the trained model for prediction. As the resolution of the largest tile was 10980*10980, it meant that this number of pixels had to be predicted which was something impossible considering the computational cost. Hence the tiles were reduced by a factor of 8 before making prediction.

7.2 Recommendations for future tasks

The major suggestion for future works from the researchers would be to deal with AttResNet individually than with other networks. AttResNet was found to be the slowest network requiring high computational cost. It needed its own preferable set of hyperparameters. The learning rate should be kept smaller than 0.001, the batch size less than 64 and number of epochs less than 100 would be preferable. If the ground truth data does not support complex networks, then the number of attention and residual blocks can be reduced as needed.

Similarly it was not much beneficial to run different models with same configuration unless the objective is just an on-the-fly comparison. The SAPCNN theoretically opted for converting the original image pixels into superpixels and only then CNN can be used on them. It was experienced that even without this conversion, SAPCNN third rank among the six. Had it been used with the suggested approach of converting to superpixels and using self-adapting pooling as proposed by [5], the results would have been even better. Same applies for the remaining models.

Another recommendation would be regarding the quality of ground truth data. The images were from 2020 and the label data were from 2015 which impacted to the final result. Hence it is suggested to take the updated data which is properly geo-referenced and complete.

Besides, it was found that index-based approaches could extract pure water pixels only. Since the ground truth data also contained the pixels for riverbanks, dried streams and small water bodies as well, comparing the output with this ground truth is not logical. Hence ground truth containing pure water pixels only can be used for the quantitative assessment purpose.

Chapter 8

Conclusion

This thesis depicts the extraction of water features from Sentinel-2 imagery using convolutional neural networks. The tasks were oriented towards the exploration of state-of-art approaches for water feature extraction, to implement those approaches and finally integrate them to design a new approach with better performance than the existing ones. The research was conducted on the 18 Terai districts on the Southern plain of Nepal covered with 11 Sentinel-2 tile scenes. The results demonstrated that novel architects like DenseNet and AttResNet can be integrated to extract water features from satellite images. The detailed conclusion of this thesis are presented in conjunction with the objectives set before starting the research works as enumerated below:

1. **To explore the state-or-art approaches used for extraction of water features from satellite images**

   Chapter 2 presented the literature on five different neural networks and four different index-based methods that were implemented in this research. The neural networks under consideration were SAPCNN, CNNWQC, CNNCWC, DenseNet and AttResNet. It talked about their brief introduction, evolution, study area, datasets, methodology used for implementation and results obtained from those networks. In addition, it also depicted the way they were implemented originally and the study area where they had reached their best performance. Besides, it also discussed briefly about other architects like MRDED, SSAE, OTOP and DeepWaterMap with their purpose. In the index-based methods, it talked about

NDWI, NDVI, MNDWI, NWI, GWI and development and EWI. It talked about their pros, cons and how principal components can be integrated with indices to propose a new algorithm like EWI. In order to make the readers acquainted with the high-tech terms used in CNN, chapter 3 gave a holistic approach on the relationship of artificial intelligence, machine learning, deep learning and data science. The chapter then discussed thoroughly about the CNN, its architecture and other terms used in CNN.

2. **To implement the state-of-art approaches in the study area data and compare the performance achieved**

   All the six neural networks and 4 index-based methods were implemented as discussed in chapter 5, the results of which are delineated in chapter 6. The NDWI, NDVI and NDVI_NDWI were implemented according to their definition in section 2.1 and EWI was implemented according to the method proposed by [25]. To implement the neural networks, the dataset was splitted into training and test set and training set was again splitted into training and validation set. Initial experiments were conducted several times to determine the common hyperparameters for all models. Finally, the parameters were set to patch sizes of 8,12,16 and 20 with step size 8 on batch size 128 for 100 epochs using the learning rate of 0.001. All the architects were improvised with uniform configuration.

   Four different experiments with different number of input channels were conducted to make a comparison chart of the six networks. The best performance when feeding only RGB channels at patch size 16 was given by DenseNet with test accuracy of 85.41% and recall values of 75 for water and 89 for non water (see table 6.2). On using 11 selected channels as input the test accuracy for patch size 16 increased to 89.53% and recall values to 84 for water and 92 for non water (see table 6.3). Another experiment to assess the contribution of DEM was conducted by feeding RGB and DEM as fourth channel on baseline model. The test accuracy was found to be increased by 2.26% on patch size 8 for baseline model though the rate decreased on increase with patch size (see table 6.4). Anyways it was decided to consider DEM as another channel for input since it was contributing to the water feature extraction to some extent though not significantly. The final experiment was conducted with 11 selected channels from Sentinel-2 integrated with DEM as 12^{th} channel. This experiment

produced the best accuracy of 90.41% on patch size 20 (see table 6.5) but considering the training time and compatibility for AttResNet it was decided to use patch size 16 for further research. Later the structure of proposed model was adjusted to make it compatible for all patch sizes.

Additionally, the four indices were also employed for extracting water features to compare with the performance of neural networks. It was found that numerical performance of NDWI, NDVI and NDVI_NDWI for all patches were similar around 75% of test accuracy and 52 of recall for water and 97 for that of non water. EWI was assessed only visually due to memory issues for computing principal components in terms of stacked patches.

Qualitative assessment of the outputs also proved that DenseNet was performing the best with more vivid outputs than the rest. The models could be ranked in the order of decreasing performance as DenseNet, CNNWQC, SAPCNN, CNNCWC, baseline and AttResNet. The performance of the models were also found better than that from index-based method in terms of extracting water features and suppressing the non water pixels.

3. **To innovate and design a CNN architecture with highest qualitative and quantitative performance using the available architectures**

   An attention block was added after the batch normalization layer of DenseNet and again experiments were performed to determine the hyper-parameters for the integrated network. It was found that the proposed network gave the best result of 90.29% of test accuracy, 86 of recall for water and 93 of recall for non water. The final configuration to achieve this result was 3 dense blocks, 1 attention depth, 64 batch size and 0.01 learning rate with other parameters unchanged (see table 6.6).

   Finally, from the visual comparison also the proposed network was found to be performing better than the DenseNet with focused attention to water features only (see figure 6.10). It was worth noting that it also removed the non water pixels from the boundary of the study area which was present in the six networks. The urban and barren pixels found in the output of AttResNet of figure 6.6 were also not carried by the integrated network though AttResNet was not performing up to the mark in the experiments. Moreover, the proposed network was found to be successful in extracting smaller water bodies like streams, ponds and

lakes as well throughout the scene. The novelty of this network lied in its composition. It integrated different blocks like dense, transition, attention and residual in one architect to perform better than the other state-of-the-art approaches.

In summary, we became successful in achieving the main aim of extracting surface water features from Sentinel-2 imagery using CNN. We implemented state-of-art technologies for water feature extraction and importantly designed a new network integrating DenseNet and AttResNet. We also validated our works vis-a-vis the traditional index-based approaches as well. We envisage that CNN techniques will improvise more and more to efficiently extract features from satellite imageries in future.

The self-assessment of Thesis Reproducibility in accordance with OSF Home Guidelines for Input Data, Preprocessing, Processing, Computational Environments and Results is: 2, 2, 1, 1 and 2 respectively.

Bibliography

[1] Y. Wang, Z. Li, C. Zeng, G. Xia, and H. Shen. "Extracting urban water by combining deep learning and Google Earth Engine." In: (), pp. 1–14.

[2] K. Mishra and P. R. C. Prasad. "Automatic Extraction of Water Bodies from Landsat Imagery Using Perceptron Model." In: *J. Comput. Environ. Sci.* 2015 (2015), pp. 1–9. ISSN: 2356-7279. DOI: 10.1155/2015/903465.

[3] S. Shrestha and L. Vanneschi. "Improved Fully Convolutional Network with conditional field for building extraction." University of Nova, Lisbon, 2018. DOI: 10.3390/rs10071135. URL: https://novaresearch.unl.pt/en/publications/improved-fully-convolutional-network-with-conditional-random-fiel.

[4] F. Ye, L. Yang, S. Tian, L. Yu, J. Qian, and Y. Qian. "Deep Learning for extracting water body from Landsat Imagery." In: *Int. J. Innov. Comput. Inf. Control* 11.6 (2015), pp. 1913–1929.

[5] Y. Chen, R. Fan, X. Yang, J. Wang, and A. Latif. "Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning." In: *Water (Switzerland)* 10 (5 May 2018). ISSN: 20734441. DOI: 10.3390/w10050585.

[6] R. John B. "Features shown on topographic maps." In: *Geol. Surv. Cricular 368* (1955), pp. 1–29.

[7] M. Alkan, U. G. Sefercik, A. M. Marangoz, and S. Karakış. "Updating Objects for Topographic Map Information Using High Resolution Optical Images of Zonguldak Test Field." In: *30th EARSel Sempozyumu* April 2015 (2010).

[8] *Landsat Program*. URL: http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson3-2/landsat.html# (visited on 09/02/2020).

[9] J. Lynn. *Landsat Overview | NASA*. June 2020. URL: https://www.nasa.gov/mission_pages/landsat/overview/index.html (visited on 09/02/2020).

[10] *What are the band designations for the Landsat satellites?* URL: https://www.usgs.gov/faqs/what-are-band-designations-landsat-satellites?qt-news_science_products=0#qt-news_science_products (visited on 09/02/2020).

[11] *Sentinel-1 - Mission Objectives - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/mission-objectives (visited on 10/13/2020).

[12] *Sentinel 2 Infographics*. URL: https://sentinels.copernicus.eu/documents/247904/4180891/Sentinel-2-infographic.pdf (visited on 10/13/2020).

[13] *Sentinel-2 - Missions - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2 (visited on 10/13/2020).

[14] *Sentinel-3 - Missions - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-3 (visited on 10/13/2020).

[15] *Sentinel-4 - Missions - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-4 (visited on 10/13/2020).

[16] *Sentinel-5 - Missions - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-5 (visited on 10/13/2020).

[17] *Sentinel-5P - Missions - Sentinel Online - Sentinel*. URL: https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-5p (visited on 10/13/2020).

[18] *MODIS Web*. URL: https://modis.gsfc.nasa.gov/about/ (visited on 10/13/2020).

[19] *MODIS Specifications*. URL: https://modis.gsfc.nasa.gov/about/specifications.php (visited on 10/13/2020).

[20] *RapidEye Satellite Sensor | Satellite Imaging Corp*. URL: https://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/rapideye/ (visited on 10/13/2020).

[21] *RapidEye - eoPortal Directory - Satellite Missions*. URL: https://earth.esa.int/web/eoportal/satellite-missions/r/rapideye (visited on 10/13/2020).

[22] *50 Satellites in Space: Types and Uses of Satellites - GIS Geography*. URL: https://gisgeography.com/earth-satellite-list/ (visited on 10/13/2020).

[23] P. Zhang, L. Chen, Z. Li, J. Xing, X. Xing, and Z. Yuan. "Automatic extraction of water and shadow from SAR images based on a multi-resolution dense encoder and decoder network." In: *Sensors (Switzerland)* 19.16 (2019). ISSN: 14248220. DOI: 10.3390/s19163576.

[24] G. Wang, M. Wu, X. Wei, and H. Song. "Water Identification from High-Resolution Remote Sensing Images Based on Multidimensional Densely Connected Convolutional Neural Networks." In: *Remote Sensing* 12.5 (2020), p. 795. ISSN: 2072-4292. DOI: 10.3390/rs12050795. URL: https://www.mdpi.com/2072-4292/12/5/795.

[25] J. Yang and X. Du. "An enhanced water index in extracting water bodies from Landsat TM imagery." In: *Annals of GIS* 23.3 (2017), pp. 141–148. ISSN: 19475691. DOI: 10.1080/19475683.2017.1340339. URL: https://doi.org/10.1080/19475683.2017.1340339.

[26] F. Isikdogan, A. C. Bovik, and P. Passalacqua. "Surface water mapping by deep learning." In: *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 10.11 (2017), pp. 4909–4918. ISSN: 21511535. DOI: 10.1109/JSTARS.2017.2735443.

[27] H. Guo, G. He, W. Jiang, R. Yin, L. Yan, and W. Leng. "A multi-scale water extraction convolutional neural network (MWEN) method for GaoFen-1 remote sensing images." In: *ISPRS Int. J. Geo-Information* 9.4 (2020). ISSN: 22209964. DOI: 10.3390/ijgi9040189.

[28] A. Veit, M. Wilber, and S. Belongie. "Residual Networks Behave Like Ensembles of Relatively Shallow Networks." In: 2016.

[29] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. "Enhanced Deep Residual Networks for Single Image Super-Resolution." In: 2017, pp. 136–144.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely Connected Convolutional Networks." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2017-January. IEEE, 2017, pp. 2261–2269. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.243. arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993https://ieeexplore.ieee.org/document/8099726/.

[31] J. Chen, C. Wang, K. Wang, and M. Liu. "Computational efficient deep neural network with differential attention maps for facial action unit detection." In: (2020). arXiv: 2011.12082. URL: http://arxiv.org/abs/2011.12082.

[32] Y. Mei, Y. Fan, Y. Zhang, J. Yu, Y. Zhou, D. Liu, Y. Fu, T. S. Huang, and H. Shi. "Pyramid attention networks for image restoration." In: *arXiv* (2020). arXiv: 2004.13824. URL: https://arxiv.org/pdf/2004.13824.pdf.

[33] G. Huang, Z. Liu, G. Pleiss, L. van der Maaten, and K. Q. Weinberger. "Convolutional Networks with Dense Connectivity." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2918284. arXiv: 2001.02394. URL: https://ieeexplore.ieee.org/document/8721151/http://arxiv.org/abs/2001.02394.

[34] F. Pu, C. Ding, Z. Chao, Y. Yu, and X. Xu. "Water-quality classification of inland lakes using landsat8 images by convolutional neural networks." In: *Remote Sensing* 11.14 (2019), p. 1674.

[35] M. Rezaee, M. Mahdianpari, Y. Zhang, and B. Salehi. "Deep convolutional neural network for complex wetland classification using optical remote sensing imagery." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.9 (2018), pp. 3030–3039.

[36] N. Wagle and T. D. Acharya. "Past and Present Practices of Topographic Base Map Database Update in Nepal." In: *ISPRS International Journal of Geo-Information* 9.6 (2020), p. 397. ISSN: 2220-9964. DOI: 10.3390/ijgi9060397. URL: https://www.mdpi.com/2220-9964/9/6/397.

[37] T. D. Acharya, A. Subedi, and D. H. Lee. "Evaluation of water indices for surface water extraction in a landsat 8 scene of Nepal." In: *Sensors (Switzerland)* 18.8 (2018), pp. 1–15. ISSN: 14248220. DOI: 10.3390/s18082580.

[38] T. D. Acharya, A. Subedi, H. Huang, and D. H. Lee. "Application of water indices in surface water change detection using Landsat imagery in Nepal." In: *Sensors Mater.* 31.5 (2019), pp. 1429–1447. ISSN: 09144935. DOI: 10.18494/SAM.2019.2264.

[39] Y. Li, X. Gong, Z. Guo, K. Xu, D. Hu, and H. Zhou. "An index and approach for water extraction using Landsat–OLI data." In: *Int. J. Remote Sens.* 37.16 (2016), pp. 3611–3635. ISSN: 13665901. DOI: 10.1080/01431161.2016.1201228.

[40] L. Ji, L. Zhang, and B. Wylie. "Analysis of dynamic thresholds for the normalized difference water index." In: *Photogrammetric Engineering and Remote Sensing* 75.11 (2009), pp. 1307–1317. ISSN: 00991112. DOI: 10.14358/PERS.75.11.1307.

[41] H. Xu. "Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery." In: *International Journal of Remote Sensing* 27.14 (2006), pp. 3025–3033. ISSN: 13665901. DOI: 10.1080/01431160600589179.

[42] H Gao, L Wang, L Jing, and J Xu. "An effective modified water extraction method for Landsat-8 OLI imagery of mountainous plateau regions." In: *IOP Conference Series: Earth and Environmental Science* 34 (2016), p. 012010. DOI: 10.1088/1755-1315/34/1/012010. URL: https://doi.org/10.1088%2F1755-1315%2F34%2F1%2F012010.

[43] D. Feng. "A New Method for Fast Information Extraction of Water Bodies Using Remotely Sensed Data." In: *Remote Sensing Technology and Application* 24 (2 Feb. 2009), pp. 167–171. ISSN: 1004-0323. DOI: 10.11873/J.ISSN.1004-0323.2009.2.167. URL: http://www.rsta.ac.cn/CN/abstract/abstract1638.shtml.

[44] C. Yang and M. Xu. "Study on the water-body extraction methods of remote sensing information mechanism." In: *Geographical Research* 17 (Jan. 1998), pp. 86–89.

[45] K. Rokni, A. Ahmad, A. Selamat, and S. Hazini. "Water Feature Extraction and Change Detection Using Multitemporal Landsat Imagery." In: *Remote Sensing* 6.5 (2014), pp. 4173–4189. ISSN: 2072-4292. DOI: 10.3390/rs6054173. URL: http://www.mdpi.com/2072-4292/6/5/4173.

[46] J. Yang and W. Li. "Feature selection methods in extracting impervious surface from Landsat TM image." In: *Annals of GIS* 19 (4 2013), pp. 253–261. ISSN: 1947-5691. DOI: 10.1080/19475683.2013.843591. URL: https://www.tandfonline.com/action/journalInformation?journalCode=tagi20.

[47] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. *SLIC superpixels**. Ecole Polytechnique F edrale de Lausanne (EPFL), 2010. URL: https://www.researchgate.net/publication/44234783.

[48] J. Cong, B. Wei, Y. Yin, X. Xi, and Y. Zheng. "Performance evaluation of simple linear iterative clustering algorithm on medical image processing." In: vol. 24. IOS Press, 2014, pp. 3231–3238. DOI: 10.3233/BME-141145.

[49] L. Yu, Z. Wang, S. Tian, F. Ye, J. Ding, and J. Kong. "Convolutional Neural Networks for Water Body Extraction from Landsat Imagery." In: *International Journal of Computational Intelligence and Applications* 16 (1 Mar. 2017). ISSN: 14690268. DOI: 10.1142/S1469026817500018.

[50] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. "Residual Attention Network for Image Classification." In: (Apr. 2017). URL: http://arxiv.org/abs/1704.06904.

[51] G. Singh, A. Mishra, and D. Sagar. "AN OVERVIEW OF ARTIFICIAL INTELLIGENCE." In: *SBIT J. Sci. Technol. ISSN 2277-8764* 2.1 (2013), pp. 3–6.

[52] A. Habeeb. "Artificial intelligence Ahmed Habeeb University of Mansoura." In: *Res. Gate* 7.2 (2017). DOI: 10.13140/RG.2.2.25350.88645.

[53] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. López García, I. Heredia, P. Malík, and L. Hluchý. "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey." In: *Artif. Intell. Rev.* 52.1 (2019), pp. 77–124. ISSN: 15737462. DOI: 10.1007/s10462-018-09679-z. URL: https://doi.org/10.1007/s10462-018-09679-z.

[54] N. Kühl, M. Goutier, R. Hirt, and G. Satzger. "Machine Learning in Artificial Intelligence: Towards a Common Understanding." In: *Proc. 52nd Hawaii Int. Conf. Syst. Sci.* September 2018 (2019). DOI: 10.24251/hicss.2019.630. arXiv: 2004.04686.

[55] *Introduction to Neural Networks: DNN/CNN/RNN*. URL: http://alinlab.kaist.ac.kr/resource/Lec1_Introduction_to_NN.pdf (visited on 09/13/2020).

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Adv. Neural Inf. Process. Syst.* 25.2 (2012), pp. 1097–1105. DOI: 10.1145/3065386.

[57] *CS231n Convolutional Neural Networks for Visual Recognition*. URL: https://cs231n.github.io/convolutional-networks/ (visited on 09/13/2020).

[58] D. Cielen, A. D. B. Meysman, and M. Ali. *Introducing Data Science*. Ed. by D. Maharry. Manning Publications Co., 2016, pp. 1–322.

[59] J. Grus. *Data Science from Scratch*. Ed. by M. Beaugureau. 1st ed. O'Reilly Media, Inc., 2015, pp. 1–464. ISBN: 9781491901427. URL: http://math.ecnu.edu.cn/~lfzhou/seminar/[Joel_Grus]_Data_Science_from_Scratch_First_Princ.pdf.

[60] D. Conway. *The Data Science Venn Diagram*. Sept. 2010. URL: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram.

[61] M. Carranza-García, J. García-Gutiérrez, and J. Riquelme. "A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks." In: *Remote Sensing* 11.3 (2019), p. 274. ISSN: 2072-4292. DOI: 10.3390/rs11030274. URL: http://www.mdpi.com/2072-4292/11/3/274.

[62] S. Albelwi and A. Mahmood. "A Framework for Designing the Architectures of Deep Convolutional Neural Networks." In: *Entropy* 19.6 (2017), p. 242. ISSN: 1099-4300. DOI: 10.3390/e19060242. URL: http://www.mdpi.com/1099-4300/19/6/242.

[63] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan. *Advances in Deep Learning*. Ed. by J. Kacprzyk. Vol. 57. Studies in Big Data. Singapore: Springer Nature Singapore Pte Ltd., 2020, pp. 1–159. ISBN: 978-981-13-6793-9. DOI: 10.1007/978-981-13-6794-6. URL: http://link.springer.com/10.1007/978-981-13-6794-6.

[64] T. Chen, R. Xu, Y. He, and X. Wang. "A gloss composition and context clustering based distributed word sense representation model." In: *Entropy* 17 (9 2015), pp. 6007–6024. ISSN: 10994300. DOI: 10.3390/e17096007.

[65] *Introduction to Convolutional Neural Networks*. URL: https://web.stanford.edu/class/cs231a/lectures/intro_cnn.pdf (visited on 09/16/2020).

[66] A. M. F. Agarap. *Deep Learning using Rectified Linear Units (ReLU)*. Feb. 2019. URL: https://github.com/AFAgarap/relu-classifier..

[67] K. Kakuda, T. Enomoto, and S. Miura. "Nonlinear activation functions in CNN based on fluid dynamics and its applications." In: *CMES - Computer Modeling in Engineering and Sciences* 118 (2019), pp. 1–14. ISSN: 15261492. DOI: 10.31614/cmes.2019.04676.

[68] N. Gajhede, O. Beck, and H. Purwins. "Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples." In: vol. 04-06-October-2016. Association for Computing Machinery, Oct. 2016, pp. 111–115. ISBN: 9781450348225. DOI: 10.1145/2986416.2986453.

[69] F. Schilling. "The Effect of Batch Normalization on Deep Convolutional Neural Networks." KTH Royal Institute of Technology, 2016.

[70] S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Cornell University, Mar. 2015.

[71] A. Ghasemian, H. Hosseinmardi, and A. Clauset. *Evaluating Overfit and Underfit in Models of Network Community Structure*. Apr. 2019. URL: https://arxiv.org/pdf/1802.10582.pdf.

[72] *Amazon Machine Learning Developer Guide*. 2021 Amazon Web Services, Inc., 2021.

[73] H. K. JABBAR and R. Z. KHAN. *(8) (PDF) METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)*. Computer Science, Communication Instrumentation Devices, Dec. 2014, pp. 163–172. URL: https://www.researchgate.net/publication/295198699_METHODS_TO_AVOID_OVER-FITTING_AND_UNDER-FITTING_IN_SUPERVISED_MACHINE_LEARNING_COMPARATIVE_STUDY.

[74] X. Ying. "An Overview of Overfitting and its Solutions." In: vol. 1168. Institute of Physics Publishing, Mar. 2019. DOI: 10.1088/1742-6596/1168/2/022022.

[75] S. L. Rhodin and E. Kvist. *A comparative study between MLP and CNN for noise reduction on images*. KTH Royal Institute of Technology, June 20191, pp. 1–46. URL: http://kth.diva-portal.org/smash/get/diva2:1352765/FULLTEXT01.pdf.

[76] P. Buhlmann and S. van de Geer. *Springer Series in Statistics*. Springer, 2011. URL: http://www.springer.com/series/692.

[77] T. Hinz, P. Barros, and S. Wermter. "The effects of regularization on learning facial expressions with convolutional neural networks." In: vol. 9887 LNCS. Springer Verlag, 2016, pp. 80–87. ISBN: 9783319447803. DOI: 10.1007/978-3-319-44781-0_10.

[78] K. Jayech. "Regularized Deep Convolutional Neural Networks for Feature Extraction and Classification." In: vol. 10635 LNCS. Springer Verlag, 2017, pp. 431–439. ISBN: 9783319700953. DOI: 10.1007/978-3-319-70096-0_45.

[79] T. Baidar, F. P. Bañón, R. F. Beltrán, and M. S. R. de Andrade Caetano. "RICE CROP CLASSIFICATION AND YIELD ESTIMATION USING MULTI-TEMPORAL SENTINEL-2 DATA: A CASE STUDY OF TERAI DISTRICTS OF NEPAL." Universitat Jaume I, Feb. 2020.

[80] C. Bircanoğlu. "A Comparison of Loss Functions in Deep Embedding." BAHCESEHIR UNIVERSITY, June 2017. URL: https://www.researchgate.net/publication/318588371.

[81] E. A. Badr, C Joun, and G. E. Nasr. "Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand." In: 2002, American Association for Artificial Intelligence, May 2002, pp. 381–384. URL: www.aaai.org.

[82] Q. Zhu, Z. He, T. Zhang, and W. Cui. "Improving classification performance of softmax loss function based on scalable batch-normalization." In: *Applied Sciences (Switzerland)* 10 (8 Apr. 2020). ISSN: 20763417. DOI: 10.3390/APP10082950.

[83] A. U. Ruby, P. Theerthagiri, I. J. Jacob, and Y. Vamsidhar. "Binary cross entropy with deep learning technique for image classification." In: *International Journal of Advanced Trends in Computer Science and Engineering* 9 (4 July 2020), pp. 5393–5397. ISSN: 22783091. DOI: 10.30534/ijatcse/2020/175942020.

[84] *Callbacks API*. URL: https://keras.io/api/callbacks/ (visited on 10/12/2020).

[85] *Difference Between a Batch and an Epoch in a Neural Network*. URL: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ (visited on 10/12/2020).

[86] *Optimizers*. URL: https://keras.io/api/optimizers/ (visited on 10/13/2020).

[87] *Adam*. URL: https://keras.io/api/optimizers/adam/ (visited on 10/13/2020).

[88] *SGD*. URL: https://keras.io/api/optimizers/sgd/ (visited on 10/13/2020).

[89] *RMSprop*. URL: https://keras.io/api/optimizers/rmsprop/ (visited on 10/13/2020).

[90] *LearningRateScheduler*. URL: https://keras.io/api/callbacks/learning_rate_scheduler/ (visited on 10/13/2020).

[91] *ReduceLROnPlateau*. URL: https://keras.io/api/callbacks/reduce_lr_on_plateau/ (visited on 10/13/2020).

[92] *Metrics*. URL: https://keras.io/api/metrics/ (visited on 10/13/2020).

[93] *Regression metrics*. URL: https://keras.io/api/metrics/regression_metrics/#meansquarederror-class (visited on 10/13/2020).

[94] *Regression metrics*. URL: https://keras.io/api/metrics/regression_metrics/#meanabsoluteerror-class (visited on 10/13/2020).

[95] *Image segmentation metrics*. URL: https://keras.io/api/metrics/segmentation_metrics/#meaniou-class (visited on 10/13/2020).

[96] *Classification metrics based on True/False positives negatives*. URL: https://keras.io/api/metrics/classification_metrics/ (visited on 10/13/2020).

[97] N. Gandhi, L. J. Armstrong, O. Petkar, and A. K. Tripathy. "Rice Crop Yield Prediction in India using Support Vector Machines." In: 2016 IEEE, July 2016. ISBN: 9781509020331.

[98] S. Thapa, L. Schipper, M. Shrestha, S. Nepal, and A. Taylor. *FINAL REPORT: NEPAL STUDY ECOSYSTEMS, DEVELOPMENT, AND CLIMATE ADAPTATION Improving the knowledge base for planning, policy and management Technical Team*. Stockholm Environment Institute, Mar. 2011.

[99] U. Raj Bhuju, P. R. Shakya, T. B. Basnet, and S. Shrestha. *Nepal Biodiversity Resource Book*. Ed. by J. M. Mendez. International Centre for Integrated Mountain Development (ICIMOD), Ministry of Environment, Science and Technology (MOEST), Government of Nepal (GoN), 2007, pp. 1–161. ISBN: 978 92 9115 033 5. URL: https://lib.icimod.org/record/7560.

[100] *The DHS Program*. URL: https://dhsprogram.com/pubs/pdf/FR78/01Chapter01.pdf (visited on 01/05/2021).

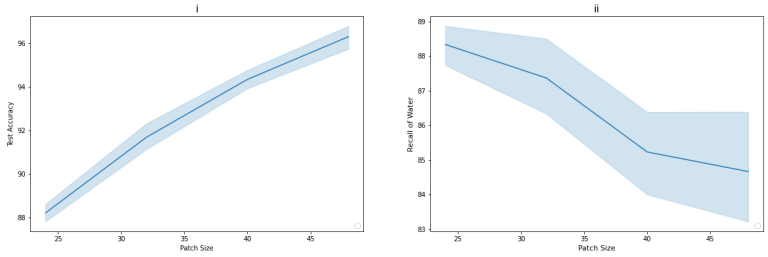
[101] M. Siwakoti and J. B. Karki. "Conservation status of Ramsar sites of Nepal Tarai: an overview." In: *Botanica Orientalis: Journal of Plant Science* 6 (Mar. 2010), pp. 76–84. ISSN: 1726-6858. DOI: 10.3126/botor.v6i0.2914.

[102] *Spatial - Resolutions - Sentinel-2 MSI - User Guides - Sentinel Online - Sentinel*. URL: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial (visited on 09/16/2020).

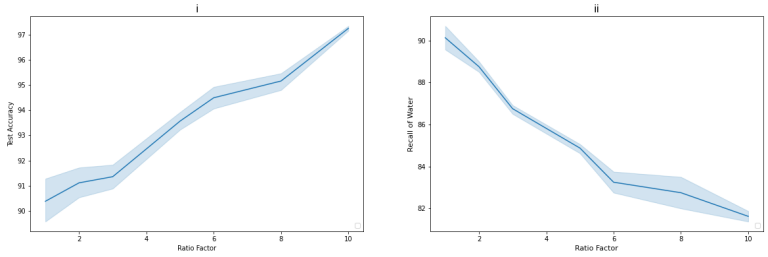
[103] J. Hamwood, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins. "Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers." In: *Biomedical Optics Express* 9.7 (2018), p. 3049. ISSN: 2156-7085. DOI: 10.1364/BOE.9.003049. URL: https://www.osapublishing.org/abstract.cfm?URI=boe-9-7-3049.

APPENDIX A

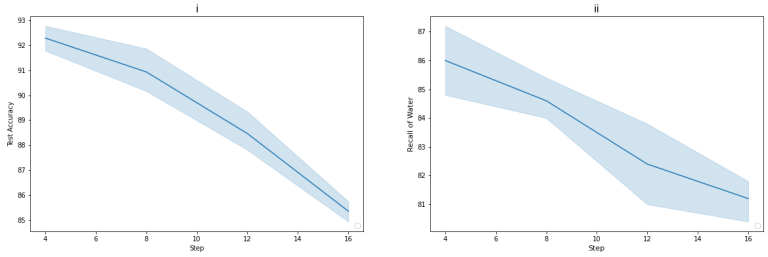
Analysis of Initial Experiments



(a) Patch size Vs i. Validation accuracy, ii. Recall Value of Water



(b) Non Water to Water Ratio Factor Vs i. Validation accuracy, ii. Recall Value of Water



(c) Step size Vs i. Validation accuracy, ii. Recall Value of Water

Figure A.1: Initial Experiments to determine the hyperparameters I

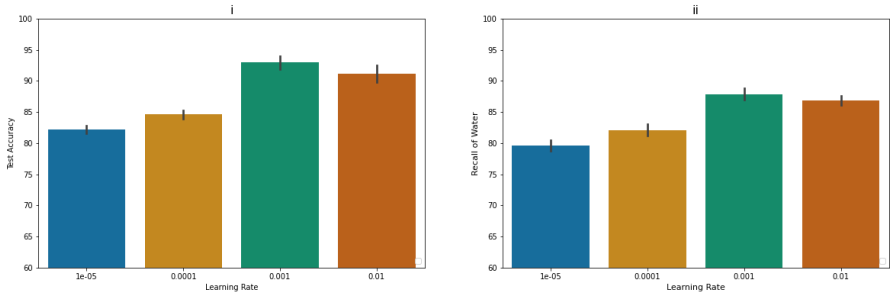


Figure A.2: Learning Rates Vs i. Validation Accuracy, ii. Recall Value of Water during initial experiments

APPENDIX B

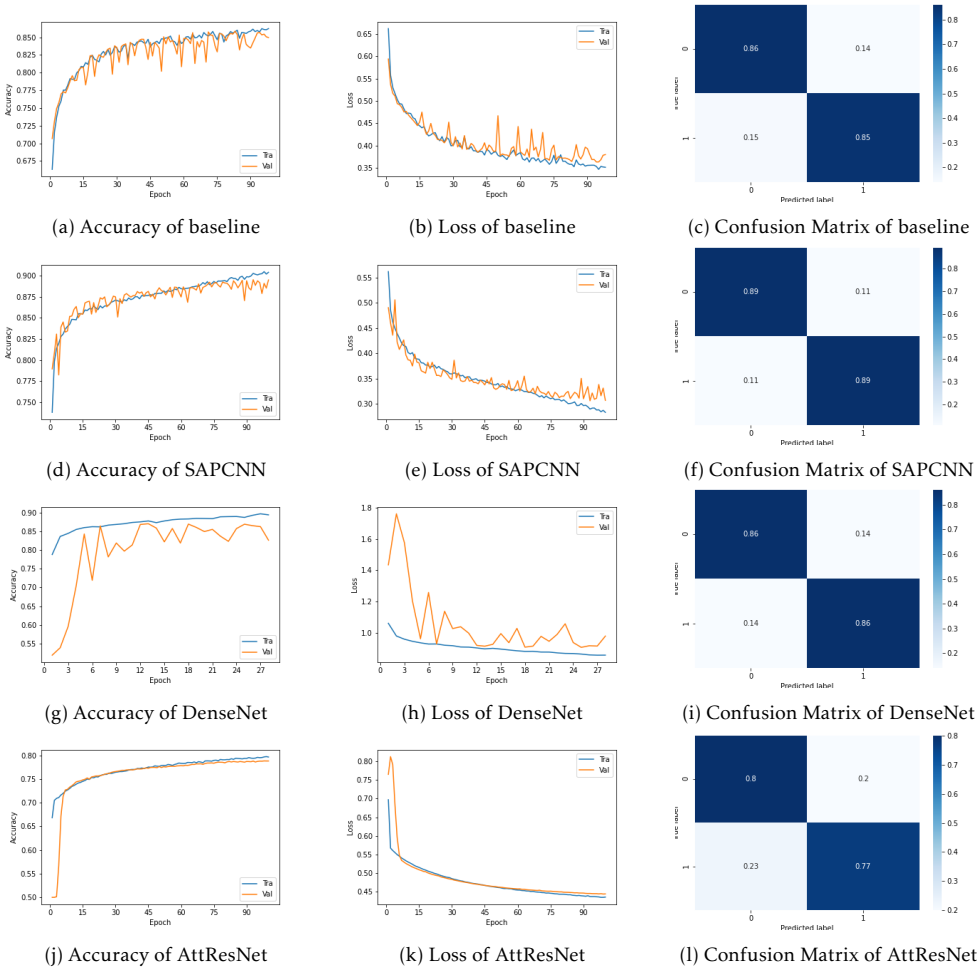
Precision and F1-Scores of Experiments

Table B.1: Precision and F1-Scores of each models obtained from four experiments

Channels	Models	Patch Size															
		8				12				16				20			
		F1-Score		Precision		F1-Score		Precision		F1-Score		Precision		F1-Score		Precision	
		Water	No Water	Water	No Water	Water	No Water	Water	No Water	Water	No Water	Water	No Water	Water	No Water	Water	No Water
RGB	baseline	69	86	75	83	73	88	79	85	75	88	79	86	76	89	79	87
	CNNWQC	72	87	74	85	76	89	79	87	77	89	78	88	77	89	79	87
	CNNCWC	No convergence				71	87	77	84	73	88	78	85	74	88	77	86
	SAPCNN	72	87	75	85	76	88	77	87	77	89	79	88	78	89	79	89
	denseNet	73	87	77	86	77	89	78	88	78	89	80	89	78	90	81	88
	attResNet	69	85	71	84	NA				73	87	75	86	NA			
RGB_DEM	baseline	72	88	80	84	75	89	80	86	78	89	80	88	78	90	81	88
S2	baseline	80	90	82	89	81	91	83	90	82	91	84	91	82	91	84	91
	CNNWQC	82	91	82	91	83	92	83	91	84	92	84	92	84	92	84	92
	CNNCWC	77	90	81	87	77	90	82	87	81	91	83	90	81	91	82	91
	SAPCNN	81	91	82	90	82	91	83	91	84	92	85	92	84	92	84	92
	denseNet	82	91	83	90	83	92	84	91	85	92	85	92	85	91	86	91
	attResNet	79	90	81	89	NA				81	90	81	90	NA			
S2_DEM	baseline	81	91	83	90	82	91	84	90	83	92	84	91	84	92	85	92
	CNNWQC	83	92	83	91	84	92	84	92	84	92	84	92	85	93	85	93
	CNNCWC	79	90	82	89	81	91	83	90	82	91	84	90	83	91	83	91
	SAPCNN	82	91	83	91	83	92	84	91	84	92	85	91	85	92	85	93
	denseNet	83	91	84	91	84	92	84	92	85	92	85	92	86	93	86	93
	attResNet	81	90	82	90	NA				81	91	83	90	NA			

ANNEX I

PERFORMANCE OF MODELS



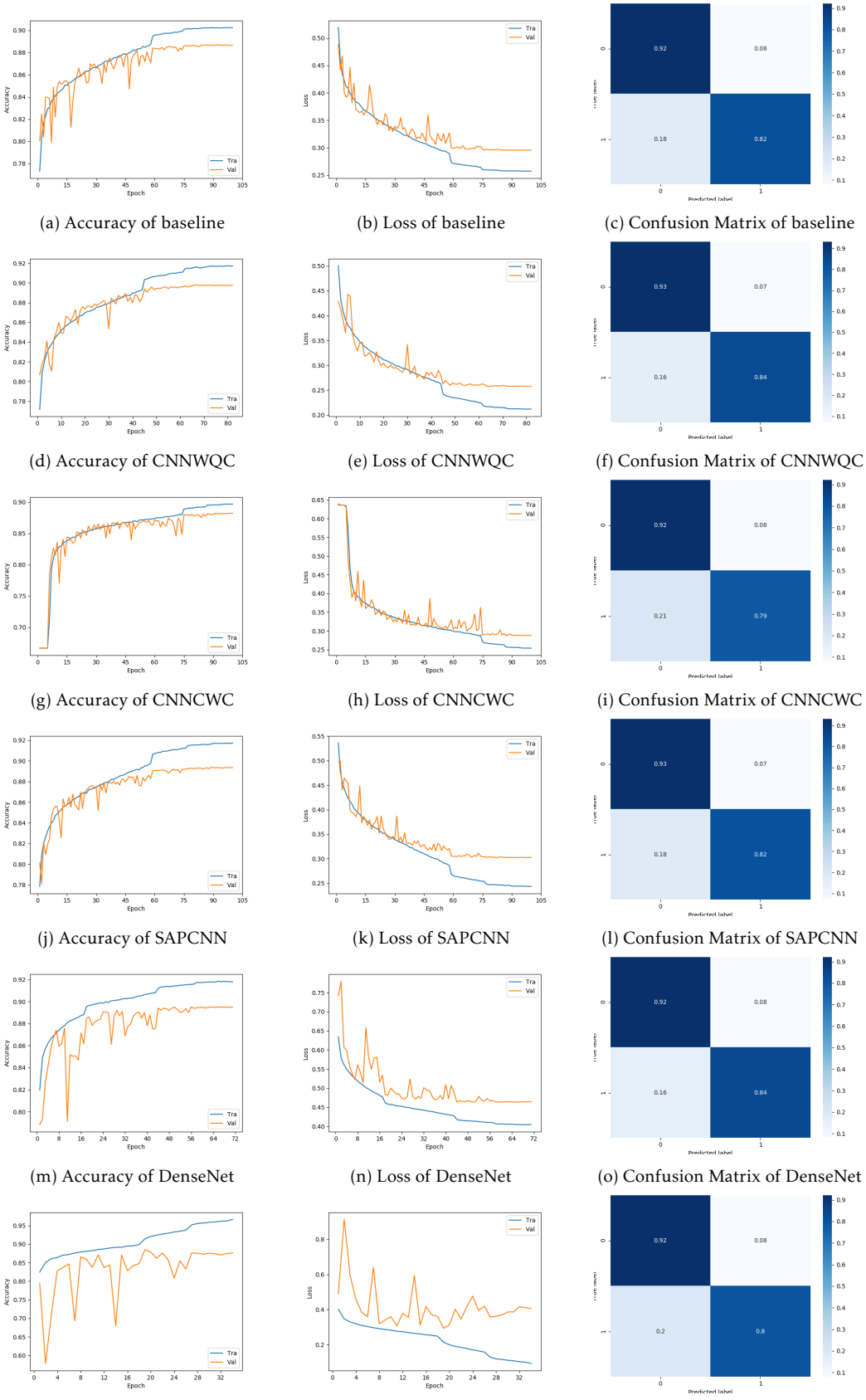
(a) Accuracy of baseline (b) Loss of baseline (c) Confusion Matrix of baseline

(d) Accuracy of SAPCNN (e) Loss of SAPCNN (f) Confusion Matrix of SAPCNN

(g) Accuracy of DenseNet (h) Loss of DenseNet (i) Confusion Matrix of DenseNet

(j) Accuracy of AttResNet (k) Loss of AttResNet (l) Confusion Matrix of AttResNet

Figure I.1: A schematic performance of the models on patch size 16 during initial experiments



(a) Accuracy of baseline (b) Loss of baseline (c) Confusion Matrix of baseline

(d) Accuracy of CNNWQC (e) Loss of CNNWQC (f) Confusion Matrix of CNNWQC

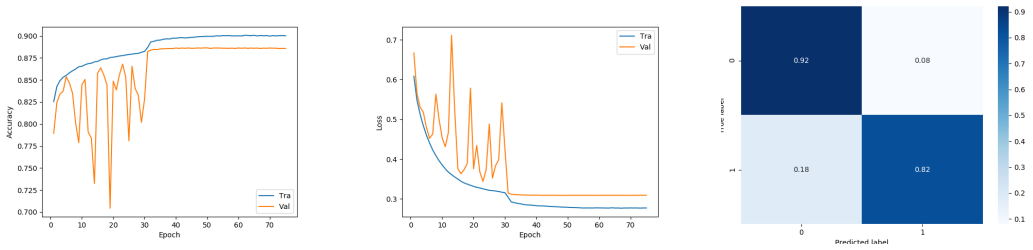
(g) Accuracy of CNNCWC (h) Loss of CNNCWC (i) Confusion Matrix of CNNCWC

(j) Accuracy of SAPCNN (k) Loss of SAPCNN (l) Confusion Matrix of SAPCNN

(m) Accuracy of DenseNet (n) Loss of DenseNet (o) Confusion Matrix of DenseNet

(p) Accuracy of AttResNet (q) Loss of AttResNet (r) Confusion Matrix of AttResNet

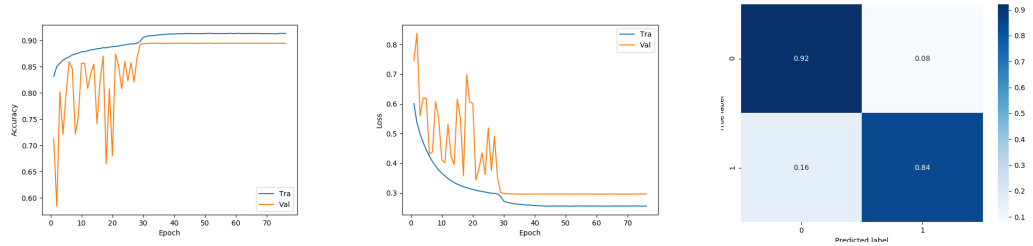
Figure I.2: A schematic performance of the models on patch size 16 during final experiments



(a) Accuracy at 8

(b) Loss at 8

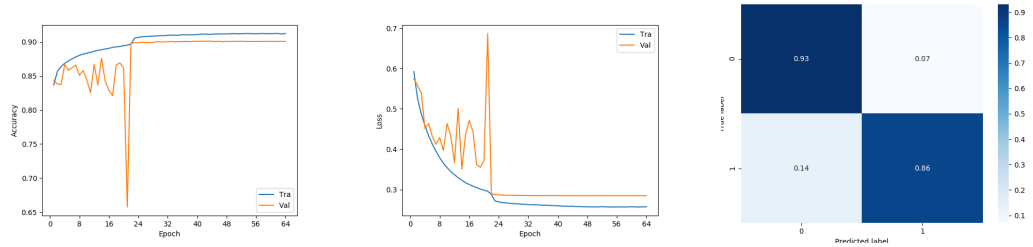
(c) Confusion Matrix at 8



(d) Accuracy at 12

(e) Loss at 12

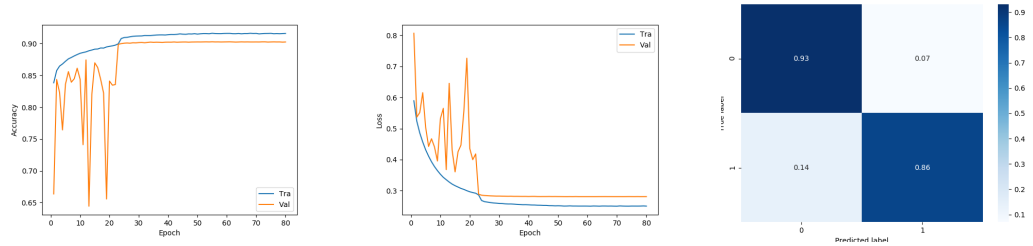
(f) Confusion Matrix at 12



(g) Accuracy at 16

(h) Loss at 16

(i) Confusion Matrix at 16

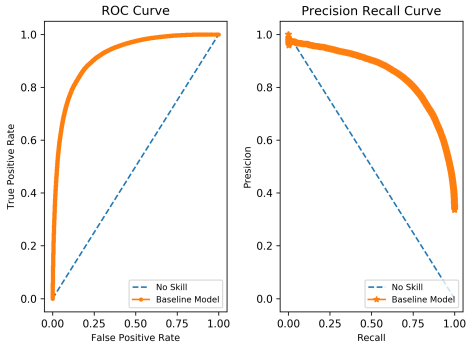


(j) Accuracy at 20

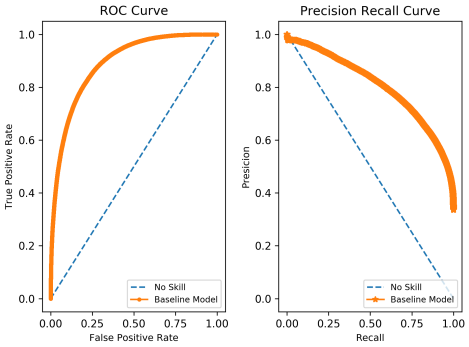
(k) Loss at 20

(l) Confusion Matrix at 20

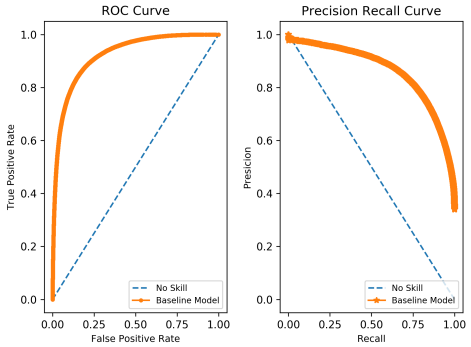
Figure I.3: A schematic performance of the proposed model on varying patch sizes during final experiments



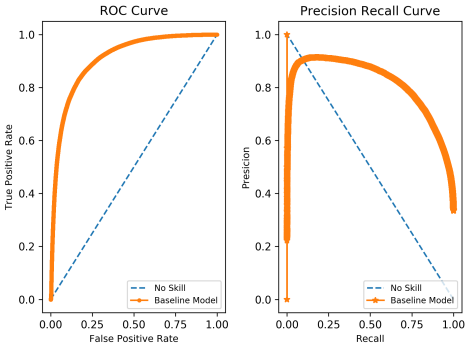
(a) Model



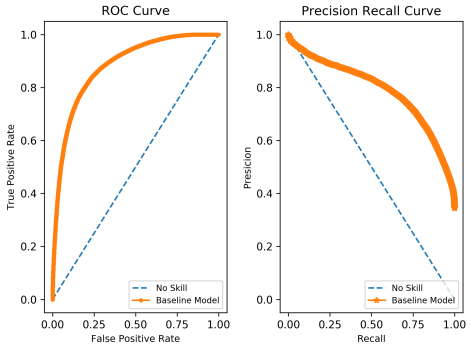
(b) ROC- and PR- Curves of CNNWQC



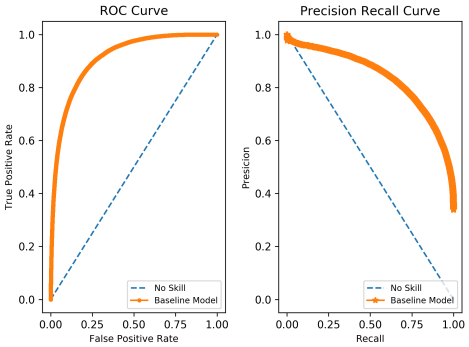
(c) CNNCWC



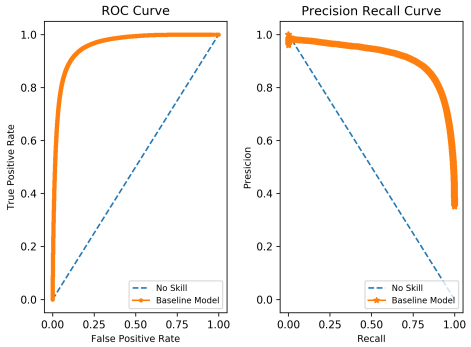
(d) ROC- and PR- Curves of SAPCNN



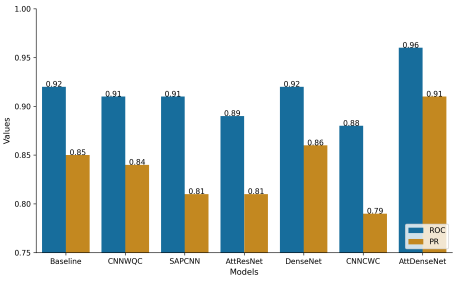
(e) DenseNet



(f) AttResNet



(g) AttDenseNet



(h) ROC and PR Values of models

Figure I.4: ROC and PR Curves of Models

Masters Program in **Geospatial Technologies**

Supported by:
Education and Culture
ERASMUS MUNDUS